

Universidad de las Ciencias Informáticas
Facultad 7



**Título: Sistema de Almacenamiento de la
Información del Bloque Quirúrgico Oftalmológico
del Sistema de Información Hospitalaria.**

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

Autor: Alexander Rodríguez Rabelo.

Tutor: Lic. Dainerys Castañero Rodríguez.

Cotutor: Ing. Mauricio G. Espinosa Robaina.

Asesor: Lic. Jorge Castillo Maceo

Ciudad Habana, julio 2007

*No son las locomotoras, sino las ideas, las
que llevan y arrastran al mundo.*

Victor Hugo

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 4 días del mes de julio del año 2007.

Alexander Rodríguez Rabelo

Dainerys Castañero Rodríguez

DATOS DE CONTACTO

Tutor: Lic. Dainerys Castañero Rodríguez. Profesora graduada en Licenciatura en Ciencias de la Computación en el año 2004. Ha impartido asignaturas de Sistemas de Bases de Datos, Ingeniería de Software y Gestión de Software. Posee categoría docente de Instructor y cursa la maestría de Ciencias de la Computación.

Correo electrónico: dainerysc@uci.cu

Cotutor: Ing. Mauricio G. Espinosa Robaina. Graduado de Ingeniero Electrónico, especialidad Máquinas Computadoras en 1985. Administrador de Red en el Instituto Cubano de Oftalmología "Ramón Pando Ferrer". Jefe de Proyecto para la Confección de Historia Clínica Computarizada de Oftalmología.

Correo electrónico: mger@infomed.sld.cu

AGRADECIMIENTOS

Quiero agradecer a todos los que me han ayudado a lograr este sueño, a los que han contribuido tanto directa como indirectamente en la conformación del presente trabajo de diploma.

Al incansable Mauricio, a las pacientes tutoras, tanto a las oficiales como las no oficiales, y a mis compañeros de módulo, porque conformamos un verdadero equipo.

También a todos mis profesores, los de ahora y los de antes, sin los que me hubiera resultado imposible llegar.

DEDICATORIA

A mi madre, mi padre y mi hermana, por el amor, por el ejemplo y por el cariño. Por ser mi vida.

A mi novia, por su inagotable paciencia y por su ternura.

A mis familiares todos, por la confianza inacabable de que llegaría a la meta.

A mis amigos, por serlo.

Al mayor y eterno maestro, Fidel.

RESUMEN

Los hospitales oftalmológicos se han transformado en los últimos años, por lo que se necesita automatizar el proceso que se lleva a cabo en ellos y no se cuenta con una aplicación para esto, por lo que el presente trabajo de diploma propone instaurar una base de datos para el almacenamiento de la información referente al proceso quirúrgico que tiene lugar en los mismos.

Para esto, se ha utilizado el Gestor de Base de Datos PostgreSQL y la herramienta de modelado Case Studio. Su utilización ha permitido generar una base de datos y sus consultas, que se encuentran optimizadas. Se garantiza un tiempo de respuesta acorde con las necesidades del sistema. Se ha establecido una arquitectura de base de datos centralizada. Así como, el medio y la forma en que interactúa el sistema con el resto de los módulos.

El sistema propuesto se encuentra en tercera forma normal, y debe llegar sólo hasta esta para lograr un equilibrio entre el modelo físico y la respuesta de las consultas. El sistema no presenta redundancias fuera de control, y se garantiza la integridad de los datos.

Con la base de datos propuesta se pretende automatizar la totalidad del proceso de almacenamiento de la información correspondiente al Bloque Quirúrgico Oftalmológico, lo que agilizará el paso de los pacientes por los servicios brindados en los hospitales oftalmológicos y facilitará el manejo y control de la información en los mismos.

PALABRAS CLAVE

Bloque Quirúrgico Oftalmológico, Sistema de Gestión de Base de Datos, Almacenamiento de Información.

ÍNDICE DE TABLAS

Tabla 2.1 Descripción de la clase ExamenRefraccion.	34
Tabla 2.2 Descripción de la clase ExamenTensionOcular.	34
Tabla 2.3 Descripción de la clase ExamenQueratometria.....	35
Tabla 2.4 Descripción de la clase LenteIntraocular.....	36
Tabla 2.5 Descripción de la clase ExamenPadecimiento.....	37
Tabla 2.6 Descripción de la clase ExamenBiometria.	37
Tabla 2.7 Descripción de la clase ExamenAnatomia.	38
Tabla 2.8 Descripción de la clase ExamenesOftalmologicos.	39
Tabla 2.9 Descripción de la clase AnuncioCirugiaCrist.....	40
Tabla 2.10 Descripción de la clase InformeOperatorioCristalino.....	41
Tabla 2.11 Descripción de la clase AnuncioOperatorioOftalmologico.	42
Tabla 2.12 Descripción de la tabla bqo_antecedentes_patologicos_ofthalmologicos.	46
Tabla 2.13 Descripción de la tabla bqo_anuncio_cirugia_crist.....	47
Tabla 2.14 Descripción de la tabla bqo_anuncio_ofthalmologico.....	47
Tabla 2.15 Descripción de la tabla bqo_biometria.	48
Tabla 2.16 Descripción de la tabla bqo_cirugia_anterior.....	48
Tabla 2.17 Descripción de la tabla bqo_cirugia_anterior_consulta.	49
Tabla 2.18 Descripción de la tabla bqo_cristalino_examen.	49
Tabla 2.19 Descripción de la tabla bqo_datos_entrada.	49
Tabla 2.20 Descripción de la tabla bqo_datos_equipo.....	50
Tabla 2.21 Descripción de la tabla bqo_datos_sistema.	50
Tabla 2.22 Descripción de la tabla bqo_examen_cristalino.	51
Tabla 2.23 Descripción de la tabla bqo_examen_fondo_ojo.....	51
Tabla 2.24 Descripción de la tabla bqo_examen_motilidad.	52
Tabla 2.25 Descripción de la tabla bqo_examen_motilidad_ocular.....	52
Tabla 2.26 Descripción de la tabla bqo_examen_ofthalmologico_anatomia.	52
Tabla 2.27 Descripción de la tabla bqo_examen_reflejo_pupilar	53
Tabla 2.28 Descripción de la tabla bqo_examen_segmento_anterior.	53

Tabla 2.29 Descripción de la tabla bqo_examen_vitreo.....	54
Tabla 2.30 Descripción de la tabla bqo_examen_anexos.....	54
Tabla 2.31 Descripción de la tabla bqo_examenes Oftalmologicos.....	54
Tabla 2.32 Descripción de la tabla bqo_informe_cirugia_refractiva.....	56
Tabla 2.33 Descripción de la tabla bqo_informe_operatorio_cristalino.....	56
Tabla 2.34 Descripción de la tabla bqo_informe_operatorio Oftalmologico.....	57
Tabla 2.35 Descripción de la tabla bqo_lente_anuncio.....	57
Tabla 2.36 Descripción de la tabla bqo_lente_informe.....	58
Tabla 2.39 Descripción de la tabla bqo_lente_intraocular.....	58
Tabla 2.40 Descripción de la tabla bqo_queratometria.....	58
Tabla 2.41 Descripción de la tabla bqo_refraccion.....	59
Tabla 2.42 Descripción de la tabla bqo_tension_ocular.....	59
Tabla 2.43 Descripción de la tabla cfg_antecedente Oftalmologico.....	60
Tabla 2.44 Descripción de la tabla cfg_datos_parametros.....	60
Tabla 2.45 Descripción de la tabla cfg_forma_extraccion.....	60
Tabla 2.46 Descripción de la tabla cfg_formula_dioptria.....	60
Tabla 2.47 Descripción de la tabla cfg_iridectomia.....	61
Tabla 2.48 Descripción de la tabla cfg_lugar_colocacion_cristalino.....	61
Tabla 2.49 Descripción de la tabla cfg_ojo.....	61
Tabla 2.50 Descripción de la tabla cfg_otropia.....	61
Tabla 2.51 Descripción de la tabla cfg_parametro_anexos.....	62
Tabla 2.52 Descripción de la tabla cfg_parametro_cristalino.....	62
Tabla 2.53 Descripción de la tabla cfg_parametro_fondo_ojo.....	62
Tabla 2.54 Descripción de la tabla cfg_parametro_medicion.....	62
Tabla 2.55 Descripción de la tabla cfg_parametro_motilidad_ocular.....	63
Tabla 2.56 Descripción de la tabla cfg_parametro_reflejo_pupilar.....	63
Tabla 2.57 Descripción de la tabla cfg_parametro_segmento_anterior.....	63
Tabla 2.58 Descripción de la tabla cfg_parametro_vitreo.....	63
Tabla 2.59 Descripción de la tabla cfg_tipo_anuncio_quirurgico.....	64
Tabla 2.60 Descripción de la tabla cfg_tipo_catarata.....	64
Tabla 2.61 Descripción de la tabla cfg_tipo_cirugia.....	64

Tabla 2.62 Descripción de la tabla cfg_tipo_informe_operatorio.....	64
Tabla 2.63 Descripción de la tabla cfg_tipo_lente.....	65
Tabla 2.64 Descripción de la tabla cfg_tipo_tratamiento_refractivo.	65

ÍNDICE DE FIGURAS

Figura 2.1 Diagrama de clases persistentes (parte 1).	31
Figura 2.2 Diagrama de clases persistentes (parte 2).	32
Figura 2.3 Diagrama Entidad Relación (parte 1).....	44
Figura 2.4 Diagrama Entidad Relación (parte 2).....	45
Figura A1.0 Estructura Anatómica del Ojo Humano	82

ÍNDICE

AGRADECIMIENTOS.....	I
DEDICATORIA.....	II
RESUMEN.....	III
ÍNDICE	VII
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Conceptos básicos relacionados con el problema.	6
1.2 Análisis de la metodología a utilizar.....	8
1.3 Herramientas a utilizar.....	9
1.4 Los Sistemas de Gestión de Bases de Datos.....	10
1.5 Sistemas automatizados precedentes vinculados al campo de acción.	15
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.	19
2.1 Estrategia de integración de la solución con otros módulos del sistema.....	19
2.2 Descripción y fundamentación de la arquitectura.....	20
2.3 Análisis de optimización de querys.	21
2.4 Selección y argumentación de los requisitos funcionales y no funcionales.....	27
2.5 Diagrama de clases persistentes obtenido a partir del diagrama de clases del diseño.	30
2.6 Descripción de las clases.....	33
2.7 Diseño de la base de datos.	42
2.7.1 Diagrama entidad relación de la base de datos.	43
2.7.2 Descripción de las tablas.	46

CAPÍTULO 3: VALIDACIÓN DEL DISEÑO REALIZADO	66
3.1 Validación teórica del diseño.....	66
3.1.1 Integridad.....	66
3.1.2 Normalización de la base de datos.....	68
3.1.3 Análisis de redundancia de la información.....	70
3.1.4 Análisis de la seguridad de la base de datos.....	71
3.1.5 Trazabilidad de las acciones.....	72
CONCLUSIONES	74
RECOMENDACIONES.....	75
REFERENCIAS BIBLIOGRÁFICAS.....	76
BIBLIOGRAFÍA.....	80
ANEXOS	82
Anexo1 Estructura anatómica del ojo humano.....	82
Anexo2 Fragmento de la licencia de Oracle.....	83
Anexo3 Ejemplo de funciones y su optimización.....	84
GLOSARIO.....	88

INTRODUCCIÓN

La modernización y desarrollo de los medios para el cuidado de la salud humana, es una de las principales prioridades para el futuro de la humanidad. Que dependerá en gran medida del potencial humano y la gestión de los conocimientos. La informática en sus diferentes manifestaciones, tiene asegurado un papel protagónico en este sentido.

Desde hace ya varias décadas, los sistemas informáticos ayudan a los profesionales de la salud en su trabajo, tanto en la gestión de la información en grandes hospitales, como en la investigación científica, la docencia médica, la asistencia a enfermos, el diagnóstico y tratamiento a pacientes y en el procesamiento de estadísticas médicas.

Cuba, en momentos que la globalización neoliberal arrasa despiadadamente por los más diversos escenarios, se propone la utilización del conocimiento y las tecnologías de forma justa y racional sobre principios éticos sostenibles.

Durante los últimos 20 años un grupo de instituciones cubanas han desarrollado sistemas encaminados a lograr determinados niveles de informatización de la salud. Es así, que en 1992 se crea la Red Telemática de la Salud, INFOMED, dentro de la estructura del Centro Nacional de Información de Ciencias Médicas. Utilizando los recursos disponibles y permitiendo enlazar a todo el sistema de salud, para dar una respuesta más eficiente en la esfera de la información científica a los profesionales, técnicos de la salud y a la situación sanitaria del país. Surge con una acertada visión de la influencia que las Tecnologías de la Información y las Comunicaciones ejercerán en la esfera de la información y conocimiento y fruto de la voluntad política del país y la colaboración y solidaridad nacional e internacional. [1]

Sin embargo, mientras no se establecieran normas claras para el desarrollo informático del Estado, se corría el riesgo de aplicar productos incompatibles, incompletos, obsoletos e insostenibles que no permitirían un desarrollo armónico e integrado de la salud cubana, propiciando incluso duplicidades innecesarias, accidentes no deseados e ineficacia en los procesos.

Ya en 1997 la alta dirección del Ministerio de Salud Pública (MINSAP) había propuesto: desarrollar en el Sistema Nacional de Salud una política única con un modelo integral de informatización a los diferentes niveles del mismo. Así como el acceso a la información como proceso que apoyara y potenciara

decisivamente la asistencia médica, la docencia, la investigación, la higiene y la epidemiología, la industria médico farmacéutica, la economía y administración de salud, que se extendiera de forma integral a todas las instituciones del país para alcanzar un Sistema Integrado de Gestión que sirviera como herramienta básica en la materialización de las estrategias y programas de salud. [2]

Actualmente, el MINSAP ha definido a la informatización como una de sus prioridades. Ha convocado para ello a un grupo de instituciones propias del sector, del Ministerio de Informática y Comunicaciones y de otros organismos de la Administración Central del Estado, para definir de conjunto la estrategia a desarrollar.

En algunos casos, se han tomado como punto de partida sistemas ya desarrollados en el país en el marco de aquella primera estrategia de desarrollo. [3]

En estos momentos se trabaja integradamente en el desarrollo de un grupo de aplicaciones básicas para la informatización del sector de la salud; en proyectos basados en nuevas tecnologías de Internet, software libre y otros que garantizan una explotación integrada y compatible y que serán introducidos utilizando como infraestructura la Red Telemática de la Salud.

Cuba ha desarrollado la colaboración médica, como parte de esta se encuentra la Misión Milagro, destinada a devolver la visión a la mayor parte de la población pobre de América Latina y el resto del mundo. Esta resulta una prioridad fundamental tanto para el Estado Cubano como para el resto de los gobernantes progresistas latinoamericanos y caribeños. El Hospital Oftalmológico "Ramón Pando Ferrer" forma parte de el proceso de informatización del sector de la salud conjuntamente con instituciones como la Universidad de las Ciencias Informáticas, específicamente con la facultad siete, que se desenvuelve en el perfil de informática para la salud.

A finales de la década del 80 en el Hospital Oftalmológico "Ramón Pando Ferrer" se desarrolló un sistema que permite gestionar y administrar parte de la información generada durante los procesos por los que atraviesa un paciente al ser atendido, el SAMC (Sistema Automatizado de Microcirugía), el cual, en la actualidad se encuentra implantado y en funcionamiento.

Dicho sistema sólo brinda una solución particular para este hospital y está implementado en "Clipper'85 Winter", lenguaje de programación creado en 1985 por Nantucket Corporation y herramienta líder de

desarrollo, bajo sistema operativo MS-DOS, de aplicaciones relacionadas con bases de datos, sobre todo programas de facturación y contabilidad.

Debido a los cambios realizados en los hospitales oftalmológicos en los últimos años y que dicho sistema sólo ofrece una solución particular a este hospital y no cubre todos los servicios oftalmológicos que se brindan en el mismo, este ha dejado de responder a intereses principales de los usuarios.

El Sistema Automatizado de Microcirugía brinda soporte solamente al servicio de microcirugía del cristalino, y no abarca otros como cirugía refractiva, terigium, cirugía de retina y miopía. Por lo que el proceso de atención a los pacientes en estos servicios se realiza manualmente en su totalidad, lo que dificulta la entrega de reportes y provoca que el desarrollo de estos servicios sea más lento. Aparejado a esto se encuentra el aumento del volumen de pacientes que se atienden diariamente debido a la puesta en marcha de la Misión Milagro, por lo que se necesita un manejo de la información de forma rápida y confiable.

Es así que se determina entonces el **problema científico**: ¿Cómo mejorar la gestión de la información referente al proceso quirúrgico en los hospitales oftalmológicos?

De aquí la **necesidad** de desarrollar un sistema que permita a los hospitales oftalmológicos digitalizar la información que se genera durante la intervención quirúrgica de un paciente, registrando y actualizando los datos de la historia clínica, lo cual supondrá un acceso rápido y sencillo a los mismos.

Este problema se enmarca en el **objeto de estudio**: proceso de automatización de la información hospitalaria.

El **campo de acción** comprendido es el proceso de automatización del almacenamiento y la manipulación de la información quirúrgica de los pacientes durante los servicios brindados en los hospitales oftalmológicos.

Por lo que el **objetivo general** de la presente investigación lo constituye: instaurar una base de datos para el almacenamiento de la información referente al proceso quirúrgico que tiene lugar en los hospitales oftalmológicos.

Este se puede a su vez, subdividir en los siguientes **objetivos específicos**:

1. Diseñar e implementar un sistema de bases de datos integrado de gestión quirúrgica en hospitales oftalmológicos.
2. Garantizar máxima seguridad y fiabilidad en los datos.
3. Garantizar la interoperabilidad y flexibilidad del sistema.

Para dar cumplimiento a estos objetivos trazados se prevé la realización de las siguientes **tareas investigativas**:

- Estudiar el análisis realizado por el analista del sistema sobre el funcionamiento del Bloque Quirúrgico Oftalmológico en los hospitales.
- Definir las herramientas a utilizar para el diseño de la base de datos.
- Analizar la estrategia de integración de la solución con otros módulos o sistemas.
- Implementar los procedimientos almacenados del módulo.
- Hacer una validación teórica y funcional del diseño de la base de datos propuesta.
- Realizar una valoración de los resultados y hacer propuestas de integración con el diseño.

Con el desarrollo del presente se pretende diseñar y desarrollar la base de datos necesaria para los procesos que se llevan a cabo en el Bloque Quirúrgico Oftalmológico de manera segura y confiable facilitando un rápido nivel de acceso a los datos recogidos en dicho proceso.

El presente trabajo de diploma está estructurado de la siguiente manera: introducción, tres capítulos de desarrollo, conclusiones, recomendaciones, referencias bibliográficas, glosario de términos y anexos.

En el Capítulo 1, Fundamentación Teórica. Se presenta el estado del arte sobre el tema a nivel internacional, nacional y de la universidad, así como de las tendencias, técnicas, tecnologías y metodologías utilizadas para la solución del problema descrito.

