

Universidad de las Ciencias Informáticas

Facultad # 7



**Título: Sistema de almacenamiento de la información del Módulo Bloque
Quirúrgico del Sistema de información Hospitalaria.**

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

Autora: Yenny Velázquez Correoso.

Tutora: Ing. Ingrid Pérez Tabrane.

Asesora: Lic. Leidy Ileana Osorio Gómez.

Ciudad de La Habana, julio 2007

“Año 49 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autora de la presente tesis y reconozco 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.

Yenny Velázquez Correoso

Ing. Ingrid Pérez Tabrane

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Nombre del tutor: Ing. Ingrid Pérez Tabrane (e-mail: ipereztabrane@uci.cu)

Graduada de Ingeniería Informática en el año 2005 en el Instituto Superior Politécnico "Julio Antonio Echeverría" (CUJAE). Actualmente se desempeña como profesor universitario en la Universidad de las Ciencias Informáticas (UCI) donde ha impartido asignaturas como Teleinformática I, Teleinformática II y Seminario de Tesis. Posee categoría Instructor Recién Graduado. Se desempeñó por un periodo de 6 meses como líder del proyecto de Hospitales de la Facultad 7 de dicha universidad, el cual actualmente se convirtió en la temática productiva de Gestión Hospitalaria (GEHOS), en esta, se encuentra al frente del Módulo de Bloque Quirúrgico del Sistema de Información Hospitalaria desarrollado en la misma.

Nombre del asesor: Lic. Leidy Iliana Osorio Gómez (e-mail: leidy@uci.cu).

Profesor graduado en Licenciatura en Contabilidad y Finanzas en la Universidad de Cienfuegos. Ha impartido las asignaturas de Economía Política del Capitalismo y Economía Política de la Construcción del Socialismo.

AGRADECIMIENTOS

Para empezar, este trabajo no habría sido posible sin la valiosa ayuda de mis familiares.

Agradezco a:

Mi mamá, que siempre ha estado al tanto de mis necesidades, a la que quiero con toda mi vida, con todo su empeño y esfuerzo ha logrado que yo obtenga este título.

Mi papá, que ha sido mi tutor y tiene parte de mi corazoncito.

Genoveva Correoso Reyes, Armando Correoso, Domingo Correoso, Yasianay Cobian Roso, Carmen Roso, Patrocinia, Larinka La O, Lucelia Montoya Reyes, Blanquita Roso, Miladis Reyes, Milagros Reyes, Elena, Nancy, Soraya, Celia, Bertha, Glaubert.

Y a todos aquellos que han aportado su granito de arena para que sea posible este gran sueño:

A mis nuevas y viejas amistades que siempre estuvieron a mi lado dándome un poquito de su alegría y comprensión:

Pedro E. Salas, Yiliam, Keila Torres, Yudelky, Leydanis, Malay, Yusmilaydis, Karelia, Aime, Suleydis.

A Norbeys Coz Cordovi, te quiero mucho y sé que has estado conmigo en los buenos y malos momentos de mi carrera, fortaleciendo aun más lo que sentimos, por tu apoyo y amor. Gracias.

A Kirenia Nariño, por soportarme tanto tiempo y ayudarme en todo lo que ha podido, eres muy especial para mí.

A mis vecinos Emilio, Magalis, Josefa. Ignacia.

Gerardo Morgade, Orlando Souлары Villalón, Daneisy Granado Morgade, Suleydis, Niobis Valiente, Aleida, Rafaela, Cachita, Maikel Leliebre, Katia.

A todos, gracias. Sientan que un pedacito de este trabajo también es suyo.

DEDICATORIA

Esta tesis está dedicada:

A mi mamita. Aunque hemos estado tanto tiempo lejos, no ha sido en vano, porque he cumplido nuestros sueños comunes.

A mi papá, por todo su apoyo y ayuda.

A La Revolución, por permitirme esta oportunidad.

RESUMEN

El objetivo de este trabajo es diseñar la base de datos del Módulo Bloque Quirúrgico que implementa el proyecto Gestión Hospitalaria desarrollado en la Universidad de Ciencias Informáticas, con la finalidad de mejorar la gestión de la información de los procesos vinculados con las intervenciones quirúrgicas que tienen lugar en los hospitales cubanos.

Para lograr este propósito, se utilizó como Gestor de Base de Datos, Postgresql 8.2 que sirve de interfaz entre la base de datos, los usuarios y las aplicaciones que lo manipulan. Todos los accesos a la base de datos se realizan a través del SGBD que mediante su lenguaje de manejo de datos permite a los usuarios la inserción, actualización, eliminación y realizar consultas a la misma en la forma más óptima posible. Otra herramienta utilizada, es Case Studio 2.22, se emplea para el diseño del diagrama entidad relación del Módulo.