En el Capítulo 2. Descripción y análisis de la solución propuesta. Se presenta la estrategia de integración de la solución con otros módulos o sistemas, la descripción de la arquitectura y su fundamentación, el análisis de optimización de queries, la referencia al trabajo del analista en cuanto a la selección y argumentación de los requisitos funcionales y no funcionales del sistema, el Diagrama de Clases

Persistentes obtenido a partir del Diagrama de Clases del Diseño y la descripción de las principales clases.

En el Capítulo 3. Validación del diseño realizado. Se presenta la validación teórica del diseño teniendo en cuenta la integridad, normalización, análisis de redundancia de información, análisis de la seguridad y la trazabilidad de las acciones llevadas a cabo en la base de datos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Este capítulo pretende facilitar la comprensión mediante la descripción y el análisis en el marco teórico del proceso de almacenamiento de la información en el módulo Bloque Quirúrgico Oftalmológico; explicar las tendencias, técnicas y tecnologías relacionadas con este; así como realizar un estudio del estado del arte de las mismas a nivel internacional, nacional y de la universidad. Es el resultado de la búsqueda y el análisis valorativo de información vinculada al respecto.

1.1 Conceptos básicos relacionados con el problema.

De forma sencilla se puede definir el **acto quirúrgico** como el conjunto de acciones efectuadas a un paciente en una misma oportunidad, por uno o más cirujanos en el pabellón o quirófano, generalmente bajo anestesia y para un tratamiento determinado, por su parte un **bloque quirúrgico** es definido como el área centralizada en la que se realiza toda la actividad quirúrgica del hospital, es decir, se realizan todas las cirugías programadas correspondientes a pacientes ya sean hospitalizados, ambulatorios o casos de urgencia.

La **especialidad de oftalmología** es la que se ocupa de la prevención y el tratamiento, tanto médico como quirúrgico, de todo lo relacionado al ojo y sus anexos, (Ver Anexo 1) como pueden ser miopía, astigmatismo, hipermetropía, cataratas, estrabismo, glaucoma, etc. [4]

A partir de aquí entonces se define al **bloque quirúrgico oftalmológico** como el área centralizada donde se realizan todas las actividades quirúrgicas correspondientes a la especialidad de oftalmología.

El **anuncio operatorio** es un documento que emite el cirujano en la consulta especializada, donde prevé todo lo concerniente a la futura intervención quirúrgica que se realizará al paciente. En este, además de la fecha, hora y lugar de la operación se recogen datos de interés quirúrgico como son: los medicamentos y el instrumental necesarios, el personal médico que intervendrá en la cirugía y datos importantes desde el punto de vista del diagnóstico realizado. Por su parte el **informe operatorio** es el documento que se emite al concluir la cirugía donde se recogen todas las incidencias de la misma y se verifica el cumplimiento o no de lo previsto en el anuncio operatorio.

También se puede definir una serie de conceptos asociados a la oftalmología que serán muy útiles a la hora de comprender el desarrollo del presente, estos serían: **agudeza visual** que es la mejor visión que una persona puede alcanzar, cuando es perfecta, se dice que esa persona tiene una agudeza visual de “la unidad”, (en Europa equivale a 1 y en medida americana equivale a 20/20). Se considera normal la agudeza visual cuando está entre 0’5 y 1. [5]

El cristalino, es el más débil de los elementos refractivos que conforman el sistema óptico humano, su poder dióptrico equivale a unas veintiuna dioptrías positivas. Su función es enfocar los rayos de luz que entran al globo ocular sobre la retina. [6] (Ver Anexo 1). Por otra parte se tiene el **lente intraocular** que es el lente que se coloca en el ojo en sustitución del cristalino cuando se realiza una cirugía de extracción del mismo y el oftalmólogo determina la necesidad de este. El tipo de lente y el poder dióptrico del mismo son determinados por el especialista a partir de un conjunto de reglas y fórmulas preestablecidas con este fin.

También se define una **base de datos** como el conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una base de datos puede considerarse una colección de datos variables en el tiempo. El software que permite la utilización y/o la actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, se denomina **Sistema de Gestión de Bases de Datos**. El objetivo fundamental de un Sistema de Gestión de Bases de Datos consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado. [7]

En la actualidad dichos sistemas se pueden agrupar en dos grandes modelos: los **Sistemas de Gestión de Bases de Datos Relacionales** que son las bases de datos que se generan y se construyen con información muy estructurada (datos) acerca de una organización o empresa determinada. Cuando un usuario realiza una consulta en una base de datos relacional, el sistema presenta como resultado la respuesta exacta a lo que se busca [8] y los **Sistemas de Gestión de Bases de Datos Documentales o Sistemas de Recuperación de Información** que son las bases de datos que se generan y se construyen con información no estructurada tipo texto (documentos) sobre uno o varios temas. Cuando un usuario realiza una consulta en una base de datos documental, el sistema presenta como resultado, no una respuesta exacta, sino documentos útiles para satisfacer la pregunta del usuario.

1.2 Análisis de la metodología a utilizar.

Siempre que se desarrolla un software, es necesario para su puesta a cabo, en tiempo y cumpliendo con todo lo que el cliente solicita, utilizar una buena metodología de desarrollo, ya que este proceso, riesgoso y difícil no está exento de errores y una correcta metodología resultaría de gran ayuda en este sentido. En el presente, los analistas del módulo determinaron utilizar la Metodología RUP (Rational Unified Process). [9]

Esta metodología se adapta mejor al proceso, debido a que está definida para proyectos de largo plazo o, en otras palabras, de grandes dimensiones. Para esto, se divide en 4 fases el desarrollo del software:

1. Inicio: Se determina la visión del proyecto.
2. Elaboración: Se determina la arquitectura óptima.
3. Construcción: Se obtiene la capacidad operacional inicial.
4. Transición: Se obtiene el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

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

Sus principales características se centran en implementar las mejores prácticas en Ingeniería de Software, forma disciplinada de asignar tareas y responsabilidades, quién hace qué, cuándo y cómo; administración de requisitos, uso de arquitectura basada en componentes, control de cambios y modelado visual del software.

Esta metodología posee tres características fundamentales, la primera de ellas es que su desarrollo es iterativo e incremental por lo que divide el proceso de desarrollo en ciclos, teniendo un producto final al terminar cada ciclo. La segunda es que está guiado por los Casos de Uso. Un Caso de Uso será aquello

que describe un fragmento de las funcionalidades del sistema que proporciona al usuario un resultado importante. Los Casos de Uso guían el diseño construcción y prueba del sistema, esto significa que guían el proceso de desarrollo.

Por último y no la menos importante, está centrado en la arquitectura, lo que le permite los desarrolladores una mayor visibilidad del sistema, pues la arquitectura es una vista del diseño completo del software con las características más importantes resaltadas, dejando a un lado los detalles. Utiliza como lenguaje de modelado UML, el que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa. [10]

1.3 Herramientas a utilizar.

Para el desarrollo del Sistema de Almacenamiento de Información del Bloque Quirúrgico Oftalmológico, se necesitan de varias herramientas de modelado, las cuales se describen a continuación.

Case Studio: herramienta libre de modelado de software que presenta varias facilidades, su interfaz gráfica, amigable e intuitiva, soporta Oracle, MySQL, MS SQL, MaxDB, Firebird, PostgreSQL y otros sistemas. El programa permite generar rápidamente diagramas gráficos de las bases de datos relacionales con las que se está trabajando, simplificando mucho el trabajo del programador. Otra de sus principales características, son su potente sistema de ingeniería inversa, que permite identificar y estructurar bases de datos ya existentes para poder trabajar con ellas sin problemas.

Esta aplicación de la familia de herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) está destinada a aumentar la productividad en el desarrollo de software reduciendo el costo de los mismos en términos de tiempo y de dinero. [11]

Por otra parte para el modelado del diagrama de clases persistentes así como el modelo de datos, se utiliza la herramienta Rational Rose Enterprise que es uno de los productos más completos de la familia

Rational Rose. Todos los productos de Rational Rose dan soporte a Lenguaje Unificado del Software, aunque no son compatibles con las mismas tecnologías de implementación. [12]

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

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

1.4 Los Sistemas de Gestión de Bases de Datos.

Las bases de datos y sus tecnologías tienen un impacto decisivo con el creciente uso de las de la informática en la totalidad de las ramas de la ciencia y de las esferas de la vida humana.

Existen muchas formas de organizar las bases de datos, pero hay un conjunto de objetivos generales que deben cumplir todos los Sistemas de Gestión de Bases de Datos, de modo que faciliten el proceso de diseño de aplicaciones y que los tratamientos sean más eficientes y rápidos, dando la mayor flexibilidad posible a los usuarios. Los objetivos fundamentales son: [13]

La abstracción de la información. Ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

La independencia de los datos, que consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

La redundancia mínima que parte de un buen diseño de una base de datos, lo que logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

La consistencia que es en aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

La seguridad de la información almacenada en una base de datos puede llegar a tener un gran valor. Los Sistemas de Gestión de Bases de Datos deben garantizar que esta información se encuentra asegurada frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado.

Normalmente, disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

La integridad, se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se protegen los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

Por otro lado se tiene el respaldo y recuperación, ya que todo Sistema de Gestión de Bases de Datos debe proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

El control de la concurrencia ya que en la mayoría de entornos, lo más habitual es que sean muchas las personas que acceden a una base de datos; bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un Sistema de Gestión de Bases de Datos debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias y el tiempo de respuesta, que lógicamente se desea minimizar, que es el tiempo que el sistema tarda en devolver la información solicitada y en almacenar los cambios realizados.

Las principales ventajas de un Sistema de Gestión de Bases de Datos son:

1. Facilidad de manejo de grandes volúmenes de información.
2. Gran velocidad en muy poco tiempo.
3. Independencia del tratamiento de información.

4. Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
5. No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
6. Integridad referencial al terminar los registros.

Por otra parte, los principales inconvenientes son:

1. El costo de actualización del hardware y software son muy elevados.
2. El nivel de conocimiento al respecto, por el administrador de la base de datos debe ser muy elevado.
3. El mal diseño de esta puede originar problemas en el futuro.
4. Un mal adiestramiento a los usuarios también puede originar problemas.
5. Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.
6. Generan campos vacíos en exceso.
7. Un mal diseño de seguridad genera problemas en esta.[14]

Actualmente los principales gestores de bases de datos que se utilizan a nivel mundial para desarrollar sistemas similares al Sistema de Almacenamiento de la Información perteneciente al Bloque Quirúrgico Oftalmológico, son Oracle, PostgreSQL, MySQL y Microsoft SQLServer.

A la hora de decidir por uno de estos gestores de base de datos y basado en los elementos definidos por los arquitectos del sistema, [15] se tuvo en cuenta: el cumplimiento de los lineamientos trazados para el proceso de informatización del Ministerio de Salud Pública, que parte de crear soluciones a partir de software libre. Luego, se decidió entre los gestores libres teniendo en cuenta las ventajas y desventajas de cada uno de ellos. El análisis de los elementos anteriores se realizó de la siguiente manera:

En el caso de **Oracle** que es uno de los sistemas gestores de bases de datos más potentes y desarrollados de hoy en día, se puede plantear que cuenta con grandes prestaciones como: soporte de transacciones, gran estabilidad y escalabilidad además de ser multiplataforma. Oracle surge a finales de los 70 bajo el nombre de Relational Software a partir de un estudio sobre Sistemas Gestores de Base de

Datos. Este estudio incluía una comparativa de productos que erigía a Relational Software como el más completo desde el punto de vista técnico. Esto se debía a que usaba la filosofía de las bases de datos relacionales, algo que por aquella época era todavía desconocido.

En la actualidad, Oracle aún encabeza la lista. Su tecnología se encuentra prácticamente en todas las industrias alrededor del mundo. Es la primera compañía de software que desarrolla e implementa software para empresas ciento por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. [16]

Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias). Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios [17]

Sin embargo existe un elemento adicional que hace imposible utilizar este gestor de base de datos y es el respaldo de sus dueños al bloqueo norteamericano hacia Cuba, esto se puede apreciar textualmente en sus licencias, (Ver Anexo 2) donde se expresa que este producto no puede ser utilizado por ningún ciudadano cubano.

Por su parte **Microsoft SQLServer**, producto de Microsoft Software Corporation es un Sistema de Gestión de Bases de Datos relacionales basado en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Este presenta múltiples ventajas que a se describen continuación: soporte de transacciones, gran estabilidad, gran seguridad, escalabilidad, soporta procedimientos almacenados, incluye también un potente entorno gráfico de administración, que permite el uso de comandos, trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información y administrar información de otros servidores de datos.

Pero este producto también cuenta con el mismo impedimento visto anteriormente en Oracle, en su licencia, puesto que Microsoft tiene como restricción que el usuario del mismo no puede ser de nacionalidad cubana. [18]

Entonces sólo queda la disyuntiva entre los dos gestores de bases de datos, más potentes, que cumplen con los lineamientos del sistema de salud con respecto al software libre, que son **MySQL** y **PostgreSQL**.

Se selecciona uno entre estos gestores, basándose en lo expuesto por los arquitectos del sistema [19] y teniendo en cuenta las ventajas y desventajas de ambos. Se decide elegir a PostgreSQL, que es un gestor de base de datos relacionales, liberado bajo la licencia de software libre.

Además ofrece una potencia adicional sustancial al incorporar cuatro conceptos adicionales básicos, a través de la cual los usuarios pueden extender fácilmente el sistema: Clases, Herencia, Tipos y Funciones. Tiene como ventaja sobre MySQL presenta una mayor velocidad de acceso a los datos en sistemas de bases de datos de gran tamaño como es el caso del presente problema.

Otras características que le aportan potencia y flexibilidad adicional son:

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

Las principales mejoras que en PostgreSQL se incluyen son:

- Los bloqueos de tabla han sido sustituidos por el control de concurrencia multiversión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg_dump mientras la base de datos permanece disponible para consultas.
- Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
- Se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadena literal, conversión de tipos y entrada de enteros binarios y hexadecimales.
- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.
- Establece condiciones o checks para validar las entradas de datos.
- Permite transacciones, es decir, múltiples operaciones de tabla o registros de manera segura.

- Permite bloqueos de registros, útil en entornos multiusuario.
- Administración de grupos de usuarios, y soporte nativo SSL.
- Múltiples lenguajes para procedimientos almacenados (incluyendo el nativo PL/PgSQL, PL/PHP, PL/Perl y PL/Python)
- La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada. [20]

1.5 Sistemas automatizados precedentes vinculados al campo de acción.

Un sistema de base de datos, de un Bloque Quirúrgico Oftalmológico debe permitir almacenar y manipular toda la información concerniente al control y seguimiento de un paciente en el área quirúrgica.

En Cuba, desde hace varios años se han estado desarrollando una serie de software para la gestión hospitalaria, como son el "GALEN HOSPITAL" y el "MEDISyS, ambos implantados en la actualidad y en funcionamiento; sin embargo, ninguno de estos cumple con los requisitos necesarios para dar solución al presente problema, ya que sobre el bloque quirúrgico poseen muy pocas funcionalidades y sobre la especialidad de oftalmología no poseen ninguna.

El único antecedente a nivel nacional del que se tiene noticia hasta el momento, que puede dar una solución al menos parcial, es el software SAMC (Sistema Automatizado de Microcirugía) instalado en el Hospital Oftalmológico "Ramón Pando Ferrer" de la Ciudad de La Habana, dicho software se encarga en parte de este tipo de gestión, este 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.

Este sólo presenta una solución particular para dicho Hospital que 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 reindexar la base de datos en cada ocasión que se quiere obtener dichos reportes.

A nivel mundial, existe un gran número de empresas que están especializadas en el desarrollo e implantación de software para el sector de la salud. Algunas de ellas desarrollan aplicaciones para hospitales y departamentos especializados de este, como los bloques quirúrgicos oftalmológicos.

Así, se pueden encontrar varios como son:

- **ActualOftalmo.** Software desarrollado por la empresa argentina ActualSoft! creada en 1987; 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. [21]
- **Hosic-V.** Es un sistema de gestión hospitalaria flexible, integrado y modular, que abarca todas las áreas de actividad de un hospital, entre ellas las que concierne al presente problema. Está constituido por módulos que pueden funcionar de forma autónoma y se adapta con facilidad a todo tipo de organización. Es también un sistema integrado, que relaciona directamente las diferentes áreas y presenta una base de datos en SQLServer 7. Desarrollado por la empresa SIVSA tiene un costo que oscila desde 500.001 dólares a 2 000.000 dependiendo de la cantidad de puestos de trabajo en los que se vaya a implantar. [22]
- **VANGUARD Optim OTM.** Es un sistema “*real time*” de planificación y documentación para el bloque quirúrgico, independientemente de la especialidad que este ofrezca. Proporciona así las bases de datos para una gestión del bloque quirúrgico más eficiente, desde la planificación, pasando por el control, hasta la elaboración de informes, aportando claras estructuras organizativas y más transparencia en todas las operaciones en torno al bloque quirúrgico [23] desarrollado por la empresa alemana Vanguard.
- **CARE2X.** Es una plataforma informática para la gestión de hospitales hecha bajo licencia de software libre. Está diseñado para integrar los diversos sistemas de información que existen en los hospitales en un solo sistema eficiente. Puede integrar casi cualquier tipo de servicios, de sistemas, de departamentos, de clínica, de procesos, de datos y de comunicación que existan en un hospital. Su diseño puede incluso manejar servicios o funciones no médicos como seguridad y

mantenimiento. Es modular y altamente escalable. Su módulo “Sala de operaciones” es el encargado de documentar las operaciones quirúrgicas (cirugía, anestesia, enfermería, material, medicinas, etc.) y planificar servicios, presenta también funciones de búsqueda y archivo, sistema de documentación, planifica operaciones quirúrgicas y documenta los procesos de tras/post operatorio con diagnósticos y procedimientos quirúrgicos. [24]

- **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, presenta también un sistema de bases de datos implementado sobre SQLServer 2000. Es propiedad de la empresa CHCA computer system inc. [25]
- **Tesis.Quirófanos.** Es un software para la gestión de la información quirúrgica, que permite a cada cirujano, desde su propio consultorio, efectuar la reserva accediendo a la agenda pactada con la clínica, verificándose en este mismo instante la disponibilidad del equipamiento necesario según el tipo de intervención programado. Será, no obstante, desde el área quirúrgica, la que confirmará en última instancia la reserva de la solicitud. Toda la actividad generada en el bloque quirúrgico, incluidos los derechos de quirófano, son transferidos en tiempo real a la cuenta del propio paciente. [26] Utiliza SQL server como gestor de bases de datos.

De los principales software existentes a nivel mundial que se utilizan y/o se desarrollan para la gestión de la información referente a bloques quirúrgicos, específicamente en los oftalmológicos, descritos anteriormente, puede verse, que no brindan una solución factible para dar solución al proceso de almacenamiento de la información correspondiente al Bloque Quirúrgico Oftalmológico de los hospitales cubanos, ya que no se ajustan totalmente a las características del mismo y en el caso de los que brindan al menos una solución parcial, presentan precios elevados que dificultan su utilización pues el despliegue de los mismos en los hospitales del país tendría un costo muy elevado.

Por tanto estos no se pueden aceptar como posibles soluciones para el presente problema; pero sí brindan una posible línea a seguir para dar solución a este, en cuanto a los principales requisitos y funcionalidades que debe cumplir el sistema, así como las principales características de este.

Se puede tomar la experiencia de estas empresas, y los resultados obtenidos como punto de partida para el análisis y el posterior desarrollo del Sistema de Almacenamiento de la Información correspondiente al Bloque Quirúrgico Oftalmológico.

En el presente capítulo se realizó el análisis de los [principales conceptos relacionados con el almacenamiento de la información del Bloque Quirúrgico Oftalmológico, se analizaron los principales gestores de base de datos, de donde se seleccionó PostgreSQL, se analizaron la metodología y las herramientas a utilizar y se realizó un análisis valorativo de los principales software que se tienen como antecedentes del sistema.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

Este capítulo, pretende facilitar la comprensión de la solución propuesta para el Sistema de Almacenamiento de la Información correspondiente al Bloque Quirúrgico Oftalmológico, mediante su descripción y análisis. Abordando para ello, la estrategia de integración con el resto de los módulos del Sistema de Información Hospitalaria con los que interactúa. Además se propone la arquitectura para el Sistema de Base de Datos, la estrategia de optimización de queries, la selección y descripción de los requisitos funcionales y no funcionales del sistema, así como la descripción de las principales clases persistentes, del diagrama entidad relación y la descripción detallada de las tablas.

2.1 Estrategia de integración de la solución con otros módulos del sistema.

El Sistema de Información Hospitalaria, concebido para optimizar el desempeño del personal de salud en los hospitales, brindar información clasificada y disminuir tramites y tiempo invertido por los pacientes en su estancia en los mismos, se encuentra dividido en módulos, los que interactúan entre sí y se complementan en un sistema que automatiza el almacenamiento de la información en todos los procesos que se llevan a cabo en los hospitales. Estos módulos son Inscripción-Admisión, Configuración, Administración, Farmacia, Banco de sangre, Citas, Urgencia, Laboratorio, Bloque Quirúrgico General y Bloque Quirúrgico Oftalmológico. La solución de este último es la que se desarrolla en el presente trabajo de diploma.

Esta solución debe integrarse con varios de los módulos descritos, sin embargo, todo el proceso de integración se lleva a cabo en la capa del negocio, a nivel de base de datos la integración se limita al almacenamiento de llaves foráneas provenientes de la base de datos de los otros módulos. De garantizar la integridad referencial de todos estos datos se encarga también la capa del negocio.

De esta manera el sistema se relaciona con el Módulo de Inscripción-Admisión, con el cual se comunica para obtener la información correspondiente a los pacientes y los funcionarios que interactúan con el sistema, en este caso se obtienen de él y se manipulan los campos *id_persona* e *id_funcionario*. Los que

Capítulo 2. Descripción y análisis de la solución propuesta.

permiten el posterior acceso a todos los datos correspondientes tanto al paciente como a cada uno de los funcionarios que interactúan con el sistema.

También se relaciona, pero de manera mucho más estrecha con el Módulo Bloque Quirúrgico General, el cual antecede al presente en la lógica del negocio a nivel general [27]. De este, se obtiene todo lo concerniente a la consulta general por la que pasa el paciente, *numero_consult* y *tipo_consulta*, al anuncio operatorio, *id_anuncio_operatorio*, y al informe operatorio, *id_informe_operatorio*; clases que en el diagrama de clases persistente utilizado para generar el diagrama de datos, como son el caso de anuncio oftalmológico e informe oftalmológico, heredan directamente de clases implementadas por el módulo Bloque Quirúrgico General pero de realizar este acople se encarga también la capa del negocio directamente.

Con el Módulo de Configuración también se interactúa, ya que este brinda todos los datos de las tablas de configuración con las que se relaciona la solución, estos son los campos *id_diagnostico* de la tabla *cfg_diagnostico* que indica las especificidades y las descripciones de los diagnósticos realizados al paciente, *id_unidad* de la tabla *cfg_unidad* que brinda los datos correspondiente a cada unidad de salud, el *id_servicio* de la tabla *cfg_servicio* que indica cada uno de los servicios que se le pueden prestar a cada paciente como son cirugía del cristalino, terigium, cirugía refractiva, miopía entre otros.

Con el Módulo de Farmacia también se interactúa, en el momento de seleccionar los medicamentos y los lentes intraoculares del paciente a partir del *id_producto* de la tabla *fa_producto*.

De esta manera, se puede llegar a la conclusión parcial que la estrategia de integración de la presente solución con los demás módulos del sistema, se lleva a cabo a nivel de negocio y aunque es reflejada a nivel de base de datos con el almacenamiento de varias llaves foráneas, es el negocio el encargado de garantizar la integridad referencial de dichos datos así como la correcta inserción, actualización o eliminación de estos.

2.2 Descripción y fundamentación de la arquitectura.

Una de los aspectos más importantes a tomar en cuenta a la hora de desarrollar un sistema de almacenamiento de la información en forma de bases de datos, es el deseo de integrar la información y

Capítulo 2. Descripción y análisis de la solución propuesta.

proporcionar un acceso rápido y controlado a la misma. Para lograrlo es importante regirse por una serie de normas que se complementan en la arquitectura de dicha base de datos.

Esta es influenciada en gran medida por el sistema informático en el que se ejecuta el sistema. Existen varias formas de diseñar una base de datos, las que se modelan en dos tipos de categorías fundamentales: las bases de datos distribuidas y las bases de datos centralizadas.

Como decisión de los arquitectos del proyecto, tomando en cuenta toda una serie de requisitos como los antes mencionados se determinó utilizar una arquitectura de bases de datos centralizada. Posibilitando que desde un servidor central se procesen los datos de todos los módulos a integrarse posteriormente en el proyecto, lo que propiciaría respuesta a cada una de las transacciones con un menor costo de tiempo.

La arquitectura de bases de datos centralizada es la más clásica. En ella, el Sistema de Gestión de la Base de Datos está implantado en una sola plataforma o servidor desde donde se gestiona directamente, de modo centralizado, la totalidad de los recursos. Es la arquitectura de los centros de proceso de datos tradicionales. Se basa en tecnologías sencillas, muy experimentadas y de gran robustez. Nacen en torno a una concepción tradicional de la organización, con estructura centralizada y jerárquica. Entre sus principales características funcionales se pueden encontrar que el servidor central es el encargado del almacenamiento de la información, este contiene todos los datos y es el responsable de la consolidación de la información, desde el servidor central se controla el acceso a múltiples terminales conectados a través de productos integrados en la arquitectura de red. [28]

2.3 Análisis de optimización de queries.

Una consulta o query es una forma de buscar, encontrar y exhibir determinada información, extrayéndola del cúmulo de datos que almacena la base de datos, es por eso que resulta de vital importancia la optimización del tiempo de respuesta de la misma, ya que esto influye directamente en la optimización del Sistema de Base de Datos.

Cuando se habla de optimización de consultas se hace referencia a mejorar los tiempos de respuesta en un Sistema de Gestión de Bases de Datos Relacional, pues la optimización es el proceso de modificar un sistema para mejorar su eficiencia y el uso de los recursos disponibles.

Capítulo 2. Descripción y análisis de la solución propuesta.

En bases de datos relacionales el lenguaje de consultas SQL es el más utilizado por los programadores y desarrolladores para obtener información. La complejidad que pueden alcanzar algunas consultas puede ser tal, que el diseño de ellas puede tomar un tiempo considerable, obteniendo no siempre una respuesta óptima. El lenguaje SQL es no procedimental, es decir, en las sentencias se indica que se quiere conseguir y no como lo tiene que hacer el intérprete para conseguirlo.

En la práctica, todo gestor de base de datos tiene sus propias especificidades para optimizar el rendimiento, por tanto, muchas veces no basta con especificar una consulta correcta, sino que además, hay que indicarle al Sistema Gestor de Base de Datos como tiene que hacerlo si se quiere que el tiempo de respuesta sea mínimo.

En todo Sistema de Gestión de Bases de Datos es absolutamente necesario garantizar un buen rendimiento en el acceso a los datos y en esto incide directamente el tiempo de respuesta de cada uno de los queries. [29]

Una base de datos se diseña a nivel físico para lograr un buen rendimiento de sus aplicaciones, sin embargo, el esquema físico producto de este proceso de diseño es apenas una determinación inicial de estructuras y caminos de acceso. Luego, con la operación de las aplicaciones sobre la base de datos, se deben medir los tiempos de respuesta y throughput de las transacciones, si estas mediciones no están dentro de los rangos esperados es necesario llevar a cabo un proceso de optimización de la base de datos.

Se define entonces el proceso de optimización de Sistemas de Base de Datos como: el proceso de realización de ajustes en el contexto o en la base de datos propiamente dicha para lograr que la misma opere con un buen rendimiento.

Según Dennis Shasha [30] existen cuatro principios básicos para la optimización de los Sistemas de Bases de Datos:

Pensar globalmente, arreglar localmente. Para ser efectivos en la tarea de optimización se debe identificar el problema adecuadamente y hacer una intervención mínima en el sistema.

El particionamiento elimina los cuellos de botella. Rara vez un sistema es lento porque todos sus recursos estén saturados.

Capítulo 2. Descripción y análisis de la solución propuesta.

El inicio es costoso, continuar es más barato. Al comenzar a leer datos de un disco, la ubicación del primer bloque es lo más costoso pues involucra ubicar ese primer bloque y mover todas las piezas a ese lugar. Una vez ubicado el primer bloque, leer los siguientes es menos costoso, entonces se debe tratar de hacer esa ubicación inicial la menor cantidad de veces posible.

Al servidor lo que es del servidor. Decisiones de donde realizar cada tarea, en el servidor o en el programa de aplicación. Además de las decisiones de los ajustes que es necesario realizar para que porciones de la base de datos operen con un mejor rendimiento, un aspecto fundamental de la tarea de optimización es diseñar desde el principio donde se van a ejecutar las diferentes tareas relativas a la base de datos (el servidor) y al programa de aplicación (el cliente). [31]

Los Sistemas de Gestión de Bases de Datos trabajan sobre una plataforma de hardware y en estrecha interacción con el sistema operativo sobre el cual funciona la plataforma. En primera instancia, hay aspectos del Hardware y del sistema operativo que afectan el rendimiento en la base de datos, de ahí que este aspecto es importante tenerlo en cuenta a la hora de determinar los requerimientos no funcionales del sistema (Ver epígrafe siguiente) también proveen servicios que tienen un impacto directo en el rendimiento, pues involucran un recargo administrativo, un overhead, como ejemplo de estos se puede ver el mecanismo de control de concurrencias del gestor de base de datos.

El trabajo sobre la base de datos se divide en transacciones. Las propiedades de consistencia y aislamiento de las transacciones están pensadas para evitar que la ejecución concurrente de las transacciones cause problemas de inconsistencias en los datos. El control de concurrencia influye mucho sobre el rendimiento del sistema, pero es delicado el compromiso entre el rendimiento y la integridad.

La optimización del rendimiento del Sistema de Almacenamiento de la información para el Bloque Quirúrgico Oftalmológico se concentró en la optimización del diseño físico y la optimización de las consultas o queries SQL.

El rendimiento de una base de datos se mide, entre otros aspectos, por los tiempos de respuesta de sus transacciones y por el throughput, que es el número de transacciones por unidad de tiempo. Las consultas tienen una variedad de forma y espacio para mejorar el rendimiento. Por lo tanto conviene distinguir los tipos que se pueden escribir.

La tipificación propuesta por Shasha [32] consiste de 8 clases de consultas, las cuales se describen como:

Capítulo 2. Descripción y análisis de la solución propuesta.

Consulta puntual (“point query”). La condición de búsqueda es igualdad y la consulta devuelve máximo una tupla.

Consulta múltiple (“multipoint query”). La condición de búsqueda es igualdad y la consulta devuelve varias tuplas.

Consulta por rango (“range query”). El rango se especifica sobre los valores de un atributo definido en un dominio ordenado, la consulta devuelve todas las tuplas cuyo valor del atributo de la condición está dentro del rango especificado.

Consulta de prefijo (“prefix match query”). Se expresa sobre un atributo o secuencia de atributos, la condición de búsqueda especifica sólo el prefijo del valor buscado.

Consulta extrema (“extremal query”). Devuelve las tuplas cuyo valor en uno o más atributos es un mínimo o un máximo.

Consulta de ordenamiento (“ordering query”). Es aquella consulta que incluye una cláusula de ordenamiento de las tuplas resultantes por el orden del valor de un atributo en particular.

Consulta de agrupación (“grouping query”). Es aquella consulta cuyas tuplas resultantes se agrupan de acuerdo al valor de uno o más atributos, generalmente, a cada grupo se le aplica una función agregada.

Consulta con Join (“join query”). Es aquella consulta que asocia dos o más relaciones a través de atributos comunes. La condición sobre la cual se asocian las tablas puede usar la igualdad o algún otro comparador.

La reescritura de una consulta, para que se ejecute más rápidamente es una acción que sólo puede traer buenos resultados y el precio a pagar es solamente la inversión de tiempo del desarrollador en reescribirla.

Antes de considerar si una consulta debe ser reescrita para mejorar su rendimiento es necesario analizar si la consulta se está ejecutando muy lentamente. Dos indicadores de ello son: La consulta hace demasiados accesos a disco, por ejemplo, siendo una consulta puntual recorre toda la tabla o que al revisar el plan de ejecución se observa que hay índices relevantes que no se utilizan.

Capítulo 2. Descripción y análisis de la solución propuesta.

Existen aspectos específicos que se han tenido en cuenta para mejorar el rendimiento de los queries [33], estos contribuirán directamente al buen rendimiento que de forma general debe tener la presente base de datos, (ver Anexo 3) estos son:

- La cláusula DISTINCT es costosa de ejecutar porque generalmente involucra un ordenamiento de las tuplas resultantes para eliminar duplicados, por esto se ha realizado un análisis a partir del cual se han encontrado vías de soluciones en las que no ha sido necesario utilizarlas.
- Las subconsultas en muchos manejadores se ejecutan ineficientemente. Una razón para ello es que al estar anidada no se utiliza algún índice relevante, al eliminar la subconsulta es posible lograr que el manejador utilice el índice apropiado. Por esto, las consultas del presente sistema utilizan los índices de cada tabla siempre que es posible, sólo se cuenta con subconsultas anidadas en caso de estos no existir, como es el caso de la inserción de un nuevo número de consulta.
- En algunas consultas el usuario almacena resultados intermedios explícitamente en tablas temporales. Estas tablas pueden bajar el rendimiento de la consulta por dos razones, una porque fuerza un orden de ejecución que quizás no sea ideal y segundo, porque obliga a ejecutar actualizaciones al diccionario de datos lo cual tiene el peligro de convertirlo en un cuello de botella. Por otro lado, las tablas temporales pueden tener un efecto positivo al reescribir consultas que contienen subconsultas correlacionadas complejas. Las subconsultas correlacionadas pueden ejecutarse ineficientemente, pero si se logra calcular primero las tuplas de la subconsulta y almacenarlas en una tabla temporal, se puede reescribir la consulta original, sin la subconsulta, accediendo a la tabla temporal.
- Las condiciones del Join se pueden evaluar más eficientemente contra un índice primario y en general, en términos de rendimiento es preferible evaluar una condición de igualdad numérica que una condición de igualdad sobre cadenas ("strings") de caracteres. Esto ha sido también tomado en cuenta, realizando Joins y vistas por los índices numéricos principales de las tablas.
- Es preferible no usar la cláusula HAVING si la condición deseada se puede expresar en la cláusula WHERE. En el presente sistema no se tubo la necesidad de utilizar dicha clausula HAVING ya que con la WHERE se resolvieron todas las necesidades del sistema.

Capítulo 2. Descripción y análisis de la solución propuesta.

- Otra particularidad importante del manejador que es bueno conocer es si el orden de las tablas en la cláusula FROM puede afectar la implementación del join utilizada.
- El uso de vistas pueden causar una ejecución ineficiente de consultas. Muchas veces la ejecución de consultas sobre las tablas base es más eficiente. En este caso, la mayor parte de las vistas generadas por el presente sistema, tienen que ver con el retorno de datos de PostgreSQL, ya que en este lenguaje sólo se puede retornar tuplas de una tabla específica y no hacerlo de forma directa. En el presente no se generaron vistas para consultas a partir de búsquedas directamente en ellas.
- La correcta gestión y utilización de los índices. Estos son campos elegidos arbitrariamente por el constructor de la base de datos que permiten la búsqueda a partir de dicho campo a una velocidad notablemente superior. Sin embargo, esta ventaja se ve contrarrestada por el hecho de ocupar mucha más memoria y de requerir para su inserción y actualización un tiempo de proceso superior.

Evidentemente, no se pueden indexar todos los campos de una tabla extensa ya que se doblaría el tamaño de la base de datos. Igualmente, tampoco sirve de mucho el indexar todos los campos en una tabla pequeña ya que las selecciones pueden efectuarse rápidamente de todos modos. Un caso en el que los índices pueden resultar muy útiles es cuando se realizan peticiones simultáneas sobre varias tablas. En este caso, el proceso de selección puede acelerarse sensiblemente si se indexan los campos que sirven de nexo entre las dos tablas. El presente sistema tiene índices por cada relación existente entre tablas, lo que facilita y agiliza las consultas donde se utilizan múltiples tablas.
- No utilizar nunca `SELECT *` porque el gestor debe leer primero la estructura de la tabla antes de ejecutar la sentencia, a pesar de que este recurso es mucho más cómodo y simple para el implementador de las consultas puesto que agiliza el tiempo del trabajo; a la hora de realizar los queries se ha especificado que campos de la base de datos se necesitan para ahorrar tiempo de ejecución.
- Al utilizar varias tablas en la consulta se especifica siempre a qué tabla pertenece cada campo, esto le ahorra al gestor el tiempo de localizar a qué tabla pertenece el campo.

2.4 Selección y argumentación de los requisitos funcionales y no funcionales.

El sistema debe ser capaz de cumplir con una serie de requisitos tanto funcionales como no funcionales.

Un requisito es una característica, propiedad o comportamiento que se desea para el sistema.

El término requisito es una palabra del lenguaje natural que se corresponde con una gama de estructuras del Lenguaje Unificado del Modelado (UML) destinadas a especificar las características deseadas para un sistema. Los requisitos funcionales, correspondientes a transacciones visibles por parte del usuario suelen capturar como casos de uso. Los requisitos no funcionales, como las métricas de rendimiento y de calidad, se pueden capturar en forma de enunciados de texto que eventualmente llegarán a los elementos del diseño final, también los comentarios y restricciones de UML se pueden emplear para representar requisitos no funcionales. [34]

De esta manera, el Sistema de Almacenamiento de la información de Bloque Quirúrgico Oftalmológico, que parte de la interrelación con el Bloque Quirúrgico General, deberá permitir al usuario los siguientes requisitos funcionales:

R 1 Gestionar Anuncio Operatorio Oftalmológico.

R 2 Cambiar estado del Anuncio Operatorio Oftalmológico.

R2.1 Suspender Anuncio Operatorio.

R2.2 Registrar causa de suspensión.

R2.3 Posponer Anuncio Operatorio.

R 3 Gestionar Hoja del Especialista Oftalmológico.

R 4 Gestionar de Informe Operatorio Oftalmológico.

R 5 Gestionar Acto Quirúrgico.