Con el diseño de la base de datos propuesta se espera mejorar los procesos que se gestionan en el área quirúrgica de los hospitales del país, perfeccionando los servicios prestados a los pacientes y la organización de los informes que se manipulan en el área. Los datos quedarán protegidos y se limita el acceso, solo a aquellos especialistas autorizados a manejarla, asegurándose que los datos entrados a la base de datos sean verídicos y confidenciales.

PALABRAS CLAVE

Base de datos, Sistemas Gestores de Base de Datos.

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	1
1.1 CONCEPTOS BÁSICOS RELACIONADOS CON EL DOMINIO DEL PROBLEMA.....	1
1.2 HERRAMIENTA A UTILIZAR	3
1.3 LOS SISTEMAS DE GESTIÓN DE BASE DE DATOS (SGBD).....	3
1.3.1 OBJETIVOS (SGBD).....	4
1.3.2 SISTEMAS GESTORES DE BASE DE DATOS LIBRES.	6
1.3.3 SISTEMAS GESTORES DE BASE DE DATOS COMERCIALES.	12
1.4 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN.....	13
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.....	19
2.1 ESTRATEGIA DE INTEGRACIÓN CON OTROS MÓDULOS O SISTEMAS.....	19
2.2 DESCRIPCIÓN DE LA ARQUITECTURA Y FUNDAMENTACIÓN.....	20
2.3 ANÁLISIS Y OPTIMIZACIÓN DE CONSULTAS.....	21
2.4 REQUISITOS FUNCIONALES Y NO FUNCIONALES DEL SISTEMA.....	23
2.5 DIAGRAMA DE CLASES PERSISTENTES OBTENIDO A PARTIR DEL DIAGRAMA DE CLASES DEL DISEÑO.....	29
2.5.1 DESCRIPCIÓN DE LAS CLASES PERSISTENTES.....	30
2.6 DIAGRAMAS ENTIDAD RELACIÓN DE LA BASE DE DATOS.....	46
2.6.1 DESCRIPCIÓN DE LAS TABLAS.....	53
CAPÍTULO 3: VALIDACIÓN DEL DISEÑO REALIZADO	84
3.1 INTEGRIDAD.....	84
3.2 NORMALIZACIÓN DE LA BASE DE DATOS	87
3.3 ANÁLISIS DE LA REDUNDANCIA DE LA INFORMACIÓN	90
3.4 ANÁLISIS DE LA SEGURIDAD DE LOS DATOS	91
3.5 TRAZABILIDAD DE LAS ACCIONES.....	91
CONCLUSIONES.....	93
RECOMENDACIONES	94
REFERENCIA BIBLIOGRÁFICA	95
BIBLIOGRAFÍA.....	96
ANEXOS.....	97
GLOSARIO	99

ÍNDICE DE TABLAS

DESCRIPCIÓN DE LAS TABLAS DE CLASES PERSISTENTES	30
Tabla 2.5.1 Anestesiaconsulta.....	30
Tabla 2.5.2 AgenteInduccionTipo.....	30
Tabla 2.5.3 AnestesiaDosificar	31
Tabla 2.5.4 AnestesiaPosoperatoria	31
Tabla 2.5.5 AnestesiaPreoperatoria.....	32
Tabla 2.5.6 AnestesiaTransoperatoria.....	32
Tabla 2.5.7 AntecedentesPatologicos.....	33
Tabla 2.5.8 AnuncioOperatorio.....	34
Tabla 2.5.9 CE_Complicaciones.....	35
Tabla 2.5.10 Cirugias_Anteriores_Consulta	36
Tabla 2.5.11 Consulta.....	36
Tabla 2.5.12 ConsultaClinico.....	37
Tabla 2.5.13 ConsultaEspecializadaGeneral	38
Tabla 2.5.14 ConsultaPediatria	39
Tabla 2.5.15 ExamenFisico.....	40
Tabla 2.5.16 InformeAnestesia	41
Tabla 2.5.17 InformeOperatorio.....	42
Tabla 2.5.18 Interrogatorio.....	43
Tabla 2.5.19 PlanificacionQuirurgica	43
Tabla 2.5.20 PlanificacionQuirurgicaPersonal.....	44
Tabla 2.5.21 Reanimacion.....	44
Tabla 2.5.22 RegistroTemporalParametros.....	45
DESCRIPCION DE LAS TABLAS DEL MODELO ENTIDAD RELACION	53
TABLA 2.6.1 bqg_agente_induccion_tipo	53
TABLA 2.6.2 bqg_agente_mantenimiento_trans	54
TABLA 2.6.3 bqg_anestesia_consulta.....	54
TABLA 2.6.4 bqg_anestesia_dosificar	55
TABLA 2.6.5 bqg_anestesia_examen_fisico.....	56
TABLA 2.6.6 bqg_anestesia_indicada.....	57
TABLA 2.6.7 bqg_anestesia_informe	57
TABLA 2.6.8 bqg_anestesia_posoperatorio	58
TABLA 2.6.9 bqg_anestesia_pre_operatoria.....	59
TABLA 2.6.10 bqg_anestesia_trans_operatoria	59
TABLA 2.6.11 bqg_antecedentes_patologico	61
TABLA 2.6.12 bqg_antecedentes_no_patologico	62