R5.1 Registrar datos del paciente en el pre_operatorio.

R5.2 Registrar datos del paciente en el trans_operatorio.

R5.3 Registrar datos del paciente en el post_operatorio

R 6 Buscar Anuncio Operatorio Oftalmológico.

Capítulo 2. Descripción y análisis de la solución propuesta.

R 7 Buscar Consultas Oftalmológicas

Para cumplir correctamente con todos estos requisitos funcionales, el sistema debe también cumplir una serie de requisitos no funcionales, a la hora de identificar estos requisitos, es necesario tomar en cuenta el hecho de que la selección de estos influye directamente en el posterior funcionamiento del Sistema de Gestión de Base de Datos, por lo que el tiempo de respuesta de la misma, va a estar supeditado en última instancia a varios de estos requisitos, específicamente los requerimientos de hardware.

Los requerimientos no funcionales del presente sistema son:

- 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 brindarse 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 rápido permitiendo la toma de decisiones en tiempo real.

- Requerimientos de soporte:

Capítulo 2. Descripción y análisis de la solución propuesta.

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

- 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), 1GHz, 10Gb HDD.

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 98 o superior y sistemas Unix, Linux. Para sistemas Windows se debe tener instalado Microsoft Framework 2.0 y en sistemas Linux la plataforma Mono 1.2 o superior.

Capítulo 2. Descripción y análisis de la solución propuesta.

- Restricciones en el diseño y la implementación:

El sistema será implementado utilizando como lenguaje de programación del lado del cliente java script auxiliándose de la tecnología AJAX.

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

Se utilizará además un grupo de bibliotecas de clases definidas, dentro de ellas se encuentran Hermes7 para la comunicación HL7 y Npgsql para la conexión al servidor de bases de datos.

El servidor debe contar con una plataforma de funcionamiento que utilice PostgreSQL como gestor de base de datos. La comunicación de de las terminales clientes con el servidor será a través de conexiones de fibra óptica.

Estos requisitos influyen directamente sobre el sistema de base de datos desarrollado, el cual da cumplimiento a todos y cada uno de ellos.

2.5 Diagrama de clases persistentes obtenido a partir del diagrama de clases del diseño.

A partir del diagrama de clases que fue presentado por el analista del presente módulo se generó el diagrama de clases persistentes correspondiente al Sistema de Almacenamiento de la Información del Bloque Quirúrgico Oftalmológico.

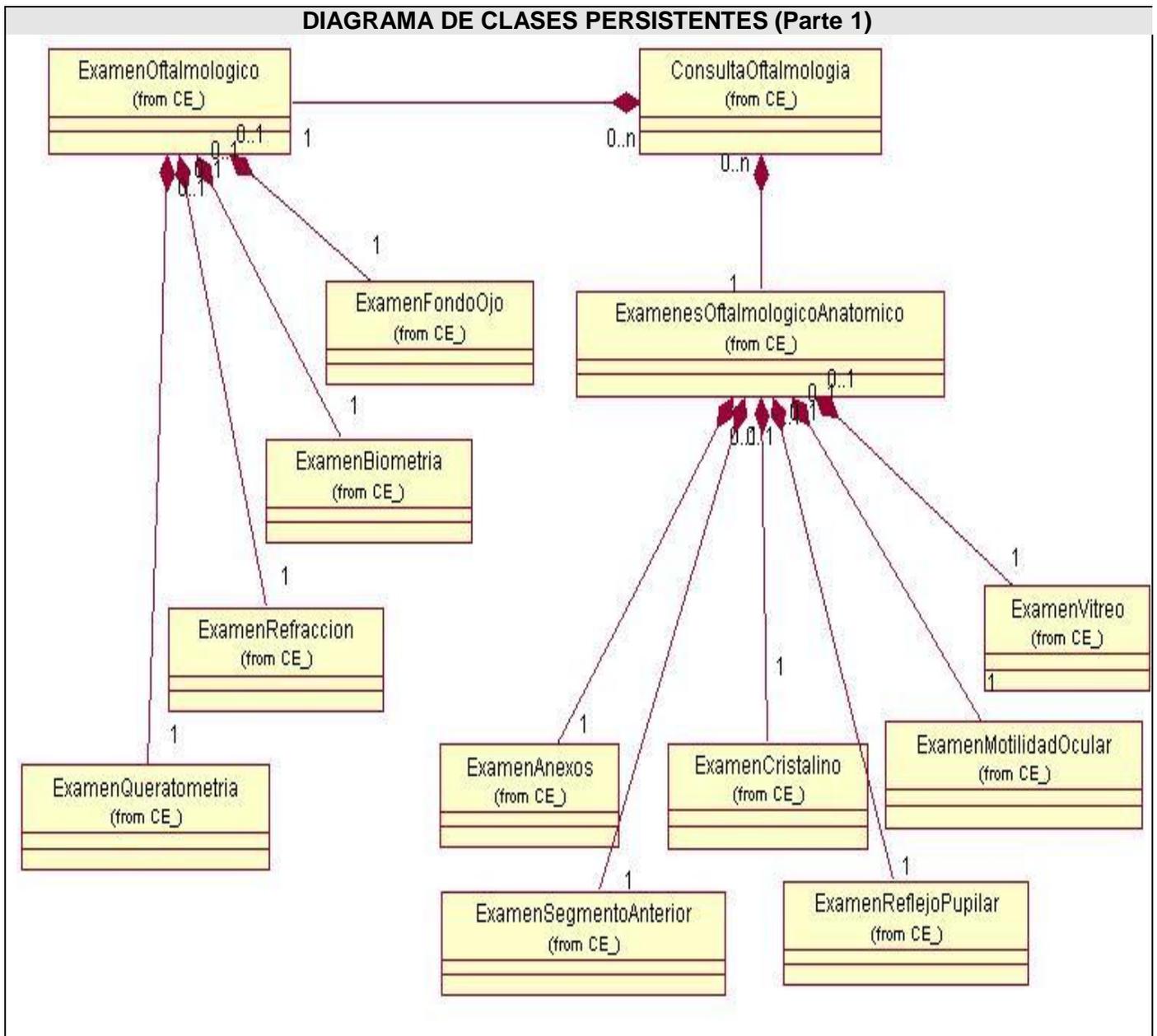


Figura 2.1 Diagrama de clases persistentes (parte 1).

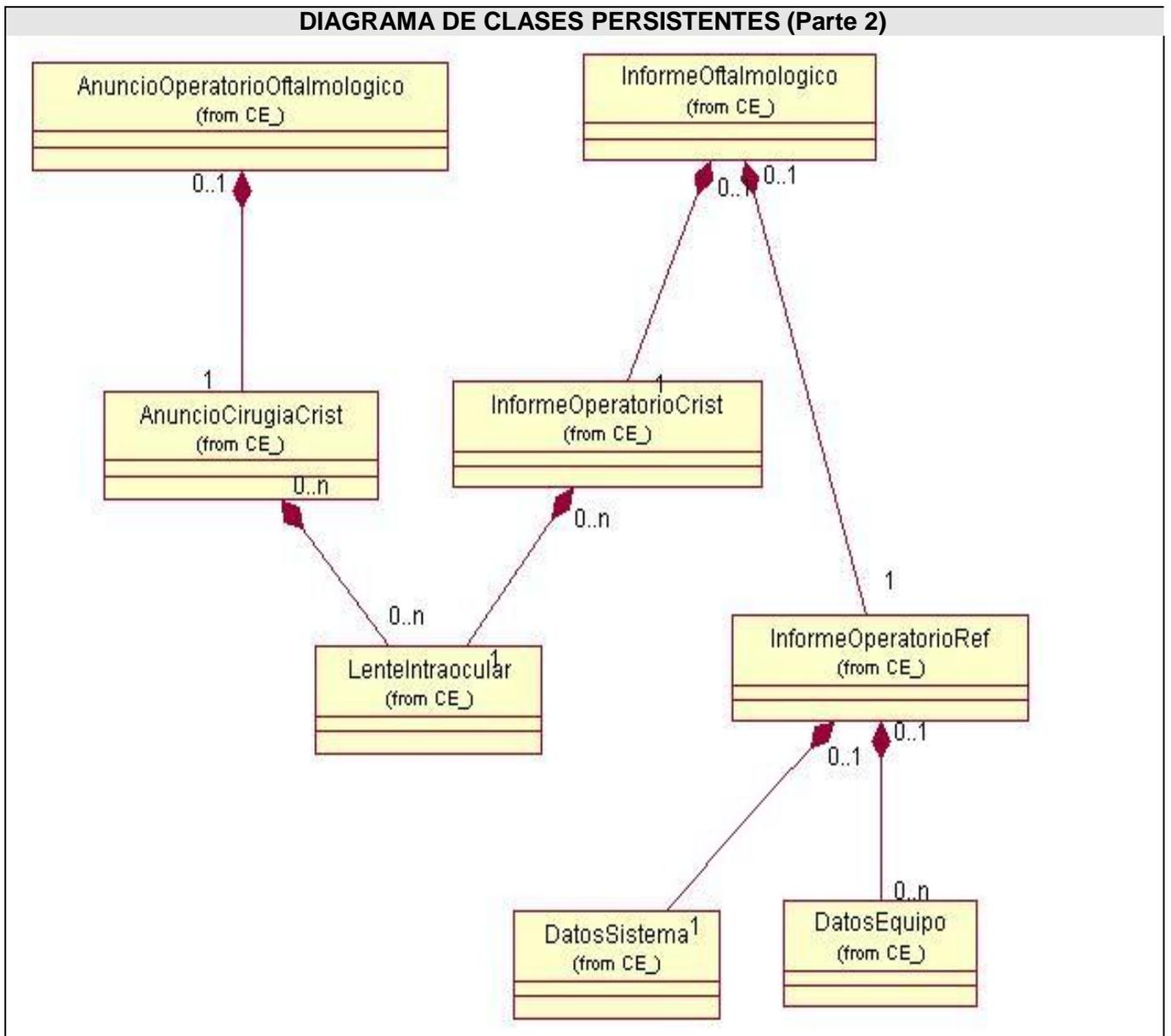


Figura 2.2 Diagrama de clases persistentes (parte 2).

2.6 Descripción de las clases.

Las principales clases que conforman el diagrama de clases persistente son:

Nombre: ExamenRefraccion	
Tipo de clase: Entidad	
Atributo	Tipo
idPersona	Int32
numeroConsulta	Int32
agudezaVisualSinCorreccion	Double
esfera	Double
cilindro	Double
eje	Double
agudezaVisualConCorreccion	Double
Responsabilidades:	
Nombre:	Refraccion
Descripción:	Constructor de la clase.
Nombre:	IdPersona
Descripción:	Propiedad para mostrar y modificar el idPersona
Nombre:	NumeroConsulta
Descripción:	Propiedad para mostrar y modificar el numeroConsulta
Nombre:	AgudezaVisualSinCorreccion
Descripción:	Propiedad para mostrar y modificar la agudezaVisualSinCorreccion
Nombre:	Esfera
Descripción:	Propiedad para mostrar y modificar la esfera
Nombre:	Cilindro
Descripción:	Propiedad para mostrar y modificar el cilindro
Nombre:	Eje

Capítulo 2. Descripción y análisis de la solución propuesta.

Descripción:	Propiedad para mostrar y modificar el eje
Nombre:	AgudezaVisualConCorreccion
Descripción:	Propiedad para mostrar y modificar la agudezaVisualConCorreccion

Tabla 2.1 Descripción de la clase ExamenRefraccion.

Nombre: ExamenTensionOcular	
Tipo de clase: Entidad	
Atributo	Tipo
idPersona	Int32
numeroConsulta	Int32
elastotonometria	Double
radioCurvatura	Double
hipertensionArterial	Boolean
Responsabilidades:	
Nombre:	TensionOcular
Descripción:	Constructor de la clase.
Nombre:	IdPersona
Descripción:	Propiedad para mostrar y modificar el idPersona
Nombre:	NumeroConsulta
Descripción:	Propiedad para mostrar y modificar el numeroConsulta
Nombre:	Elastotonometria
Descripción:	Propiedad para mostrar y modificar la elastotonometria
Nombre:	RadioCurvatura
Descripción:	Propiedad para mostrar y modificar el radioCurvatura
Nombre:	HipertensionArterial
Descripción:	Propiedad para mostrar y modificar elhipertensionArterial

Tabla 2.2 Descripción de la clase ExamenTensionOcular.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: ExamenQueratometría	
Tipo de clase: Entidad	
Atributo	Tipo
idPersona	Int32
numeroConsulta	Int32
meridianoDebil	Double
meridianoFuerete	Double
ejeFuerte	Double
ejeDebil	Double
Responsabilidades:	
Nombre:	Queratometría
Descripción:	Constructor de la clase.
Nombre:	IdPersona
Descripción:	Propiedad para mostrar y modificar el idPersona
Nombre:	NumeroConsulta
Descripción:	Propiedad para mostrar y modificar el numeroConsulta
Nombre:	MeridianoDebil
Descripción:	Propiedad para mostrar y modificar el meridianoDebil
Nombre:	MeridianoFuerete
Descripción:	Propiedad para mostrar y modificar el meridianoFuerete
Nombre:	EjeFuerte
Descripción:	Propiedad para mostrar y modificar el ejeFuerte
Nombre:	EjeDebil
Descripción:	Propiedad para mostrar y modificar el ejeDebil

Tabla 2.3 Descripción de la clase ExamenQueratometría.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: LenteIntraocular	
Tipo de clase: Entidad	
Atributo	Tipo
idLio	Int32
idTipoLente	Int32
poderDioptrico	Double
constanteFabricante	String
modeloLente	String
Para cada responsabilidad:	
Nombre:	LenteIntraocular
Descripción:	Constructor de la clase.
Nombre:	IdLio
Descripción:	Propiedad para mostrar y modificar el idLio
Nombre:	IdTipoLente
Descripción:	Propiedad para mostrar y modificar el idTipoLente
Nombre:	PoderDioptrico
Descripción:	Propiedad para mostrar y modificar el poderDioptrico
Nombre:	ConstanteFabricante
Descripción:	Propiedad para mostrar y modificar la constanteFabricante
Nombre:	ModeloLente
Descripción:	Propiedad para mostrar y modificar el modeloLente

Tabla 2.4 Descripción de la clase LenteIntraocular.

Nombre: ExamenOftalmologicosAnatomia	
Tipo de clase: Entidad	
Atributo	Tipo
idAnatomiaDelOjo	Int32
idPadecimientoOftalmologico	Int32
Para cada responsabilidad:	

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre:	ExamenPadecimiento
Descripción:	Constructor de la clase.
Nombre:	idAnatomiaDelOjo
Descripción:	Propiedad para mostrar y modificar el idAnatomiaDelOjo
Nombre:	idPadecimientoOftalmologico
Descripción:	Propiedad para mostrar y modificar el idPadecimientoOftalmologico

Tabla 2.5 Descripción de la clase ExamenPadecimiento.

Nombre: ExamenBiometria	
Tipo de clase: Entidad	
Atributo	Tipo
idPersona	Int32
numeroConsulta	Int32
profundidadCamaraAnterior	Double
grosarioDelCristalino	Double
longitudAccial	Double
Para cada responsabilidad:	
Nombre:	Biometria
Descripción:	Constructor de la clase.
Nombre:	idPersona
Descripción:	Propiedad para mostrar y modificar el idPersona
Nombre:	NumeroConsulta
Descripción:	Propiedad para mostrar y modificar el numeroConsulta
Nombre:	ProfundidadCamaraAnterior
Descripción:	Propiedad para mostrar y modificar la profundidadCamaraAnterior
Nombre:	GrosarioDelCristalino
Descripción:	Propiedad para mostrar y modificar el grosarioDelCristalino
Nombre:	LongitudAccial
Descripción:	Propiedad para mostrar y modificar la longitudAccial

Tabla 2.6 Descripción de la clase ExamenBiometria.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: ExamenAnatomia	
Tipo de clase: Entidad	
Atributo	Tipo
idAnatomiaDelOjo	Int32
idPadecimientoOftalmologico	Int32
NumeroConsulta	Int32
idPersona	Int32
idOjos	Int32
Para cada responsabilidad:	
Nombre:	ExamenAnatomia
Descripción:	Constructor de la clase.
Nombre:	IdAnatomiaDelOjo
Descripción:	Propiedad para mostrar y modificar el idAnatomiaDelOjo
Nombre:	IdPadecimientoOftalmologico
Descripción:	Propiedad para mostrar y modificar el idPadecimientoOftalmologico
Nombre:	NumeroConsulta
Descripción:	Propiedad para mostrar y modificar el numeroConsulta
Nombre:	IdPersona
Descripción:	Propiedad para mostrar y modificar el idPersona
Nombre:	IdOjos
Descripción:	Propiedad para mostrar y modificar el idOjos

Tabla 2.7 Descripción de la clase ExamenAnatomia.

Nombre: ExámenesOftalmologicos	
Tipo de clase: Entidad	
Atributo	Tipo
NumeroConsulta	Int32

Capítulo 2. Descripción y análisis de la solución propuesta.

idPersona	Int32
idOjos	Int32
iolMaster	Boolean
Para cada responsabilidad:	
Nombre:	ExamenesOftalmologicos
Descripción:	Constructor de la clase.
Nombre:	NumeroConsulta
Descripción:	Propiedad para mostrar y modificar el numeroConsulta
Nombre:	IdPersona
Descripción:	Propiedad para mostrar y modificar el idPersona
Nombre:	IdOjos
Descripción:	Propiedad para mostrar y modificar el idOjos
Nombre:	iolMaster
Descripción:	Propiedad para mostrar y modificar el iolMaster

Tabla 2.8 Descripción de la clase ExamenesOftalmologicos.

Nombre: AnuncioCirugiaCrist	
Tipo de clase: Entidad	
Atributo	Tipo
idAnuncioOperatorio	Int32
idFormulasDioptrias	Int32
idLugarColocacion	Int32
idOjos	Int32
afaquia	Boolean
causasNoImplante	String
refraccionEsperada	Double
Para cada responsabilidad:	
Nombre:	AnuncioCirugiaCrist
Descripción:	Constructor de la clase.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre:	IdAnuncioOperatorio
Descripción:	Propiedad para mostrar y modificar el idAnuncioOperatorio
Nombre:	IdFormulasDioptrias
Descripción:	Propiedad para mostrar y modificar el idFormulasDioptrias
Nombre:	IdLugarColocacion
Descripción:	Propiedad para mostrar y modificar el idLugarColocacion
Nombre:	IdOjos
Descripción:	Propiedad para mostrar y modificar el idOjos
Nombre:	Afaquia
Descripción:	Propiedad para mostrar y modificar la afaquia
Nombre:	CausasNoImplante
Descripción:	Propiedad para mostrar y modificar las causasNoImplante
Nombre:	RefraccionEsperada
Descripción:	Propiedad para mostrar y modificar la refraccionEsperada

Tabla 2.9 Descripción de la clase AnuncioCirugiaCrist.

Nombre: InformeOperatorioCristalino	
Tipo de clase: Entidad	
Atributo	Tipo
idInformeOperatorio	Int32
idFormeExtraccion	Int32
idIridectomia	Int32
idLugarColocacion	Int32
refraccionEsperada	Double
ipertensionAretrialSecundaria	Boolean
afaquia	Boolean
especificacionNoLente	String
tiempoFaco	DateTime
idOjos	Int32

Capítulo 2. Descripción y análisis de la solución propuesta.

Para cada responsabilidad:	
Nombre:	InformeOperatorioCristalino
Descripción:	Constructor de la clase.
Nombre:	IdInformeOperatorio
Descripción:	Propiedad para mostrar y modificar el idInformeOperatorio
Nombre:	IdFormeExtraccion
Descripción:	Propiedad para mostrar y modificar el idFormeExtraccion
Nombre:	IdIridectomia
Descripción:	Propiedad para mostrar y modificar el idIridectomia
Nombre:	IdLugarColocacion
Descripción:	Propiedad para mostrar y modificar el idLugarColocacion
Nombre:	RefraccionEsperada
Descripción:	Propiedad para mostrar y modificar la refraccionEsperada
Nombre:	IpertensionAretrialSecundaria
Descripción:	Propiedad para mostrar y modificar la ipertensionAretrialSecundaria
Nombre:	Afaquia
Descripción:	Propiedad para mostrar y modificar la afaquia
Nombre:	EspecificacionNoLente
Descripción:	Propiedad para mostrar y modificar la especificacionNoLente
Nombre:	TiempoFaco
Descripción:	Propiedad para mostrar y modificar el tiempoFaco
Nombre:	IdOjos
Descripción:	Propiedad para mostrar y modificar el idOjos

Tabla 2.10 Descripción de la clase InformeOperatorioCristalino.

Nombre: AnuncioOperatorioOftalmologico	
Tipo de clase: Entidad	
Atributo	Tipo
anuncioCirugiaCrist	AnuncioCirugiaCrist
listLenteAnuncio	List<LenteAnuncio>
Para cada responsabilidad:	
Nombre:	AnuncioOperatorioOftalmologico
Descripción:	Constructor de la clase.
Nombre:	anuncioCirugiaCrist
Descripción:	Propiedad para mostrar y modificar el anuncioCirugiaCrist
Nombre:	listLenteAnuncio
Descripción:	Propiedad para mostrar y modificar la listLenteAnuncio

Tabla 2.11 Descripción de la clase AnuncioOperatorioOftalmologico.

2.7 Diseño de la base de datos.

A partir del diagrama de clases persistentes se realizaron una serie de transformaciones para conformar el Diagrama Entidad Relación, estas son:

Procesos de normalización del diseño de la base de datos. Al normalizar la base de datos, se crean nuevas tablas como resultado de la aplicación de las reglas de normalización, con lo que se logra un diseño físico más óptimo.

Selección de tablas de configuración para la homogenización del sistema. Para lograr la homogenización de la base de datos se seleccionan y se crean una serie de tablas de configuración a partir de los atributos nomencladores de las tablas del sistema. Esto facilita la posterior configuración del mismo.

En el presente sistema se pueden ver ejemplos de estos como son: las tablas `cfg_ojo` para clasificar los ojos a intervenir quirúrgicamente, `cfg_antecedente_ofthalmologico`, para clasificar todos y cada uno de los

antecedentes patológicos oftalmológicos necesarios en la consulta especializada y `cfg_tipo_cirugia` para clasificar cada una de las cirugías a realizar, entre otras.

Determinación de dependencias entre tablas. La dependencia entre las tablas no se puede modelar igual en un Diagrama de Clases que en un Diagrama Entidad Relación, este es el caso de las herencias y las relaciones de inclusión cuyos modelados difieren en ambos diagramas y la generación de tablas intermedias a partir de la relación de muchos a muchos entre dos o más entidades, que genera una nueva tabla en el Diagrama Entidad Relación y esto no es así en un diagrama de clases.

De esta forma queda conformado el diseño físico del Sistema de Almacenamiento de la Información correspondiente al Bloque Quirúrgico Oftalmológico.

2.7.1 Diagrama entidad relación de la base de datos.

El Modelo Entidad Relación se ha convertido en un estándar a la hora de modelar bases de datos relacionales, es un concepto de modelado para bases de datos mediante el cual se visualizan los objetos que pertenecen a esta como entidades las cuales tienen atributos y se vinculan mediante relaciones.

Es una representación lógica y completa de la información que logra representar el esquema físico de la base de datos; por lo que a continuación se presenta este modelo y posteriormente se describe su implementación a partir de la descripción detallada de cada una de las tablas que lo componen.

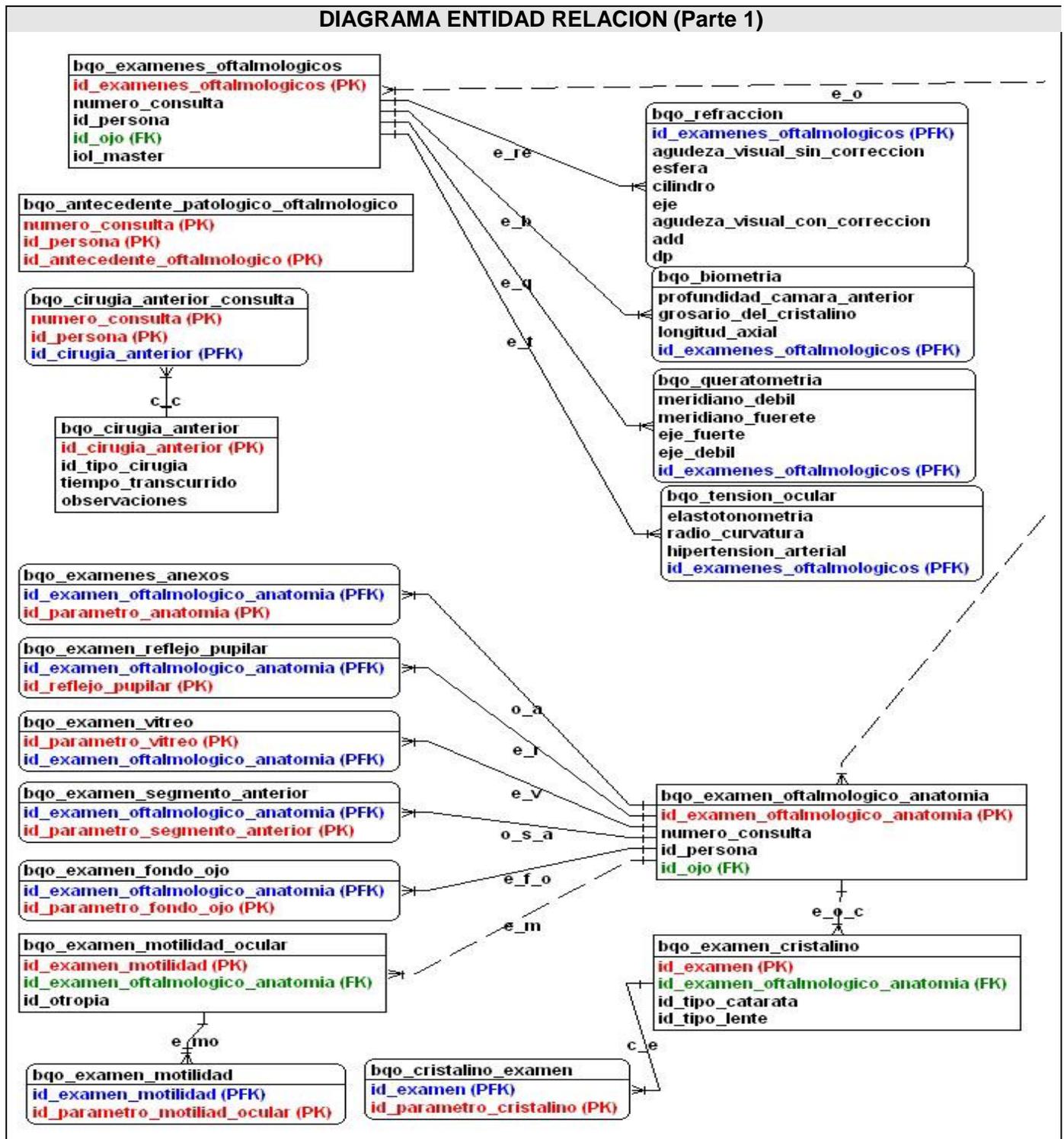


Figura 2.3 Diagrama Entidad Relación (parte 1)

Capítulo 2. Descripción y análisis de la solución propuesta.

2.7.2 Descripción de las tablas.

Para su mejor comprensión se realiza la descripción detallada de cada una de las tablas del Sistema de Almacenamiento de la Información del Bloque Quirúrgico Oftalmológico, así como cada uno de sus campos y sus atributos.

Nombre: bqo_antecedentes_patologicos Oftalmologicos		
Descripción: Guarda la lista de antecedentes patológicos oftalmológicos de la consulta de oftalmología.		
Atributo	Tipo	Descripción
id_persona	interger	Llave foránea proveniente de la tabla bqg_datos_consultas del módulo BQG para identificar a qué consulta pertenece el examen.
numero_consulta	interger	Llave foránea proveniente de la tabla bqg_datos_consultas del módulo BQG para identificar a qué consulta pertenece el examen.
id_antecedentes Oftalmologicos	interger	Llave foránea proveniente de la tabla cfg_antecedentes Oftalmologicos que identifica el antecedente patológico oftalmológico que se adiciona en la consulta.

Tabla 2.12 Descripción de la tabla bqo_antecedentes_patologicos_Oftalmologicos.

Nombre: bqo_anuncio_cirugia_crist		
Descripción: Almacena los datos del anuncio operatorio del subservicio cristalino		
Atributo	Tipo	Descripción
id_anuncio_operatorio	interger	Llave foránea proveniente de la tabla bqo_anuncio_Oftalmologico, para identificar a que anuncio operatorio oftalmológico pertenece el anuncio del cristalino.
id_formula_dioptria	interger	Llave foránea proveniente de la tabla cfg_formula_dioptria que identifica la fórmula utilizada para el cálculo de las dioptrías del lente propuesto.
afaquia	boolean	Identifica si la operación va a ser afaquica o no.

Capítulo 2. Descripción y análisis de la solución propuesta.

id_lugar_colocacion_cristalino	integer	Llave foránea proveniente de la tabla cfg_lugar_colocacion_cristalino que identifica el lugar de colocación del lente.
causas_no_implante	varchar	Describe las causas de no implante del lente en caso de ocurrir.

Tabla 2.13 Descripción de la tabla bqo_anuncio_cirugia_crist.

Nombre: bqo_anuncio Oftalmológico		
Descripción: Almacena los datos generales correspondientes al anuncio operatorio oftalmológico realizado.		
Atributo	Tipo	Descripción
id_anuncio_operatorio	integer	Llave foránea proveniente de la tabla bqg_anuncio_operatorio del módulo BQG para identificar a que anuncio operatorio general pertenece el especializado.
id_ojo	integer	Llave foránea proveniente de la tabla cfg_ojo que identifica el ojo a realizar el anuncio operatorio.
refracción_esperada	double precision	Guarda el valor de la refracción esperada luego de la operación.
id_tipo_anuncio_quirurgico	integer	Llave foránea proveniente de la tabla cfg_tipo_anuncio_operatorio que identifica de que tipo de anuncio operatorio oftalmológico específico se almacena los datos.

Tabla 2.14 Descripción de la tabla bqo_anuncio_ofthalmologico.

Nombre: bqo_biometria		
Descripción: Almacena el examen de biometría.		
Atributo	Tipo	Descripción
id_examenoftalmologico	integer	Llave foránea identificativa que especifica a que examen oftalmológico pertenece la biometría
id_persona	integer	Llave foránea proveniente de la tabla bqg_datos_consultas del módulo Bloque Quirúrgico General para identificar a que examen oftalmológico general pertenece este examen.

Capítulo 2. Descripción y análisis de la solución propuesta.

numero_consulta	integer	Llave foránea proveniente de la tabla bqq_datos_consultas del módulo BQG para identificar a que examen oftalmológico general pertenece este examen.
profundidad_camara_anterior	double precision	Guarda el valor correspondiente a la profundidad de la cámara anterior.
grosario_del_cristalino	double precision	Guarda el valor del grosario del cristalino.
longitud_accial	double precision	Guarda el valor de la longitud axial del ojo.

Tabla 2.15 Descripción de la tabla bqq_biometria.

Nombre: bqq_cirugia_anterior		
Descripción: Almacena los datos de las cirugías anteriores realizadas al paciente consultado.		
Atributo	Tipo	Descripción
id_cirugia_anterior	integer	Llave primaria que identifica a cada una de las cirugías anteriores realizadas.
id_tipo_cirugia	integer	Llave foránea proveniente de la tabla cfg_tipo_cirugia del módulo de configuración que identifica el tipo de la cirugía realizada.
tiempo_transcurrido	timestamp	Almacena el tiempo transcurrido desde la operación hasta el momento de la consulta.
observaciones	varchar	Almacena las observaciones que el médico haga sobre las cirugías realizadas anteriormente.

Tabla 2.16 Descripción de la tabla bqq_cirugia_anterior.

Nombre: bqq_cirugia_anterior_consulta		
Descripción: Almacena la lista de posibles cirugías realizadas anteriormente a la consulta.		
Atributo	Tipo	Descripción
id_persona	integer	Llave foránea proveniente de la tabla bqq_datos_consultas del módulo BQG para identificar a que consulta pertenece la cirugía.
numero_consulta	integer	Llave foránea proveniente de la tabla bqq_datos_consultas del módulo BQG para identificar a que consulta pertenece la cirugía.

Capítulo 2. Descripción y análisis de la solución propuesta.

id_cirugias_anteriores	integer	Llave foránea proveniente de la tabla bqo_cirugias_anteriores que identifica la cirugía anterior que se adiciona en la consulta.
------------------------	---------	--

Tabla 2.17 Descripción de la tabla bqo_cirugia_anterior_consulta.

Nombre: bqo_cristalino_examen		
Descripción: Almacena la lista de parámetros existentes en la anatómica consulta del cristalino		
Atributo	Tipo	Descripción
id_examen	integer	Llave foránea proveniente de la tabla bqo_examen_cristalino que identifica el examen al cual pertenece el parámetro.
id_parametro_cristalino	integer	Llave foránea perteneciente a la tabla cfg_parametro_cristalino que identifica cada uno de los parámetros de los exámenes del cristalino.

Tabla 2.18 Descripción de la tabla bqo_cristalino_examen.

Nombre: bqo_datos_entrada		
Descripción: Almacena los datos recogidos en la entrada del informe operatorio de la cirugía refractiva.		
Atributo	Tipo	Descripción
id_datos_entrada	integer	Llave primaria que identifica cada uno de los datos de entrada
id_informe_operatorio	integer	Llave foránea proveniente de la tabla bqo_informe_cirugia_refractiva que identifica de qué informe son los datos de entrada.
id_datos_parametros	integer	Llave foránea perteneciente a la tabla cfg_datos_parametros que identifica qué tipo de parámetro tiene el dato de entrada.
id_parametro_medicion	integer	Llave foránea proveniente de la tabla cfg_parametro_medicion que identifica cada parámetro de entrada.
valor	double precision	Valor asignado a cada parámetro con su tipo en los datos de entrada.

Tabla 2.19 Descripción de la tabla bqo_datos_entrada.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: bqo_datos_equipo		
Descripción: Almacena los datos correspondientes al equipo oftalmológico		
Atributo	Tipo	Descripción
id_datos_equipo	integer	Llave primaria que identifica cada equipo con sus datos.
marca	varchar	Marca del equipo
numero_serie	varchar	Número de serie del equipo
versión_software	double precision	Versión de software utilizada en el equipo.
nomograma	varchar	Nomograma del equipo
id_unidad	integer	Llave foránea proveniente de la tabla cfg_unidad correspondiente al módulo de configuración que identifica a qué unidad corresponde el equipo.

Tabla 2.20 Descripción de la tabla bqo_datos_equipo.

Nombre: bqo_datos_sistema		
Descripción: Almacena los datos provenientes del sistema en la cirugía refractiva.		
Atributo	Tipo	Descripción
id_informe_operatorio	integer	Llave foránea proveniente de la tabla bqo_informe_cirugia_refractiva que identifica a qué informe corresponden los datos del sistema.
ablación_por_capa	double precision	Datos recogidos por el sistema, correspondiente a cada uno con el nombre de los atributos.
scanner_offset	integer	
laser_hv_iniciar_tratamiento	integer	
laser_hv_finalizar_tratamiento	integer	
energía_laser_ajustada	double precision	
tratamiento_desde_ultimotestfluence	integer	

Tabla 2.21 Descripción de la tabla bqo_datos_sistema.

Nombre: bqo_examen_cristalino		
Descripción: Almacena los datos concernientes al examen anatómico del cristalino		
Atributo	Tipo	Descripción
id_examen	integer	Llave primaria que identifica a cada examen del

Capítulo 2. Descripción y análisis de la solución propuesta.

		cristalino.
id_examen Oftalmologico Anatomia	interger	Llave foránea proveniente de la tabla bqo_examen Oftalmologico Anatomia que identifica a qué examen general pertenece el del cristalino realizado.
id_tipo Catarata	interger	Llave foránea proveniente de la tabla cfg_tipo Catarata que identifica cada tipo de catarata presente en el examen.
id_tipo Lente	interger	Llave foránea proveniente de la tabla cfg_tipo Lente que identifica el tipo de lente presente en el paciente durante el examen.

Tabla 2.22 Descripción de la tabla bqo_examen_cristalino.

Nombre: bqo_examen_fondo_ojo		
Descripción: Almacena la lista de exámenes de fondo de ojo		
Atributo	Tipo	Descripción
id_examen Oftalmologico Anatomia	interger	Llave foránea que proviene de la tabla bqo_examen Oftalmologico Anatomia que especifica a qué examen corresponde la lista de parámetros de fondo de ojos.
id_parametro_fondo_ojo	interger	Llave foránea proveniente de la tabla cfg_parametro_fondo_ojo que describe el parámetro.

Tabla 2.23 Descripción de la tabla bqo_examen_fondo_ojo.

Nombre: bqo_examen_motilidad		
Descripción: Almacena la lista de exámenes de motilidad del examen anatómico del ojo		
Atributo	Tipo	Descripción
id_examen_motilidad	interger	Llave foránea que proviene de la tabla bqo_examen_motilidad_ocular que especifica a qué examen corresponde cada parámetro.
id_parametro_motilidad_ocular	interger	Llave foránea proveniente de la tabla

Capítulo 2. Descripción y análisis de la solución propuesta.

		cfg_parametro_motilidad_ocular que describe el parámetro.
--	--	---

Tabla 2.24 Descripción de la tabla bqo_examen_motilidad.

Nombre: bqo_examen_motilidad_ocular		
Descripción: Almacena la lista de exámenes de motilidad ocular y la clasificación en esotropía o exotropía de cada motilidad		
Atributo	Tipo	Descripción
id_examen_motilidad	integer	Llave primaria que identifica cada examen de motilidad ocular y su clasificación
id_examen_ofthalmologico_anatomia	integer	Llave foránea que indica a qué examen anatómico pertenece el examen de motilidad ocular.
id_otropia	integer	Llave foránea proveniente de la tabla cfg_otropia que almacena la clasificación de la motilidad.

Tabla 2.25 Descripción de la tabla bqo_examen_motilidad_ocular.