TABLA 2.6.123 bqq_anuncio_operatorio	63
TABLA 2.6.14 bqq_apto_quirurgico	65
TABLA 2.6.15 bqq_caract_boca_exa_fisico	65
TABLA 2.6.16 bqq_caract_digestivo_exa_fisico.....	66
TABLA 2.6.17 bqq_complicaciones.....	67
TABLA 2.6.18 bqq_consulta_parentesco	67
TABLA 2.6.19 bqq_consulta_seguimiento.....	68
TABLA 2.6.20 bqq_datos_consulta.....	68
TABLA 2.6.21 bqq_embarazo_antecedentes.....	69
TABLA 2.6.22 bqq_estertores_exa_fisico.....	70
TABLA 2.6.23 bqq_examen_fisico	71
TABLA 2.6.24 bqq_habito_toxico_consulta	73
TABLA 2.6.25 bqq_informe_anestesia	74
TABLA 2.6.26 bqq_informe_operatorio	74
TABLA 2.6.27 bqq_interconsulta.....	76
TABLA 2.6.28 bqq_interrogatorio.....	77
TABLA 2.6.29 bqq_planificacion_por_cirujano	78
TABLA 2.6.30 bqq_planificacion_quirurgica	78
TABLA 2.6.31 bqq_posnatales_antecedentes	79
TABLA 2.6.32 bqq_premedicacion.....	80
TABLA 2.6.33 bqq_registro_temporal_parametros	80
TABLA 2.6.34 bqq_tratamiento_actual.....	81
TABLA 2.6.35 bqq_valoracion.....	82

INTRODUCCIÓN

Las computadoras se hacen cada día más importantes en diversas áreas del quehacer humano, ellas han alcanzado incorporarse poco a poco en la educación, comercio, la industria y otras esferas en la sociedad, debido a su rapidez para analizar, procesar y comunicar grandes cantidades de información. Uno de los campos de la actividad humana que se ha visto más beneficiado por la informática es el área de la salud.

Los avances en el cómputo para la medicina son múltiples. Todos ellos son posibles luego de haber convertido los sistemas analógicos a digitales. Esto quiere decir que la forma de un órgano o el ritmo del corazón, son transformados en un código binario, para que las computadoras puedan recibir los datos y procesarlos. *(Bienvenido al Hospital)*

A tono con los avances tecnológicos y científico técnicos alcanzados en la medicina, ya en la década del 70 del pasado siglo, se perfilan los primeros pasos en la creación de un Sistema de Información Médica en el sector de la salud, que posteriormente, se transformó en el actual Sistema de Información Hospitalaria, tan indispensable en la creación de programas y una amplia base de datos, que posibilite un óptimo intercambio de información y mejore la calidad y eficiencia del trabajo en estas instituciones.

Su impacto ha sido muy sobresaliente, ya que busca elevar la calidad de la atención del paciente, de los servicios brindados y aplicar la información obtenida a las áreas de investigación, clínica, docencia y administración. *(Sistema de Información Hospitalaria)*

En correspondencia con los avances operados en este sector en los últimos años, los Sistemas de Información Hospitalaria (SIH) se están convirtiendo en la columna vertebral de la integración de los procesos administrativos y clínicos y en la organización de asistencia sanitaria. En la actualidad la Tecnología de la Informática está indisolublemente ligada a este sector. **Ver Anexo 1.**

La necesidad de mejorar los servicios y encontrar nuevas soluciones a viejos problemas, han hecho insustituible la incorporación de la informática de forma inmediata a las modernas tecnologías del sector de Salud.

Los resultados que en el campo de la salud ha logrado el Estado Cubano, han sido el fruto del trabajo desplegado desde el triunfo mismo de la Revolución en 1959. A pesar de la difícil situación económica que enfrentó en aquel momento, provocada por el deterioro económico heredado del pasado y agravado por el genocida bloqueo impuesto por los EEUU desde ese mismo instante, declaró y aplicó de forma gratuita los servicios médicos y hospitalarios, para darle oportunidades de participación a un amplio sector de la población que en el pasado había estado excluida de estos servicios, por no contar con los recursos económicos que se les exigían para ser atendidos, dedicando especial atención, como ningún otro país, a la

salud del pueblo, a la creación de especialistas, instituciones científicas y técnicas, manteniendo progresivos avances que en la actualidad son compatibles o superiores a los obtenidos en países desarrollados, de tal forma que el sistema de salud cubano tiene una prestigiosa fama reconocida a nivel internacional.

Durante todos estos años el Sistema de Salud Cubano, ha estado empeñado en lograr progresivamente una mayor eficiencia en el mejoramiento de la calidad a la atención médica del pueblo en sentido general, y en particular ha dedicado grandes recursos a la construcción, mantenimiento y reparación de sus instalaciones, al equipamiento con tecnología de punta y a la capacitación del personal, de forma tal que asegure a nivel nacional el incremento de los indicadores de salud de la población y su nivel de participación en estos servicios; además de tomar las medidas para incrementar la atención a los trabajadores y a sus condiciones de trabajo.

A pesar del impacto económico provocado por el Periodo Especial se ha mantenido actualizada en la medida de sus posibilidades, en campo de la Salud, su desarrollo es reconocido por los especialistas de nivel internacional que tienen oportunidad de conocer el trabajo. Cuba es miembro fundador de la Asociación Mundial de Informática Médica (IMIA), de la cual existe una filial en América Latina (IMIALAT).*(Introducción)*

El trabajo desplegado en este frente ha motivado a distintas instituciones a incorporar en el mercado, software de salud, que ayuden a agilizar los procesos de gestión en la salud. Actualmente existen sistemas como: el Galen Lab instalado en los CDI en Venezuela y el SIH instalado en el hospital cubano Hermanos Almejeiras.

La Universidad de Ciencias Informáticas (UCI) se encuentra inmersa en la tarea de crear un sistema que permita la automatización de los hospitales del país, para lograr un mejor funcionamiento de los mismos, con la implementación de un SIH (Sistema de Información Hospitalaria), con la finalidad de satisfacer las necesidades de generación de información, almacenar, procesar y reinterpretar datos médico-administrativos de cualquier institución hospitalaria, permitiendo la optimización de los recursos humanos y materiales, además de minimizar los inconvenientes trámites burocráticos que enfrentan los pacientes.

Dentro de los proyectos que se desarrollan en esta Universidad se encuentra el proyecto GEHOS (Gestión Hospitalaria) que está compuesto por diferentes módulos: Inscripción – Admisión, Historia Clínica, Bloque Quirúrgico Oftalmológico, Farmacia, Laboratorio, Banco de Sangre, Turnos, Seguridad y Bloque Quirúrgico. A este último está dedicado el siguiente trabajo.

El área quirúrgica o el Bloque Quirúrgico de cualquier institución hospitalaria requiere de una correcta gestión y organización, esto es de gran importancia para ofrecer una asistencia de calidad al paciente y lograr su máximo rendimiento. Y es que, esta unidad es uno de los pilares sobre los que se sustenta la actividad del centro de salud y, por tanto, su organización y eficiencia repercute en el funcionamiento del hospital. Esta área exige de una gran inversión tecnológica, de capital humano y una de las zonas con mayor riesgo dentro del hospital.

De hecho, las organizaciones de salud tienen que dedicar una gran parte de sus esfuerzos a la implantación de programas que mejoren la calidad y la seguridad clínica. La complejidad de la gestión del área quirúrgica ha obligado a desarrollar nuevos modelos de gestión, centrados en la autogestión y en la gestión por procesos. Estos cambios organizacionales se han dejado notar en los profesionales implicados, que deben adaptarse a las nuevas estructuras y a la incorporación de las nuevas tecnologías, que no sólo facilitan la gestión sino que contribuyen a la mejora del rendimiento del quirófano.

En Cuba, la información estadística relacionada con los procesos quirúrgicos, se llevan a cabo de forma manual y por consiguiente requieren de un conjunto de documentos que reflejen los datos necesarios de los pacientes, lo que complica y demora el trabajo de los médicos. A esto se le suma los errores humanos que pueden ocurrir, sobre todo con el empleo de modelos y planillas que complican aún más estas tareas.

Dado el incremento de la técnica de computación en esta rama y con el objetivo de buscar la excelencia en el óptimo empleo de los recursos, en ocasiones son usados software, que no cumplen con todos los requerimientos necesarios que se enmarcan en estos complejos procesos que se desarrollan, omiten datos imprescindibles que pueden entorpecer el resultado a lograr.

Por otra parte la incorporación de las nuevas tecnologías de punta a los procesos quirúrgicos, ha generado su especialización, multiplicando su complejidad y el número de intervenciones. La sustitución de la cirugía tradicional por Cirugía Mayor Ambulatoria (CMA) o Cirugía sin Ingreso, contribuye a que el tiempo de estancia de los pacientes en los hospitales se reduzca, pero los documentos a llenar son los mismos. Lo que genera un mayor volumen de trabajo a estos especialistas, sobre todo si se tiene en cuenta que son éstos los únicos responsables de realizar las indicaciones de los complementarios y de llenar los documentos como: anuncio operatorio, informe operatorio y otros, no existiendo otra instancia o persona que valide las mismas; y dado el caso que se extravíe uno de estos documentos, se entorpece el proceso y se retrasa la acción quirúrgica del paciente.

De igual forma, hay que tener en cuenta a los pacientes que deben ser intervenidos de urgencia, que el proceso manual puede provocar retardo en la atención rápida que debe brindarse en estos casos.