Nombre: bqo_examen_ofthalmologico_anatomia		
Descripción: Esta tabla contiene los datos generales de los exámenes de anatomía oftalmológica		
Atributo	Tipo	Descripción
id_persona	integer	Llave foránea proveniente de la tabla bqg_datos_consultas del módulo BQG para identificar a qué consulta pertenece el examen.
numero_consulta	integer	Llave foránea proveniente de la tabla bqg_datos_consultas del módulo BQG para identificar a que consulta pertenece el examen.
id_ojos	integer	Llave foránea proveniente de la tabla cfg_ojos que identifica el ojo al cual se le realiza el examen.
id_examen_ofthalmologico_anatomia	integer	Llave primaria que identifica a cada uno de los exámenes anatómicos de oftalmología realizados.

Tabla 2.26 Descripción de la tabla bqo_examen_ofthalmologico_anatomia.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: bqo_examen_reflejo_pupilar		
Descripción: Almacena la lista de exámenes de reflejo pupilar		
Atributo	Tipo	Descripción
id_examen Oftalmologico Anatomia	integer	Llave foránea que proviene de la tabla bqo_examen Oftalmologico Anatomia que especifica a qué examen corresponde la lista de parámetros de reflejo pupilar.
id_parametro_reflejo_pupilar	integer	Llave foránea proveniente de la tabla cfg_parametro_reflejo_pupilar que describe el parámetro.

Tabla 2.27 Descripción de la tabla bqo_examen_reflejo_pupilar

Nombre: bqo_examen_segmento_anterior		
Descripción: Almacena la lista de exámenes del segmento anterior		
Atributo	Tipo	Descripción
id_examen Oftalmologico Anatomia	integer	Llave foránea que proviene de la tabla bqo_examen Oftalmologico Anatomia que especifica a qué examen corresponde la lista de parámetros de segmento anterior.
id_parametro_segmento_anterior	integer	Llave foránea proveniente de la tabla cfg_parametro_segmento_anterior que describe el parámetro.

Tabla 2.28 Descripción de la tabla bqo_examen_segmento_anterior.

Nombre: bqo_examen_vitreo		
Descripción: Almacena la lista de exámenes de vítreo		
Atributo	Tipo	Descripción
id_examen Oftalmologico Anatomia	integer	Llave foránea que proviene de la tabla bqo_examen Oftalmologico Anatomia que especifica a qué examen corresponde la lista de

Capítulo 2. Descripción y análisis de la solución propuesta.

		parámetros de vítreo.
id_parametro_vitreo	integer	Llave foránea proveniente de la tabla cfg_parametro_vitreo que describe el parámetro.

Tabla 2.29 Descripción de la tabla bqo_examen_vitreo.

Nombre: bqo_examen_anexos		
Descripción: Almacena la lista de exámenes de los anexos oculares		
Atributo	Tipo	Descripción
id_examen Oftalmologico Anatomia	integer	Llave foránea que proviene de la tabla bqo_examen Oftalmologico Anatomia que especifica a qué examen corresponde la lista de parámetros de los anexos.
id_parametro_anexos	integer	Llave foránea proveniente de la tabla cfg_parametro_anexos que describe el parámetro.

Tabla 2.30 Descripción de la tabla bqo_examen_anexos.

Nombre: bqo_examenes Oftalmologicos		
Descripción: Esta tabla contiene los datos generales de los exámenes oftalmológicos		
Atributo	Tipo	Descripción
id_persona	integer	Llave foránea proveniente de la tabla bqg_datos_consultas del módulo BQG para identificar a qué consulta pertenece el examen.
numero_consulta	integer	Llave foránea proveniente de la tabla bqg_datos_consultas del módulo BQG para identificar a qué consulta pertenece el examen.
id_ojo	integer	Llave foránea proveniente de la tabla cfg_ojo que identifica el ojo al cual se le realiza el examen.
iol_master	boolean	Atributo que identifica si el examen fue realizado o no en el equipo iol_master.

Tabla 2.31 Descripción de la tabla bqo_examenes Oftalmologicos.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: bqo_informe_cirugia_refractiva		
Descripción: Almacena los datos correspondientes al informe operatorio oftalmológico de la especialidad cirugía del cristalino.		
Atributo	Tipo	Descripción
id_informe_operatorio	integer	Llave foránea perteneciente a la tabla bqo_informe_operatorio Oftalmológico que indica a qué informe general pertenece el de la cirugía refractiva.
id_tratamiento_refractivo	integer	Llave foránea proveniente de la tabla cfg_tratamiento_refractivo que indica qué tratamiento refractivo fue el utilizado durante la intervención quirúrgica.
id_datos_equipo	integer	Llave foránea proveniente de la tabla bqo_datos_equipo que indica con qué equipo fueron realizadas las mediciones de la cirugía refractiva.
metodo	varchar	Método utilizado en la cirugía
nombre_archivo_entrada	varchar	Nombre del archivo entrado al sistema para realizar la cirugía.
temperatura	integer	Temperatura ambiente
humedad	integer	Humedad relativa ambiente
comentario	varchar	Comentario del especialista sobre la cirugía.
minimo_tiempo_tratamiento	integer	Tiempo mínimo a realizar el tratamiento
tiempo_entre_pulsos	integer	Tiempo entre los pulsos.
numero_mensajes_eye_tracking	integer	Número de mensajes del eyetracking
estado_eye_tracking	varchar	Estado del eyetracking
numero_pulsacione	integer	Numero de pulsaciones láser.
numero_capas	integer	Numero de capas
numero_errores	integer	Numero de errores de medición
tratamiento_realizado_porcentaje	integer	Porcentaje de tratamiento realizado
tratamiento_realizado_valor	double precision	Valor del tratamiento realizado
tratamiento_realizado	varchar	Descripción del tratamiento realizado en la cirugía.

Capítulo 2. Descripción y análisis de la solución propuesta.

id_funcionario_usuario	integer	Llave foránea proveniente de la tabla cfg_funcionario del módulo de configuración que describe el funcionario del equipo.
------------------------	---------	---

Tabla 2.32 Descripción de la tabla bqo_informe_cirugia_refractiva.

Nombre: bqo_informe_operatorio_cristalino		
Descripción: Almacena los datos correspondientes al informe operatorio de la especialidad del cristalino.		
Atributo	Tipo	Descripción
id_informe_operatorio	integer	Llave foránea proveniente de la tabla bqg_informe_operatorio del BQG que almacena el anuncio operatorio general al cual pertenecen los datos del especializado.
id_forma_extraccion	integer	Llave foránea proveniente de la tabla cfg_forma_extraccion que almacena la forma de extracción del cristalino utilizada en la cirugía.
id_iridectomia	integer	Llave foránea proveniente de la tabla cfg_iridectomia que almacena la forma de iridectomía llevada a cabo en la cirugía.
id_lugar_colocacion	integer	Llave foránea proveniente de la tabla cfg_lugar_colocacion que almacena el lugar de colocación del lente intraocular implantado en la cirugía.
ipertencion_arterial_secundaria	boolean	Almacena si existe o no hipertensión arterial secundaria durante la cirugía.
afaquia	boolean	Almacena si el paciente presenta afaquia luego de la cirugía.
especificacion_no_lente	varchar	Almacena las especificaciones del por que de no implantar lente al paciente.
tiempo_faco	timestamp	Almacena el tiempo de duración de la faco.

Tabla 2.33 Descripción de la tabla bqo_informe_operatorio_cristalino.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: bqo_informe_operatorio Oftalmológico		
Descripción: almacena los datos correspondientes al informe operatorio oftalmológico de manera general.		
Atributo	Tipo	Descripción
id_anuncio_operatorio	integer	Llave foránea correspondiente a la tabla bqg_informe_operatorio perteneciente al módulo BQG que identifica a qué informe general corresponde el oftalmológico.
id_tipo_informe Oftalmológico	integer	Llave foránea proveniente de la tabla cfg_tipo_informe_operatorio que describe de qué tipo es el informe oftalmológico.
id_ojo	integer	Llave foránea correspondiente a la tabla cfg_ojo que describe a qué ojo se le realizara el informe operatorio.
refraccion	double precision	Refracción obtenida al concluir la intervención quirúrgica.

Tabla 2.34 Descripción de la tabla bqo_informe_operatorio_oftalmologico.

Nombre: bqo_lente_anuncio		
Descripción: Almacena la posible lista de lentes utilizada en el anuncio operatorio		
Atributo	Tipo	Descripción
id_anuncio_operatorio	integer	Llave foránea que proviene de la tabla bqo_anuncio_cirugia_crist que almacena el anuncio al cuál se le relaciona el lente.
id_lio	integer	Llave foránea proveniente de la tabla bqo_lente_intraocular que almacena el lente correspondiente al anuncio.

Tabla 2.35 Descripción de la tabla bqo_lente_anuncio.

Nombre: bqo_lente_informe		
Descripción: Almacena la posible lista de lentes utilizada en el informe operatorio		
Atributo	Tipo	Descripción
id_informe_operatorio	integer	Llave foránea que proviene de la tabla bqo_anuncio_cirugia_crist que almacena el informe

Capítulo 2. Descripción y análisis de la solución propuesta.

		operatorio al cuál se le relaciona el lente.
id_lio	integer	Llave foránea proveniente de la tabla bqo_lente_intraocular que almacena el lente correspondiente al informe.

Tabla 2.36 Descripción de la tabla bqo_lente_informe.

Nombre: bqo_lente_intraocular		
Descripción: Almacena los datos correspondiente a cada lente intraocular		
Atributo	Tipo	Descripción
id_lio	integer	Llave primaria que identifica a cada lente intraocular.
id_tipo_lente	integer	Llave foránea proveniente de la tabla cfg_tipo_lente que almacena el tipo de lente intraocular.
poder_dioptrico	double precision	Almacena el valor del poder dióptrico del lente.
constante_fabricante	varchar	Almacena el nombre de la constante del fabricante utilizada en el lente.
modelo_lente	varchar	Almacena el modelo del lente intraocular.

Tabla 2.39 Descripción de la tabla bqo_lente_intraocular.

Nombre: bqo_queratometria		
Descripción: Almacena los datos correspondientes al examen de queratometría.		
Atributo	Tipo	Descripción
id_examen Oftalmológico	integer	Llave foránea proveniente de la tabla bqo_examenes Oftalmológicos que indica a qué grupo de exámenes pertenece la queratometría.
meridiano_debil	double precision	Almacena el valor del meridiano débil
meridiano_fuerte	double precision	Almacena el valor del meridiano fuerte
eje_debil	double precision	Almacena el valor del eje débil
eje_fuerte	double precision	Almacena el valor del eje fuerte

Tabla 2.40 Descripción de la tabla bqo_queratometria.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: bqo_refraccion		
Descripción: Almacena los valores correspondientes al examen de refracción.		
Atributo	Tipo	Descripción
id_examen Oftalmologico	integer	Llave foránea proveniente de la tabla bqo_examenes Oftalmologicos que indica a qué grupo de exámenes pertenece la refracción.
agudeza_visual_sin_correccion	double precision	Almacena el valor correspondiente a la agudeza visual sin corrección.
esfera	double precision	Almacena el valor correspondiente a la esfera.
cilindro	double precision	Almacena el valor correspondiente al cilindro.
eje	double precision	Almacena el valor correspondiente al eje.
agudeza_visual_con_correccion	double precision	Almacena el valor correspondiente a la agudeza visual con corrección.

Tabla 2.41 Descripción de la tabla bqo_refraccion.

Nombre: bqo_tension_ocular		
Descripción: Almacena los valores correspondientes al examen de tensión ocular.		
Atributo	Tipo	Descripción
id_examen Oftalmologico		Llave foránea proveniente de la tabla bqo_examenes Oftalmologicos que indica a qué grupo de exámenes pertenece la tensión ocular.
elastotonometria	double precision	Almacena el valor de la elastotonometria.
radio_curvatura	double precision	Almacena el valor de radio_curvatura.
hipertension_arterial	boolean	Almacena si existe o no hipertensión arterial en el paciente.

Tabla 2.42 Descripción de la tabla bqo_tension_ocular.

Nombre: cfg_antecedente Oftalmologico		
Descripción: tabla de configuración que describe los antecedentes patológicos.		
Atributo	Tipo	Descripción
id_antecedentes Oftalmologicos	integer	Llave principal que identifica cada antecedente

Capítulo 2. Descripción y análisis de la solución propuesta.

		oftalmológico
descripción	varchar	Descripción de cada antecedente oftalmológico.

Tabla 2.43 Descripción de la tabla `cfg_antecedente_ofthalmologico`.

Nombre: <code>cfg_datos_parametros</code>		
Descripción: tabla de configuración que describe los datos de los parámetros de entrada de la cirugía refractiva.		
Atributo	Tipo	Descripción
<code>id_datos_parametros</code>	integer	Llave principal que identifica cada dato de los parámetros de entrada de la cirugía refractiva
descripción	varchar	Descripción de cada dato correspondiente a los parámetros de entrada de la cirugía refractiva.

Tabla 2.44 Descripción de la tabla `cfg_datos_parametros`.

Nombre: <code>cfg_forma_extraccion</code>		
Descripción: tabla de configuración que describe la forma de extracción del cristalino.		
Atributo	Tipo	Descripción
<code>id_forma_extraccion</code>	integer	Llave principal que identifica cada forma de extracción del cristalino
descripción	varchar	Descripción de cada forma de descripción.

Tabla 2.45 Descripción de la tabla `cfg_forma_extraccion`.

Nombre: <code>cfg_formula_dioptria</code>		
Descripción: tabla de configuración que describe las formulas utilizadas para el cálculo de las dioptrías de un lente intraocular.		
Atributo	Tipo	Descripción
<code>id_formula_dioptria</code>	integer	Llave principal que identifica cada formula utilizada
descripción	varchar	Descripción de cada formula dióptrica.

Tabla 2.46 Descripción de la tabla `cfg_formula_dioptria`.

Nombre: <code>cfg_iridectomia</code>		
Descripción: tabla de configuración que describe las formas de iridectomías		

Capítulo 2. Descripción y análisis de la solución propuesta.

Atributo	Tipo	Descripción
id_iridectomia	integer	Llave principal que identifica cada iridectomía
descripción	varchar	Descripción de cada iridectomía.

Tabla 2.47 Descripción de la tabla cfg_iridectomia.

Nombre: cfg_lugar_colocacion_cristalino		
Descripción: tabla de configuración que describe el lugar de colocación del lente intraocular en sustitución del cristalino		
Atributo	Tipo	Descripción
id_lugar_colocacion_cristalino	integer	Llave principal que identifica cada lugar de colocación del lente
descripción	varchar	Descripción de cada lugar de colocación del lente.

Tabla 2.48 Descripción de la tabla cfg_lugar_colocacion_cristalino.

Nombre: cfg_ojo		
Descripción: tabla de configuración que describe el nombre de cada ojo		
Atributo	Tipo	Descripción
id_ojo	integer	Llave principal que identifica cada ojo
descripción	varchar	Descripción de cada ojo.

Tabla 2.49 Descripción de la tabla cfg_ojo.

Nombre: cfg_otropia		
Descripción: tabla de configuración que describe la clasificación de la motilidad ocular		
Atributo	Tipo	Descripción
id_otropia	integer	Llave principal que identifica cada clasificación de la motilidad ocular
descripción	varchar	Descripción de cada clasificación de la motilidad ocular

Tabla 2.50 Descripción de la tabla cfg_otropia.

Nombre: cfg_parametro_anexos		
Descripción: tabla de configuración que describe los parámetros de los exámenes de los anexos oftalmológicos		
Atributo	Tipo	Descripción

Capítulo 2. Descripción y análisis de la solución propuesta.

id_parametro_anexos	integer	Llave principal que identifica cada parámetro de los anexos.
descripción	varchar	Descripción de cada parámetro de los anexos.

Tabla 2.51 Descripción de la tabla cfg_parametro_anexos.

Nombre: cfg_parametro_cristalino		
Descripción: tabla de configuración que describe los parámetros de los exámenes del cristalino		
Atributo	Tipo	Descripción
id_parametro_cristalino	integer	Llave principal que identifica cada parámetro de los exámenes del cristalino
descripción	varchar	Descripción de cada parámetro de los exámenes del cristalino

Tabla 2.52 Descripción de la tabla cfg_parametro_cristalino.

Nombre: cfg_parametro_fondo_ojo		
Descripción: tabla de configuración que describe los parámetros de los exámenes de fondo de ojo		
Atributo	Tipo	Descripción
id_parametro_fondo_ojo	integer	Llave principal que identifica cada parámetro de los exámenes de fondo de ojos.
descripción	varchar	Descripción de cada parámetro de fondo de ojos.

Tabla 2.53 Descripción de la tabla cfg_parametro_fondo_ojo.

Nombre: cfg_parametro_medicion		
Descripción: tabla de configuración que describe los parámetros de medición de los datos de entrada de la cirugía refractiva.		
Atributo	Tipo	Descripción
id_parametro_medicion	integer	Llave principal que identifica cada parámetro de medición de los datos de entrada.
descripción	varchar	Descripción de cada parámetro de medición.

Tabla 2.54 Descripción de la tabla cfg_parametro_medicion.

Nombre: cfg_parametro_motilidad_ocular		
Descripción: tabla de configuración que describe los parámetros de los exámenes de motilidad ocular.		

Capítulo 2. Descripción y análisis de la solución propuesta.

Atributo	Tipo	Descripción
id_parametro_motilidad_ocular	integer	Llave principal que identifica cada parámetro de los exámenes de motilidad ocular.
descripción	varchar	Descripción de cada parámetro de motilidad ocular.

Tabla 2.55 Descripción de la tabla `cfg_parametro_motilidad_ocular`.

Nombre: <code>cfg_parametro_reflejo_pupilar</code>		
Descripción: tabla de configuración que describe los parámetros de los exámenes de reflejo pupilar		
Atributo	Tipo	Descripción
id_parametro_reflejo_pupilar	integer	Llave principal que identifica cada parámetro de los exámenes de reflejo pupilar.
descripción	varchar	Descripción de cada parámetro de reflejo pupilar.

Tabla 2.56 Descripción de la tabla `cfg_parametro_reflejo_pupilar`.

Nombre: <code>cfg_parametro_segmento_anterior</code>		
Descripción: tabla de configuración que describe los parámetros de los exámenes del segmento anterior.		
Atributo	Tipo	Descripción
id_parametro_segmento_anterior	integer	Llave principal que identifica cada parámetro de los exámenes de segmento anterior.
descripción	varchar	Descripción de cada parámetro del segmento anterior.

Tabla 2.57 Descripción de la tabla `cfg_parametro_segmento_anterior`.

Nombre: <code>cfg_parametro_vitreo</code>		
Descripción: tabla de configuración que describe los parámetros de los exámenes de vítreo.		
Atributo	Tipo	Descripción
id_parametro_vitreo	integer	Llave principal que identifica cada parámetro de los exámenes de vítreo.
descripción	varchar	Descripción de cada parámetro de vítreo.

Tabla 2.58 Descripción de la tabla `cfg_parametro_vitreo`.

Capítulo 2. Descripción y análisis de la solución propuesta.