Otro elemento a tener en cuenta es el relacionado con la historia clínica del paciente, que en la actualidad cuando es intervenido en un centro de salud de Provincia o Municipio, al ser intervenido fuera de estos, hay que ejecutar de nuevo todos los procedimientos, al no contar actualmente con una base de datos intra-hospitalaria de fácil acceso que le dé seguimiento a su enfermedad o estado de salud.

A todo esto se suma que todo el trabajo del Bloque Quirúrgico del Hospital, debe estar precedido de una exhaustiva planificación de los recursos a emplear, tales como personal médico y técnico, ropa lisa y de cuerpo, camas, instrumental, medicamentos y otros, que aseguren el correcto desarrollo de las intervenciones planificadas en cada jornada de trabajo.

Con lo anteriormente expuesto se plantea el siguiente **problema científico**: ¿Cómo automatizar la gestión de la información referente a los procesos vinculados con las intervenciones quirúrgicas que tienen lugar en los hospitales cubanos?

El **objeto de estudio** está relacionado con los procesos vinculados con las intervenciones quirúrgicas.

Como **campo de acción** se define:

Procesos vinculados con las intervenciones quirúrgicas que tienen lugar en los hospitales cubanos.

El **objetivo general** es diseñar una base de datos que mejore la gestión de la información referente a los procesos vinculados con las intervenciones quirúrgicas que tienen lugar en los hospitales del país.

Las **tareas** que se llevarán a cabo para darle cumplimiento a los objetivos trazados son:

1. Estudiar los procesos vinculados con las intervenciones quirúrgicas en los hospitales cubanos.
2. Estudiar los sistemas similares tanto nacionales como internacionales.
3. Definir las herramientas a utilizar para el diseño de la base de datos.
4. Obtener los artefactos de la metodología RUP que describen la base de datos.
5. Analizar la estrategia de integración de la solución con otros módulos o sistemas.
6. Implementar los procedimientos almacenados del Módulo.

7. Hacer una validación teóricamente del diseño de la base de datos propuesta.
8. Realizar una valoración de los resultados y hacer propuestas de iteración en el diseño.

La investigación realizada está comprendida en 3 capítulos que se describen a continuación.

Capítulo 1 Fundamentación Teórica. Se realiza el estado del arte relacionado con el objeto de estudio en cuestión y se conforma una descripción de las herramientas a utilizar en el diseño de la base de datos.

Capítulo 2 Descripción y Análisis de la Solución Propuesta. Se describe la arquitectura a utilizar así como las tablas de las clases persistentes del análisis y las diseñadas en el Modelo Entidad Relación del Módulo Bloque Quirúrgico.

Capítulo 3 Validación Teórica del Diseño. Contiene información referente a los procedimientos a seguir para realizar un diseño de la base de datos lo más óptimo posible, para darle cumplimiento a lo planteado se midieron los siguientes aspectos: integridad, mínima redundancia, trazabilidad, seguridad y normalización de la base de datos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se abordará acerca de los principales conceptos y tecnologías que se utilizan en el diseño de una Base de Datos del Módulo Bloque Quirúrgico. Se propone describir el estado del arte de los Sistemas Gestores de Base de Datos; los diferentes SIH que tienen implementado el Bloque Quirúrgico y las herramientas de base de datos a utilizar en el transcurso de desarrollo del software.

1.1 CONCEPTOS BÁSICOS RELACIONADOS CON EL DOMINIO DEL PROBLEMA.

En el área quirúrgica de un hospital (Bloque Quirúrgico) se realizan varios procesos que en conjunto son vitales para un buen funcionamiento del mismo. Las principales actividades que allí se llevan a cabo son: la consulta especializada, la consulta de anestesia, la intervención quirúrgica y la planificación de las operaciones.

Cuando un paciente es atendido en el área quirúrgica del hospital es preciso registrar en la Historia Clínica cada servicio que este recibe, así como los informes que se generan con el objetivo de realizar un seguimiento del comportamiento del paciente en esta área. También se hace necesario registrar todo lo referente a los procesos pre-operatorio, trans-operatorios y post-operatorio por los que pasa el paciente durante la intervención quirúrgica

La entrada al área quirúrgica del paciente se produce cuando este se presenta a una **consulta especializada** en uno de los servicios que aquí se brindan, el paciente llega con una remisión o un turno médico y es atendido por un especialista, el cual diagnostica al paciente y toma todos los datos personales y patológicos del mismo. Si el diagnóstico realizado por el especialista lleva un tratamiento quirúrgico se emite el anuncio operatorio donde se plasma la fecha de la intervención quirúrgica.

Luego el paciente es remitido a la **consulta de anestesia**, donde el especialista en anestesia realiza un interrogatorio, de ser necesario indica varios análisis y valora si el paciente está apto para ser operado. En esta consulta se comienza a llenar la hoja de anestesia que tendrá el paciente durante todo el proceso de intervención quirúrgica.

Después de ser aprobado por el especialista de anestesia es cuando el paciente entra al **pre-operatorio**, donde el anestesiólogo evaluará en qué condiciones médicas llega el paciente, para lo cual hará un interrogatorio, un examen físico, revisará los exámenes complementarios e interconsultas y evaluará la necesidad o no de realizar otros. Esta valoración pre-anestesia le permitirá optimizar su estado físico previo a la cirugía, de tal manera que pueda enfrentar el procedimiento quirúrgico en las mejores condiciones médicas posibles.

Si no existe ninguna complicación en esta sala que pueda anular el acto quirúrgico, ya el paciente está listo para la **intervención quirúrgica**, que es el conjunto de acciones que se le acometen al paciente en el quirófano, comienza en la sala pre-operatorio donde el anestesista le aplica al paciente la anestesia pre-operatorio.

Luego este es trasladado al salón y se le aplica la anestesia operatoria. Si no existe ninguna complicación se somete a la cirugía propuesta por el cirujano en la consulta especializada. Cuando termina la intervención quirúrgica se llenan los informes finales; informe operatorio que recoge datos específicos de la intervención quirúrgica, y el informe estadístico que recoge todos los materiales que se utilizaron durante el proceso. Estos informes son enviados al departamento de estadística para que quede constancia de que se realizó el acto quirúrgico.

La Planificación quirúrgica, como su nombre lo indica es la programación de las cirugías que se realizan en los diferentes quirófanos que tiene el departamento. Además se encarga de distribuir la cantidad de salones que tiene la dirección por los diferentes servicios que tienen. Es realizada por una comisión integrada por: Jefe Servicio, Jefe Esterilización, Jefe Abastecimiento, Jefe Banco de Sangre, Jefe Enfermería de los salones de operación y el Vise Director Quirúrgico.

Anualmente se realiza un estudio y se dan los salones a los diferentes servicios, teniendo en cuenta la morbilidad (es el estudio de los efectos de una enfermedad en una población).

Mensual y semanalmente el Jefe Servicio puede ajustar el plan según los cambios que puedan ocurrir. En este momento a cada paciente anunciado previamente se le da la fecha definitiva de la operación.

1.2 HERRAMIENTA A UTILIZAR

Una Herramienta Case en Ingeniería de Sistemas Asistida por Ordenador (Computer-Aided Systems Engineering - CASE) es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias del desarrollo de sistemas. Estas herramientas representan la mejor base para el proceso de análisis y desarrollo de software. (*Sistema de gestión de base de datos*)

La herramienta Case Studio, permite diseñar base de datos profesionales. Es la utilizada para el diseño del Módulo de Bloque Quirúrgico en su versión 2.22, su intuitiva interfaz gráfica hace fácil las tareas más complicadas. Mediante Case Studio se puede realizar diagramas de flujo con muy poco tiempo y esfuerzo. Permite la creación visual de los diagramas de entidad relación (DER). Soporta varios formatos de bases de datos, como Oracle, MySQL, Postgresql, Access, etc. Trabaja bajo casi la mayoría de los sistemas operativos: Windows 95/98/Me/NT/2000/XP.

1.3 LOS SISTEMAS DE GESTIÓN DE BASE DE DATOS (SGBD).

Las bases de datos y sus tecnologías tienen un impacto decisivo con el creciente uso de las computadoras. Desempeñan un papel crucial en casi todas las áreas de aplicaciones de los ordenadores, como los negocios, la ingeniería, la medicina, el derecho, la educación, la bibliotecología entre otras esferas.

Las bases de datos representan un conjunto de información almacenada en memoria auxiliar que permite acceso directo, unido a programas que manipulan esos datos.

Es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina es accesible en tiempo real y compatible con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.

Los Sistemas de gestión de base de datos (SGBD), son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Estos incorporan como herramienta fundamental dos lenguajes, para la definición y la manipulación de los datos. El lenguaje de definición de datos (DDL, Data Definition Language) provee de los medios necesarios para definir los datos con precisión, especificando las distintas estructuras. Acorde con el modelo de

arquitectura de tres niveles, habrá un lenguaje de definición de la estructura lógica global, otro para la definición de la estructura interna, y un tercero para la definición de las estructuras externas.

El lenguaje de manipulación de datos (DML, Data Manipulation/ Management Language), es el encargado de facilitar a los usuarios el acceso y manipulación de los datos. Pueden diferenciarse en procedimentales (aquellos que requieren qué datos se necesitan y cómo obtenerlos) y no procedimentales (qué datos se necesitan, sin especificar como obtenerlos), y se encargan de la recuperación de los datos almacenados, de la inserción y supresión de datos en la base de datos, y de la modificación de los existentes. (*Gestion de Datos*)

1.3.1 OBJETIVOS (SGBD).

Existen muchas formas de organizar las bases de datos, pero hay un conjunto de objetivos generales que deben cumplir todas los SGBD, 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 de los SGBD son:

Abstracción de la información. Los SGBD 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.