Nombre: cfg_tipo_anuncio_quirurgico		
Descripción: tabla de configuración que describe los tipos de anuncio quirúrgico existentes en el sistema.		
Atributo	Tipo	Descripción
id_tipo_anuncio_quirurgico	integer	Llave principal que identifica cada tipo de anuncio quirúrgico.
descripción	varchar	Descripción de cada tipo de anuncio quirúrgico.

Tabla 2.59 Descripción de la tabla cfg_tipo_anuncio_quirurgico.

Nombre: cfg_tipo_catarata		
Descripción: tabla de configuración que describe los tipos de cataratas		
Atributo	Tipo	Descripción
id_tipo_catarata	integer	Llave principal que identifica cada tipo de catarata.
descripción	varchar	Descripción de cada tipo de catarata.

Tabla 2.60 Descripción de la tabla cfg_tipo_catarata.

Nombre: cfg_tipo_cirugia		
Descripción: tabla de configuración que describe los tipos de cirugías oftalmológicas.		
Atributo	Tipo	Descripción
id_tipo_cirugia	integer	Llave principal que identifica cada tipo de cirugía.
descripción	varchar	Descripción de cada tipo de cirugía.

Tabla 2.61 Descripción de la tabla cfg_tipo_cirugia

Nombre: cfg_tipo_informe Oftalmologico		
Descripción: tabla de configuración que describe los tipos de informes operatorios de oftalmología.		
Atributo	Tipo	Descripción
id_tipo_informe_oftalmologico	integer	Llave principal que identifica cada tipo de informe operatorio de oftalmología.
descripción	varchar	Descripción de cada tipo de informe oftalmológico.

Tabla 2.62 Descripción de la tabla cfg_tipo_informe_operatorio.

Nombre: cfg_tipo_lente		
Descripción: tabla de configuración que describe los tipos de lentes intraoculares.		
Atributo	Tipo	Descripción

Capítulo 2. Descripción y análisis de la solución propuesta.

id_tipo_lente	integer	Llave principal que identifica cada tipo de lente.
descripción	varchar	Descripción de cada tipo de lente.

Tabla 2.63 Descripción de la tabla cfg_tipo_lente.

Nombre: cfg_tipo_tratamiento_refractivo		
Descripción: tabla de configuración que describe tipo de tratamiento refractivo llevado a cabo durante la cirugía		
Atributo	Tipo	Descripción
id_tipo_tratamiento_refractivo	integer	Llave principal que identifica cada tipo de tratamiento refractivo.
descripción	varchar	Descripción de cada tipo de tratamiento refractivo.

Tabla 2.64 Descripción de la tabla cfg_tipo_tratamiento_refractivo.

En este capítulo ha quedado plasmado el diseño del Sistema de Almacenamiento de la Información correspondiente al Bloque Quirúrgico Oftalmológico. Para ello se ha modelado un diagrama entidad relación para generar el modelo físico de la base de datos que cumple con los requisitos funcionales preestablecido por los analistas del sistema.

Se ha generado una base de datos con sus consultas, las cuales se encuentran optimizadas y se garantiza un tiempo de respuesta de las mismas acordes con las necesidades del sistema. Se ha establecido la arquitectura de base de datos centralizada para el sistema. Así como el medio y la forma en que interactúa el mismo con el resto de los módulos.

CAPÍTULO 3: VALIDACIÓN DEL DISEÑO REALIZADO

El siguiente capítulo pretende realizar la validación del diseño de la base de datos para el almacenamiento y la manipulación de la información correspondiente al Bloque Quirúrgico Oftalmológico.

Dentro de esta validación se abordará la integridad de la información, la normalización del modelo de datos realizado, el análisis de la redundancia de la información, la seguridad del sistema de bases de datos diseñado y la trazabilidad de las acciones.

Por último, se realizará la valoración del trabajo desarrollado y la propuesta de integración con el diseño.

3.1 Validación teórica del diseño.

El diseño de un sistema de bases de datos antes de su puesta a punto, debe ser validado teóricamente para garantizar su correcto funcionamiento. Para esto se deben tener en cuenta una serie de requisitos como son la integridad de los datos, la normalización de dicho modelo, la existencia o no de redundancia de la información a almacenar, la seguridad del Sistema de Base de Datos y la trazabilidad de cada una de las acciones realizadas durante la manipulación de la información.

3.1.1 Integridad.

Una vez definida la estructura de datos del modelo relacional, es necesario analizar las reglas de integridad de los datos almacenados en dicha estructura que deben cumplirse para garantizar que los mismos sean correctos.

Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, proteger los datos ante fallos de hardware, usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

Consiste en garantizar la no contradicción entre los datos almacenados de modo que, en cualquier momento del tiempo, estos datos sean correctos, es decir, que no se detecte inconsistencia entre ellos.

La integridad está relacionada de forma directa con la minimización de redundancia. Es más sencillo garantizar la integridad si se elimina la inconsistencia entre los datos que pretenden representar el mismo “hecho”. [35]

Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada uno de ellos, a este tipo se les denomina restricciones de dominios. Hay además dos reglas de integridad muy importantes que son restricciones que se deben cumplir en todas las bases de datos relacionales y en todos sus estados o instancias. Estas reglas son: la regla de integridad de entidades y la regla de integridad referencial. [36]

Una restricción de dominio (o regla de integridad de dominio) es aquella que se aplica a toda una columna definida en el dominio en cuestión, restringe cada posible inserción en dicha columna para que acepte solamente un rango de valor predeterminado acorde con el negocio del sistema, de forma tal que si se tratara de insertar un valor fuera de este rango la operación fallará. Estas restricciones forman parte de la garantía de la integridad de la presente base de datos.

En cuanto a estas restricciones en el Sistema de Almacenamiento de la Información del Bloque Quirúrgico Oftalmológico se han definido una serie de dominios en la base de datos acordes con cada uno de los atributos que específicamente lo necesitan en el presente sistema, este es el caso de las pruebas oftalmológicas de refracción y queratometría donde el valor del eje del ojo del paciente tiene que estar entre 0 y 180 grados, para garantizar esto se creó en la base de datos el dominio `bqo_eje_dom`, el cual es aplicado a las tuplas eje de la tabla `bqo_refraccion` y a las tuplas eje_fuerte y eje_debil de la tabla `bqo_queratometria`. De igual manera se creó una restricción de dominio llamada `bqo_cilindro_dom` para garantizar que el valor correspondiente a la tupla cilindro de la tabla `bqo_refraccion` sea siempre menor que cero.

También se encuentran las reglas de integridad de entidades, la primera de estas se aplica a las claves primarias de las relaciones base: ninguno de los atributos que componen la clave primaria puede ser nulo.

Por definición, una clave primaria es un identificador irreducible que se utiliza para identificar de modo único las tuplas. Que es irreducible significa que ningún subconjunto de la clave primaria sirve para identificar las tuplas de modo único. Si se permite que parte de la clave primaria sea nula, se está diciendo que no todos sus atributos son necesarios para distinguir las tuplas, con lo que se contradice la

irreductibilidad. Nótese que esta regla sólo se aplica a las relaciones base y a las claves primarias, no a las claves alternativas.

En el presente sistema, que utiliza como Sistema Gestor de Base de Datos PostgreSQL en la versión 8.2 garantiza que siempre se cumpla con esta regla de integridad, lanzando excepciones de error en caso de que se intente introducir un valor de clave primaria nula.

Otra regla es la integridad referencial, la cual se aplica a las claves ajenas: si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos. [37]

La integridad referencial es una propiedad deseable en las bases de datos. Gracias a esta se garantiza que una entidad (fila o registro) siempre se relaciona con otras entidades válidas, es decir, que existen en la base de datos. Implica que en todo momento dichos datos sean correctos, sin repeticiones innecesarias, datos perdidos o relaciones mal resueltas.

Todas las bases de datos relacionales gozan de esta propiedad gracias a que el software gestor de base de datos vela por su cumplimiento. En el presente, el gestor de bases de datos PostgreSQL 8.2 vela por el cumplimiento de la integridad referencial de los datos.

3.1.2 Normalización de la base de datos.

La normalización es el proceso por el cual se estudia la semántica de los atributos de una relación y se divide la relación original en varias relaciones que no tengan redundancia descontrolada que pueda producir problemas de actualización. La base del proceso es la noción de dependencias funcionales entre atributos. La normalización es una técnica, desarrollada inicialmente por E.F. Codd en 1972, para diseñar la estructura lógica de una base de datos en el modelo relacional, es un proceso en el cual se va comprobando el cumplimiento de una serie de reglas, o restricciones, por parte de un esquema de relación; cada regla que se cumple aumenta el grado de normalización del esquema de relación; si una regla no se cumple, el esquema de relación se debe descomponer en varios esquemas de relación que sí la cumplan por separado. Para evitar anomalías de actualización, es recomendable llegar al menos hasta la tercera forma normal. [38]

La normalización ayuda a estructurar la base de datos de tal forma que las actualizaciones de tuplas individuales sean más aceptables lógicamente de lo que serían de otro modo, es decir, si el diseño no

estuviese normalizado. Esto se logra debido a que los predicados son más sencillos con un diseño completamente normalizado.

Existen varios niveles de normalización o formas normales:

1. Primera Forma Normal. (1NF)
2. Segunda Forma Normal. (2NF)
3. Tercera Forma Normal. (3NF)
4. Forma Normal de Boyle-Codd (BCNF)
5. Quinta Forma Normal. (5NF)

Se dice que una tabla está en una forma normal si satisface cierto conjunto de normas preestablecidas, así se tiene que un esquema está en primera forma normal si y sólo si los dominios de todos los atributos de la relación son atómicos. [39]

Para que un sistema se encuentre en la segunda forma normal entonces, debe cumplir que está en primera forma normal y además que cada atributo del sistema de relación que no está en la clave primaria depende funcionalmente de la clave primaria completa y no sólo de una parte de esta. [40] Por último se tiene que un esquema de relación se encuentra en tercera forma normal si y sólo si, está en segunda forma normal y además cada atributo del esquema de relación que no está en la clave primaria depende funcionalmente de la clave primaria, y no de ningún otro atributo. [41] El presente sistema consta de un esquema de relaciones que cumple con estas condiciones por lo que se puede afirmar que se encuentra en tercera forma normal.

Sin embargo la optimización y el rendimiento de las consultas o queries realizados a la presente base de datos se ven afectados directamente por la normalización del sistema,

A menudo, para tablas con más de dos atributos, existen varias posibles descomposiciones de la tabla original en tablas normalizadas. Para determinar cuál es el diseño más conveniente, es necesario conocer el perfil de transacciones que se van a usar en la base de datos. Si la mayor parte de las transacciones que se realizan son de inserción o actualización es más conveniente una mayor normalización de la base de datos, pues cada índice primario no denso tendría un nivel menos que el índice de una relación no

normalizada y en la memoria principal cabrían más tuplas de relación con lo que se aumentaría el radio para búsquedas aleatorias.

Sin embargo si las consultas serían de selección múltiple no es conveniente la subdivisión de tablas en otras para la normalización total del sistema, por ejemplo, una sola tabla normalizada que contenga los atributos X Y Z es mejor que dos tablas normalizadas X Y , X Z pues se pueden acceder todos los atributos sin necesidad de hacer join. El diseño con las dos relaciones se denomina partición vertical y es mejor en las siguientes condiciones: Los usuarios acceden a los datos particionados como están en las dos tablas. Los atributos contenidos en Y y los atributos contenidos en Z tienen valores cuya longitud es muy grande, por ejemplo ocupan un tercio o más del tamaño de un bloque.

Como en el presente sistema se realizan tanto operaciones de inserción como consultas múltiples se ha preferido priorizar el rendimiento de este por encima de la estructura normalizada por lo que se decidió solamente llegar hasta la tercera forma normal, y no proseguir con las formas normales siguientes para que se tengan que realizar un número menor de JOINS y la selección de los datos se realice con estos ubicados de manera contigua en memoria, recordando también que en postgresSQL por cada consulta que devuelva la combinación de una o más tablas hay que realizar una vista, lo que ralentizaría la respuesta del mismo; de esta forma se ha logrado agilizar la velocidad de respuesta del presente sistema de bases de datos.

3.1.3 Análisis de redundancia de la información.

En sistemas que no son de bases de datos, cada aplicación tiene sus propios archivos. A menudo esto puede conducir a una redundancia considerable de los datos almacenados, con el consecuente desperdicio de espacio de almacenamiento. Sin embargo estos archivos pueden integrarse y eliminar así las redundancias, en tanto el administrador de datos esté consciente de los requerimientos de datos de las aplicaciones.

No toda la redundancia debe o puede ser eliminada. En ocasiones existen razones sólidas, tácticas o del negocio, para tener varias copias distintas de los mismos datos. Sin embargo, cualquier redundancia de este tipo debe ser controlada cuidadosamente; es decir, el Sistema de Gestión de Base de Datos debe estar al tanto de ella, si es que existe, y debe asumir la responsabilidad de propagar las actualizaciones. [42].

Por consiguiente se puede definir que evitar la redundancia de un sistema es evitar que dos tipos de registros tengan la misma información sobre el mismo tipo de entidad.

La redundancia de los datos debe ser controlada, de forma que no existan duplicidades perjudiciales ni innecesarias, y que las redundancias físicas, convenientes muchas veces a fin de responder a objetivos de eficiencia, sean tratadas por el mismo sistema, de modo que no puedan producirse incoherencias. Por tanto, un dato se actualizará lógicamente por el usuario de forma única, y el sistema se preocupará de cambiar físicamente todos aquellos campos en los que el dato estuviese repetido, en caso de existir redundancia física.

Partiendo de todo esto, en el desarrollo del Sistema de Almacenamiento de la Información Correspondiente al Bloque Quirúrgico Oftalmológico se ha tenido en cuenta una redundancia mínima de la información, quedando solamente la redundancia correspondiente a los datos del examen del cristalino, presentes en las tablas bqo_cristalino_examen y bqo_examen_cristalino. Esta es manejada por el sistema porque es necesario guardar siempre con anterioridad las especificaciones del examen que se va a mostrar al usuario, que deben ser cargados desde la base de datos, y luego almacenar en otra tabla la selección hecha de ellos por el usuario, en este caso de la propagación de las actualizaciones se encarga el gestor de base de datos a partir de dependencias de integridad referencial.

3.1.4 Análisis de la seguridad de la base de datos.

El concepto de seguridad de los datos en Sistemas de Bases de Datos, se asocia frecuentemente con la integridad de los mismos, pero estos conceptos difieren entre sí, ya que la seguridad se refiere a la protección de los datos contra su revelación, su alteración o su destrucción no autorizada, es decir, garantiza que los usuarios tengan permisos de hacer las operaciones que estén tratando de hacer; mientras que la integridad, como se vio anteriormente en la sección 3.1.1, se refiere a la precisión o validez de esos datos, es decir, que las operaciones que el usuario trate de hacer sean correctas. [43]

El Sistema de Almacenamiento de la Información del Bloque Quirúrgico Oftalmológico debe manejar un grupo de datos sensibles con relación a la salud de las personas, los que son estrictamente confidenciales por lo que es responsabilidad del mismo garantizar la confidencialidad y seguridad de estos.

Este sistema, como se explicó anteriormente, forma parte del Sistema de Información Hospitalaria, en el que se desarrolla a la par un módulo de seguridad, que se encarga de la misma en las tres capas de la aplicación, lo que incluye, como es lógico, la capa de datos. Por lo que el presente módulo solamente se hace cargo de la seguridad que brinda el propio gestor de bases de datos PostgreSQL 8.2.

En este sentido la seguridad de PostgreSQL brinda la facilidad de definir usuarios y roles, dándole a cada uno los permisos necesarios para acceder a la base de datos, es así que se define el rol “Usuario de Base de Datos” que va a tener los permisos necesarios para realizar consultas, y modificar el contenido de la base de datos pero sin afectar el diseño de la misma, o sea, no puede añadir, o eliminar tablas, funciones, índices u otras estructuras del sistema; y el rol “Administrador de la Base de Datos”, que sí tiene todos los permisos necesarios para modificar todo lo concerniente a la misma.

De esta manera se definen entonces los usuarios que van a desempeñar cada uno de los roles, que son los casos de los usuarios gehosbs y gehosbr que pertenece al rol de usuario de bases de datos y el usuario postgres que va a ejercer el rol de administrador de base de datos, quien va a ser el único en este rol, puesto que es el responsable de administrar y mantener el Sistema de Bases de Datos correspondiente al Bloque Quirúrgico Oftalmológico.

3.1.5 Trazabilidad de las acciones.

La trazabilidad de un sistema es aquello que deja un rastro de las acciones que el usuario realiza con los datos. Por ejemplo, si un usuario elimina un registro, queda constancia del registro que se eliminó, quien lo eliminó y cuándo. De esta forma, puede hacerse un seguimiento de todas las acciones que se han realizado sobre un registro concreto. O todas las acciones que ha realizado un usuario concreto. Pudiendo conformar un historial de las acciones sobre los datos. La trazabilidad puede realizarse sobre tablas o sobre procesos. Incluso pueden crearse trazabilidades sobre errores o conflictos de la aplicación. [44]

En el Sistema de Almacenamiento de la Información correspondiente al Bloque Quirúrgico Oftalmológico la trazabilidad de las acciones sólo se lleva a nivel de gestor de base de datos, a partir de los logs de PostgreSQL ya que de la trazabilidad detallada de todas las transacciones se encarga el módulo de seguridad del proyecto.

Capítulo3. Validación del diseño realizado.

En este capítulo se ha realizado la validación teórica del Sistema de Almacenamiento de la Información correspondiente al Bloque Quirúrgico Oftalmológico. Para ello se ha analizado cómo el mismo se encuentra en tercera forma normal, y que debe llegar sólo hasta esta forma normal para que las consultas realizadas al mismo se puedan realizar de manera más eficiente.

El sistema se encuentra libre de redundancias fuera de control, la existente es manejada correctamente por el mismo, logrando así que la integridad de los datos no se vea afectada en este sentido, el resto de las reglas de integridad se llevan a cabo de manera satisfactoria, cumpliendo con la integridad de dominio, de entidad y referencial.

Por otra parte de la seguridad y la trazabilidad de las acciones el sistema sólo se encarga de esta, a nivel de gestor de base de datos, ya que del resto se encarga un módulo de seguridad que se desarrolla paralelo al mismo.

CONCLUSIONES

Al concluir el desarrollo del Trabajo de Diploma se obtuvieron los resultados esperados en función de objetivos y tareas trazadas. En este sentido:

1. Se ha diseñado e implementado el Sistema de Almacenamiento de la Información correspondiente al Bloque Quirúrgico Oftalmológico que se encargará de la gestión quirúrgica en hospitales oftalmológicos.
2. Se ha desarrollado una aplicación flexible y capaz de integrarse con el resto de los módulos del sistema.
3. La integridad de los datos almacenados en el sistema se ha garantizado en su totalidad.
4. El sistema se ha diseñado de forma tal que es configurable para cada uno de los usuarios del mismo.
5. La seguridad del sistema se maneja a nivel de gestor de base de datos.

RECOMENDACIONES

1. Continuar agregando nuevos subservicios a los ya existentes en el Sistema de Almacenamiento de la Información correspondiente al Bloque Quirúrgico Oftalmológico.
2. Realizar pruebas de concurrencia y pruebas de carga al sistema.