Independencia. La independencia de los datos 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.

Redundancia mínima. Un buen diseño de una base de datos 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.

Consistencia. 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.

Seguridad. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD 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, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

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

Respaldo y recuperación. Los SGBD deben 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.

Control de la concurrencia. En la mayoría de los entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Es también frecuente que dichos accesos se realicen de forma simultánea, así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

Tiempo de respuesta. Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Ventajas de los SGBD:

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.

Desventajas:

1. El costo de actualización del hardware y software son muy elevados.
2. Costo (salario) del administrador de la base de datos es costoso.
3. El mal diseño de esta puede originar problemas a futuro.
4. Un mal adiestramiento a los usuarios puede originar problemas a futuro.
5. Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.
6. Generan campos vacíos en exceso.
7. El mal diseño de seguridad genera problemas en esta. (*Sistema de gestión de base de datos*)

Las compañías desarrolladoras de software han implementado numerosos SGBD, estos se pueden clasificar en SGBD Libres y SGBD Comerciales atendiendo las diferencias entre utilizar software libre, software propietario o diferentes combinaciones. Entre los principales que se encuentran a nivel mundial se citan los siguientes:

SGBD libres: PostgreSQL, MySQL.

SGBD comerciales: Microsoft SQL Server, Oracle

En consecuencia con lo planteado se describirá estos SGBD con sus principales características para llegar a un consenso y determinar el que cual se utilizará para el diseño de la base de datos del Módulo Bloque Quirúrgico.

1.3.2 SISTEMAS GESTORES DE BASE DE DATOS LIBRES.

Los Sistemas gestores de base de datos están divididos en dos grandes grupos software libre y software propietario.

El software libre (en inglés *free software*) es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente.

El software libre es parte integral en la construcción de una sociedad libre, justa, ética e incluyente en que las personas tengan la posibilidad de ayudarse mutua y solidariamente.

Dentro de esta clasificación se puede encontrar diversos SGBD entre los más comunes que se utilizan a nivel mundial se encuentran PostgreSQL y Mysql.

Postgresql es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRESQL, de la Universidad de Berkeley. El director de este proyecto es el profesor Michael Stonebraker, y fue patrocinado por Defense Advanced Research Projects Agency (DARPA), el Army Research Office (ARO), el National Science Foundation (NSF), y ESL, Inc.

Postgresql es una derivación libre (OpenSource) de este proyecto, y utiliza el lenguaje SQL92/SQL99. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. (*¿Que es Postgres?:*)

Las principales características de PostgreSQL son las siguientes:

1. Implementación del estándar SQL92/SQL99.
2. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
3. Incorpora una estructura de datos array.
4. Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
5. Permite la declaración de funciones propias, así como la definición de disparadores.
6. Soporta el uso de índices, reglas y vistas.
7. Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
8. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Características más importantes a destacar en el SGBD Postgresql 8.2:

1. Atomicidad (Indivisible) es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
2. Consistencia es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
3. Aislamiento es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.
4. Durabilidad es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.
5. Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, entre otros.
6. Documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios.
7. Comunidades muy activas, varias comunidades en castellano.
8. Soporte nativo para los lenguajes más populares de medio: PHP, C, C++, Perl, Python, etc.
9. Soporte de todas las características de una base de datos profesional (triggers, store procedures – funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.)
10. Utilidades para limpieza de la base de datos (Vacuum)
11. Utilidades para análisis y optimización de Querys.
12. Almacenaje especial para tipos de datos grandes (TOAST)
13. Probado con torres de discos duros de alta disponibilidad.
14. Aplicativos de comparación de bases de datos para testear replicaciones.

DBMS Objeto-Relacional

Postgresql aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transactions, optimización de consultas, herencia, y arrays.

Altamente Extensible

PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario.

Soporte SQL Comprensivo

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

Integridad Referencial

Postgresql soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

API Flexible

La flexibilidad del API de Postgresql ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS Postgresql. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

Lenguajes Procedurales

Postgresql tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de Postgresql es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

MVCC

MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que Postgresql usa para evitar bloqueos innecesarios. Si alguna vez ha usado algún DBMS (DataBase Management System) con capacidades SQL, tal como MySQL o Access, probablemente habrá notado que hay ocasiones en que una lectura tiene que esperar para acceder a información de la base de datos. La espera está provocada por usuarios que están escribiendo en la base de datos. Resumiendo, el lector está bloqueado por los escritores que están actualizando registros.

Mediante el uso de MVCC, Postgresql evita este problema por completo. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgresqlQL

mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. Postgresql es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

Cliente/Servidor

Postgresql usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a Postgresql.