REFERENCIAS BIBLIOGRÁFICAS.

- 1 Delgado Ramos Ariel y Vidal Ledo Maria. Informática en la salud pública cubana. 2006 [extraído el 17 de febrero de 2006]; desde:
http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm#cargo#cargo.
- 2 Ledo, M. V. (2007). "El desarrollo de la Sociedad de la Información en Cuba." Extraído el 14 de mayo de 2007, desde
http://www.cecam.sld.cu/pages/rcim/revista_9/articulos_hm/alfabetizdigital.htm.
3. Ibídem.
4. Pérez, M.M.L. Miniglosario en inglés y español de términos más útiles en Oftalmología. 27 de abril del 2003 [extraído el 29 de mayo de 2007]; desde:
http://bvs.sld.cu/revistas/aci/vol11_4_03/aci04403.htm.
5. Desconocido. Conceptos previos oftalmológicos. 2007 febrero 2005 [extraído el 29 de abril de 2007]; desde: <http://www.avianconsultores.es/jmpascualpascual/lasik/enfermedades.html>.
6. Franckowiak, D.F.Q. Fisiología ocular. [Extraído el 29 de abril de 2007]; desde:
http://www.sld.cu/galerias/pdf/sitios/oftalmologia-fac/fisiologia_ocular.pdf.
7. GARCIA, L.R.M.M., Diseño de bases de datos. 1999, Ciudad Habana. Pág. 12.
8. Hansen, G.W.H.y.J.V., Diseño y administración de Bases de Datos. 2da ed: Prentice Hall. P.12-16.
9. Fuentes Mainoldis y Ramos Alain, Propuesta de Arquitectura para el Proyecto Hospitales. 2007, Universidad de las ciencias Informáticas: Ciudad habana. p. 9.
10. Ibídem.
11. [extraído el 18 de mayo de 2007]; Sitio Oficial de CASE]. Desde:
<http://www.casestudio.com/enu/default.aspx>.
12. Rational Rose Enterprise. 18 de mayo de 2007 [tomado el 18 de mayo de 2007]; desde:
http://www-306.ibm.com/software/info/ecatalog/es_ES/products/M221280M46834Z27.html.
13. Ibídem.

14. Sistema de gestión de base de datos. [Tomado el 2/02/2007]; desde:
http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos.
15. Fuentes Mainoldis y Ramos Alain, Propuesta de Arquitectura para el Proyecto Hospitales. 2007, Universidad de las ciencias Informáticas: Ciudad habana. p. 9.
16. 29 de abril de 2007 [extraído el 29 de abril de 2007]; Sitio oficial de Oracle]. Desde:
<http://www.oracle.com/global/es.html>.
17. Colaboradores de Wikipedia. Oracle [en línea]. Wikipedia, La enciclopedia libre, 2007 [fecha de consulta: 21 de marzo del 2007]. Disponible en
<http://es.wikipedia.org/w/index.php?title=Oracle&oldid=7671094>.
18. Cuadro de licencias de Microsoft Servidores. [Extraído el 19 de abril de 2007]; Licencia de SQL server 2000]. Desde: http://www.microsoft.com/spain/servidores/cuadro_licencias.msp.
19. Fuentes Mainoldis y Ramos Alain, Propuesta de Arquitectura para el Proyecto Hospitales. 2007, Universidad de las ciencias Informáticas: Ciudad habana. p. 9.
20. Ibídem.
21. [Extraído el 19 de abril de 2007]; Sitio oficial de ActualSoft]. Desde:
<http://www.actualsoft.com.ar/nosotros.htm>.
22. [Extraído el 18 de abril de 2007]; Sitio oficial de Hosisic-V]. Desde:
<https://solutionfinder.microsoft.com/SDK/Solutions/SolutionDetailsView.aspx>.
23. Vanguard, Medical Services for Europe. [Extraído el 28 de abril de 2007] desde:
<http://www.vanguard.de>.
24. Care 2X Módulo Sala de Operaciones. [Extraído el 2 de febrero 2007]; Desde:
<http://care2x.org/index.php>.
25. Opera, sistema de gestión de cirugía. [Extraído el 29 de abril de 2007]; Manual del usuario]. Desde: http://www.chca.ca/nouv_spa.html.
26. Tesis.Quirófanos. [Extraído el 17 de mayo de 2007]; Sitio oficial del producto]. Desde:
http://www.sisinf.com/quiروفans_es.htm.

27. Vazquez Ariel y Macías Yipsy, Análisis y Diseño del Módulo Bloque Quirúrgico Oftalmológico del Sistema de información Hospitalaria. 2007, Universidad de las ciencias Informáticas: Ciudad habana.
28. González, P.F.R., ARQUITECTURAS DESISTEMAS DE BASES DE DATOS. 1999/2000: Universidad de Castilla. 54-57.
29. Mota, Soraya Abad., Entonación de Bases de Datos (Apuntes de electiva libre). Abril Julio 2005.
30. Shasha, D., Database Tuning. Principles, Experiments and Trouble Shooting Techniques: Morgan Kaufmann 2003.
31. Ibídem.
32. Ibídem.
33. Mota, Soraya Abad., Entonación de Bases de Datos (Apuntes de electiva libre). Abril Julio 2005
34. Rumbaugh, J Jacobson, I y Booch, G., El Lenguaje Unificado del Modelado. Pearson Education SA ed. 2000.
35. Date, C.J., Sistema de bases de datos. 2003, La Habana: Editorial Félix Varela. pag18.
36. Márquez, M.M. Reglas de integridad. 2001 [Extraído el 28 de mayo de 2007]; desde: <http://www3.uji.es/~mmarques/f47/apun/node1.html>
37. Ibídem.
38. Mota, Soraya Abad., Entonación de Bases de Datos (Apuntes de electiva libre). Abril Julio 2005
39. Date, C.J., Sistema de bases de datos. 2003, La Habana: Editorial Félix Varela. Pág. 368.
40. Ibídem.
41. Ibídem.
42. Ibídem.
43. Ibídem.
44. Monte, J. L. (2006). "Generadores de códigos." Retrieved 2 de junio de 2007, from <http://moga.awardspace.com/wp-content/uploads/2007/03/gencod.pdf>.

45. CONDICIONES DE USO. [Extraído el 29 de abril de 2007]; Licencia web del producto]. Desde:
<http://www.oracle.com/global/es/corporate/copyright.html>.

BIBLIOGRAFÍA

1. Care 2X Módulo Sala de Operaciones. [Extraído el 2 de febrero 2007]; Desde: <http://care2x.org/index.php>.
2. CONDICIONES DE USO. [Extraído el 29 de abril de 2007]; Licencia web del producto]. Desde: <http://www.oracle.com/global/es/corporate/copyright.html>.
3. Date, C.J., Sistema de bases de datos. 2003, La Habana: Editorial Félix Varela.
4. Delgado Ramos Ariel y Vidal Ledo Maria. Informática en la salud pública cubana. 2006 [extraído el 17 de febrero de 2006]; desde: http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm#cargo#cargo.
5. Franckowiak, D.F.Q. Fisiología ocular. [Extraído el 29 de abril de 2007]; desde: http://www.sld.cu/galerias/pdf/sitios/oftalmologia-fac/fisiologia_ocular.pdf.
6. García, L.R.M.M., Diseño de bases de datos. 1999, Ciudad Habana. Editorial Pueblo y Educación.
7. González, P.F.R., Arquitecturas de Sistemas de Bases de Datos. 1999/2000: Universidad de Castilla.
8. Ledo, M. V. (2007). "El desarrollo de la Sociedad de la Información en Cuba." Extraído el 14 de mayo de 2007, desde http://www.cecam.sld.cu/pages/rcim/revista_9/articulos_hm/alfabetizdigital.htm.
9. Mota, Soraya Abad., Entonación de Bases de Datos (Apuntes de electiva libre). Abril Julio 2005.
10. Opera, sistema de gestión de cirugía. [Extraído el 29 de abril de 2007]; Manual del usuario]. Desde: http://www.chca.ca/nouv_spa.html.
11. Pérez, M.M.L. Miniglosario en inglés y español de términos más útiles en Oftalmología. 27 de abril del 2003 [extraído el 29 de mayo de 2007]; desde: http://bvs.sld.cu/revistas/aci/vol11_4_03/aci04403.htm.
12. Rational Rose Enterprise. 18 de mayo de 2007 [tomado el 18 de mayo de 2007]; desde: http://www-306.ibm.com/software/info/ecatalog/es_ES/products/M221280M46834Z27.html

13. Rumbaugh, J Jacobson, I y Booch, G., El Lenguaje Unificado del Modelado. 2000 Editorial Pearson Education SA.
14. Tesis.Quirófanos. [Extraído el 17 de mayo de 2007]; Sitio oficial del producto]. Desde:
http://www.sisinf.com/quirofans_es.htm.
15. Vanguard, Medical Services for Europe. [Extraído el 28 de abril de 2007] desde:
<http://www.vanguard.de>.

ANEXOS

Anexo1 Estructura anatómica del ojo humano.

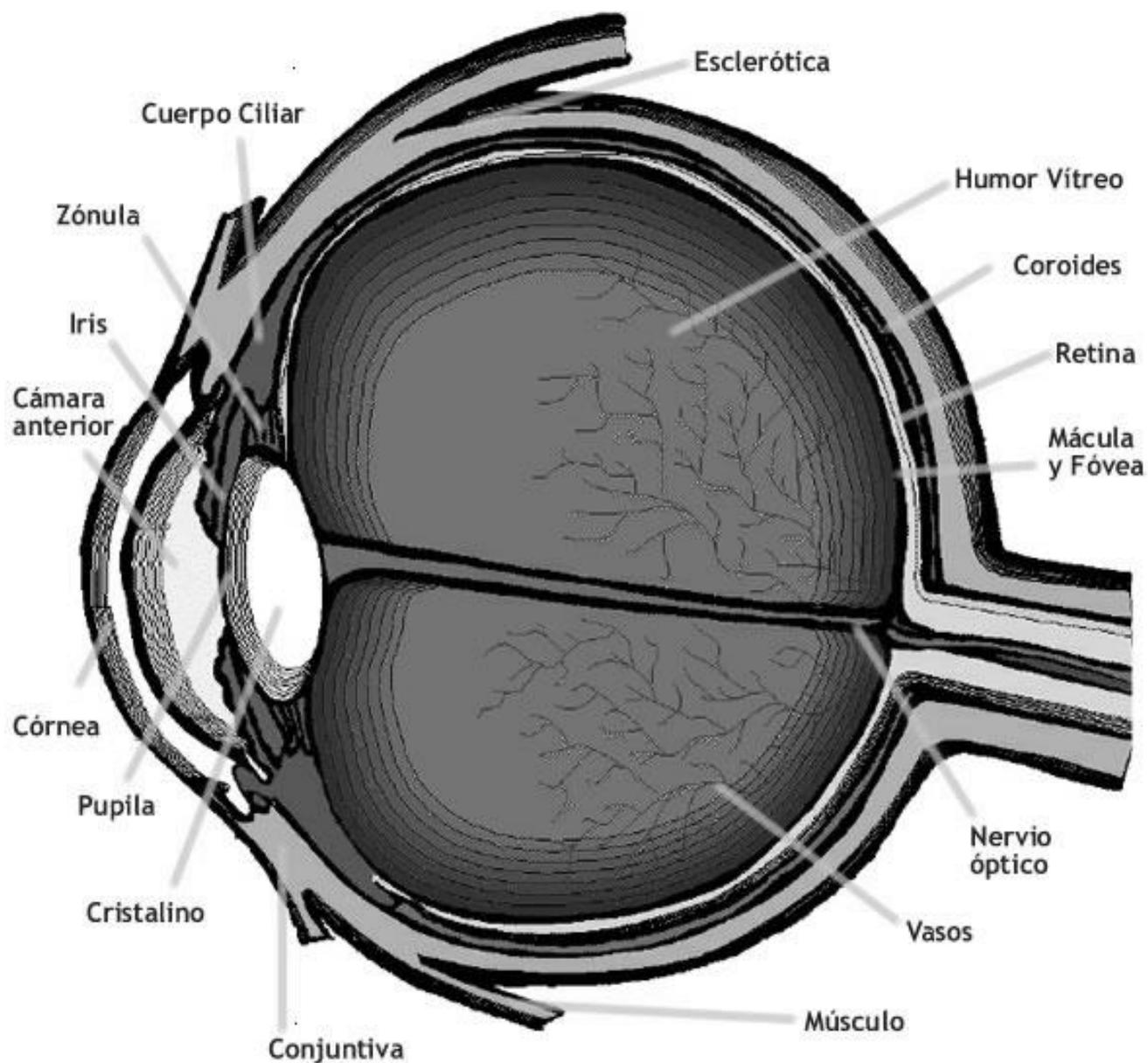


Figura A1.0 Estructura Anatómica del Ojo Humano

Anexo2 Fragmento de licencia de Oracle.

“12.- Restricciones a la Exportación / Compatibilidad Legal.

Ud. no podrá acceder, descargar, utilizar o exportar la Web Site, o el contenido, software, productos o servicios facilitados en la Web Site en violación de las leyes o regulaciones Norteamericanas de exportación, o en violación de cualesquiera otras normas o regulaciones aplicables. Ud. acepta cumplir con todas las leyes, restricciones y regulaciones de exportación de cualquier agencia o autoridad de los Estados Unidos u otros países, y no facilitar directa o indirectamente o hacer disponible de cualquier otra forma los servicios y productos de Oracle en violación de cualquiera de estas restricciones, leyes o regulaciones, o sin todas las aprobaciones necesarias, incluyendo, sin quedar limitadas a ellas, para el desarrollo, diseño, manufactura o producción de armas nucleares, químicas, biológicas o de destrucción masiva. Cuando sea aplicable, Ud. obtendrá y será responsable de todos los gastos derivados de cualesquiera licencias necesarias y/o excepciones respecto a su propio uso de los servicios de Oracle fuera de los Estados Unidos. Ni los servicios de Oracle ni la información o tecnología complementarias podrán ser descargadas o facilitadas o puestas a disposición, ya sea directa o indirectamente, i) en Cuba, Iran, Irak, Libia, Corea del Norte, Sudan, Siria o cualquier otro país sometido a sanciones comerciales por los Estados Unidos, a particulares o entidades controladas por estos países, o a nacionales o residentes de estos países distintos a nacionales que sean residentes permanentes completamente legales de países no sometidos a este tipo de sanciones; o ii) a cualquier persona que se encuentre en la lista de "Specially Designated Nationals and Blocked Persons" del Departamento del Tesoro de los Estados Unidos o en la "Table of Denial Orders" del Departamento de Comercio de los Estados Unidos. Mediante la aceptación de estos términos y Condiciones de Uso, acepta lo anterior y garantiza que Ud. no se encuentra en, bajo el control de, o es un nacional o residente de cualquiera de los países mencionados o se encuentra incluido en cualquiera de las listas especificadas” [45]

Anexo3 Ejemplo de funciones y su optimización.

Función para insertar el nuevo número de la consulta donde se utiliza de forma óptima una consulta anidada.

```
CREATE OR REPLACE FUNCTION "public"."bqg_datos_consulta_ins" (pid_persona integer,
pfecha_consulta timestamptz, pmotivo_consulta varchar, pimpresion_diagnostico varchar,
phistoria_enfermedad_actual varchar, pid_sensibilidad_medicamentosa integer,
pid_tipo_consulta integer) RETURNS integer AS
$body$
declare
    autoincrement INTEGER;
BEGIN
    SELECT max(numero_consulta) INTO autoincrement FROM bqg_datos_consulta WHERE
id_persona = pid_persona;
    if (autoincrement IS NULL) THEN
        autoincrement := 1;
    else
        autoincrement := autoincrement + 1;
    end if;
    INSERT INTO bqg_datos_consulta
    (
        numero_consulta,
        id_persona,
        fecha_consulta,
        motivo_consulta,
```

```

        impresion_diagnostico,
        historia_enfermedad_actual,
        id_sensibilidad_medicamentosa,
        id_tipo_consulta
    )
    values
    (
        autoincrement,
        pid_persona,
        pfecha_consulta,
        pmotivo_consulta,
        pimpresion_diagnostico,
        phistoria_enfermedad_actual,
        pid_sensibilidad_medicamentosa,
        pid_tipo_consulta
    );
    RETURN autoincrement;
end
$body$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT SECURITY INVOKER;

```

Función para seleccionar todos los datos correspondiente a las formulas dióptricas para el cálculo de la refracción. En esta se puede ver la utilización óptima de la sentencia SELECT

```

CREATE OR REPLACE FUNCTION "public"."esp_cfg_formula_dioptrica_selall"
(pid_calculo_refraccion integer) RETURNS SETOF "public"."cfg_formula_dioptrica" AS
$body$
DECLARE

```

```

search cfg_formula_dioptrica%rowtype;
BEGIN
  FOR search IN
    SELECT cfg_formula_dioptrica.id_formula_dioptria,
           cfg_formula_dioptrica.formula,cfg_formula_dioptrica.descripcion
    FROM   cfg_formula_dioptrica join cfg_condicion_formula
    ON     cfg_formula_dioptrica.id_formula_dioptria =cfg_condicion_formula.id_formula_dioptria
    WHERE
           cfg_condicion_formula.id_calculo_refraccion = pid_calculo_refraccion
    LOOP
      RETURN NEXT search;
    END LOOP;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT SECURITY INVOKER;

```

Función para determinar cuales antecedentes patológicos oftalmológicos estuvieron en algún momento presente y en la actualidad no es así. En esta función se trazo una estrategia de consulta en la que se evitó el huso de la clausula DISTINCT

```

CREATE OR REPLACE FUNCTION "public"."esp_bqo_antecedentes_ofthalmologicos"
(pnumero_consulta integer, pid_persona integer) RETURNS SETOF
"public"."bqo_antecedente_patologico_ofthalmologico" AS
$body$
DECLARE
  search bqo_antecedente_patologico_ofthalmologico%rowtype;
BEGIN
  FOR search IN

```

```
SELECT antecedentes_patologicos FROM bqo_antecedente_patologico_ofthalmologico
WHERE id_persona = pid_persona and numero_consulta != pnumero_consulta
      LOOP
      RETURN NEXT search;
END LOOP;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT SECURITY INVOKER;
```

GLOSARIO

1. **INFOMED**: Red telemática del Sistema Nacional de Salud Cubano.
2. **RUP**: Metodología de desarrollo de software de sus siglas en inglés (Rational Unified Process).
3. **UML**: Lenguaje Unificado del Modelado de sus siglas en inglés (Unified Model Language)
4. **CASE**: Ingeniería de Software Asistida por Ordenador de sus siglas en inglés (Computer Aided Software Engineering).
5. **SQL**: Lenguaje de consulta estructurado de sus siglas en inglés (Structured Query Language). Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.
6. **RAM**: Memoria de acceso aleatorio ó memoria de acceso directo de sus siglas en inglés (Random Access Memory Module) Es una memoria volátil, es decir, pierde su contenido al apagar el sistema, se utiliza normalmente como memoria temporal para almacenar resultados intermedios y datos similares no permanentes.
7. **AJAX**: acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas.
8. **HL7**: de sus siglas en inglés (Health Level Seven) es un estándar para el intercambio de información médica.