Write Ahead Logging (WAL)

La característica de Postgresql conocida como *Write Ahead Logging* incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual se puede restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos. (*Características de PostgreSQL*)

Desventajas de PostgreSQL

Postgresql consume gran cantidad de recursos, tiene un límite de 8 K por fila, aunque se puede aumentar a 32 K, con una disminución considerable del rendimiento y además es de 2 a 3 veces más lento que MySQL. El primer contacto con Postgresql es un poco duro porque la sintaxis de algunos de sus comandos no es nada intuitiva. También resultan engorrosas las pequeñas variaciones que presenta este gestor en algunos de los tipos de datos que maneja. (*Postgres vs MySql*)

Otro SGBD que se encuentra dentro de la clasificación software libre es **MySQL**, sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL,

pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- Se puede utilizar procedimientos almacenados (nuevo en la versión 5).
- Incorpora un conjunto de herramientas para la migración de datos y objetos desde otros sistemas de bases de datos, incluidos Oracle, Microsoft SQL Server y Access (nuevo en la versión 5).

1.3.3 SISTEMAS GESTORES DE BASE DE DATOS COMERCIALES.

El **software comercial** es el que puede ser libre o no, que es comercializado por las compañías que lo producen y estas cobran dinero por el producto, su distribución o soporte.

A continuación se mencionan dos de los SGBD más comercializados en el mundo del software.

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

1. Soporte de transacciones, que no son mas que una interacción con una estructura de datos que, aún siendo compleja y estar compuesta por varios procesos que se han de aplicar uno después del otro, se pretende que sea equivalente a una interacción atómica. Es decir, que se realice de una sola vez.
2. Estabilidad, o sea, que su nivel de fallos disminuye, en dependencia de la estabilidad que se requiera.
3. Escalabilidad, es decir, su posibilidad de estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos, siendo capaz de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes.
4. Es multiplataforma.

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. En los parches de actualización previstos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

Aunque su dominio en el mercado de servidores empresariales, ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS

con licencia libre como Postgresql, MySql o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.(*Oracle*)

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.

Entre sus características figuran:

- 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 DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.
- Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase ASE. Al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

1.4 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN.

En el mundo son diversas las empresas que se dedican al desarrollo de software de salud, posibilitando incorporar en esta rama de la sociedad tecnologías que ayudan mejorar su productividad y brindar facilidades al personal del Hospital, aumentando la eficiencia laboral y la satisfacción del paciente. Las

necesidades y los constantes cambios de la industria de la salud obligan a estas empresas a estar en una constante evolución en las plataformas tecnológicas.

Se describe a continuación algunas empresas y sus productos que están vinculados al campo de acción.

SISinf

Consultores expertos en informática sanitaria (**SISinf**), constituye una empresa especializada en sistemas de información y gestión sanitaria con una dilatada trayectoria en el campo hospitalario.

Es un equipo de profesionales con más de 25 años en el sector, altamente calificados y con una amplia experiencia en la gestión de proyectos, que alcanza desde la consultoría previa hasta a la implementación y soporte posterior. Colaboran en el desarrollo y la implementación de numerosos proyectos, tanto en el ámbito público como en el privado.

Uno de los productos desarrollados por esta empresa es **TESIS-Quirófanos** 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. (*TESIS.Quirófano*)

El producto **TESIS-Quirófanos** está diseñado en SQL Server, como gestor de base de datos.

SIVSA.

SIVSA es una empresa que pertenece al grupo Crohn Technologies y cuenta con más de 15 años de experiencia.

Uno de los productos desarrollados por esta empresa es **Hosic-V**, es un Sistema de Gestión e Información Hospitalaria flexible.

Hosic-V es un sistema informático compuesto por un extenso conjunto de aplicaciones interrelacionadas que controla y gestiona la totalidad de los datos clínicos, logísticos y económicos de un hospital. El funcionamiento del hospital se optimiza y se hace posible el seguimiento de cada una de las áreas y los datos individuales de cada paciente.

Es uno de los sistemas más flexibles porque está constituido por módulos, algunos de los cuales pueden funcionar autónomamente y se adaptan fácilmente a los diversos tipos de organización, es un sistema integrado porque relaciona directamente las Áreas principales de gestión de un hospital (Asistencial, Clínica, Económica, Servicios y Control de Gestión) permitiendo la unicidad en la captura de la información y por tanto una total fiabilidad de la misma.

El módulo de quirófanos se encarga de mantener un control informatizado de los tiempos de ocupación de los quirófanos, de los servicios que los utilizan, de las técnicas a utilizar, de las personas que participan, para conseguir la optimización de la programación de intervenciones tanto de pacientes internos como externos. Así mismo, se puede realizar un control y seguimiento del paciente durante su estancia en el área quirúrgica, de forma que se incorporen a su historial todas aquellas actividades de pre-operatorio, intervención y post-operatorio. También se permite la gestión de las distintas pruebas pre-operatorias que puedan proceder de una petición en lista de espera. (*Producto*)

VANGUARD

El Grupo **VANGUARD** es un consorcio internacional de alta tecnología con empresas filiales dentro de Europa (entre otros países, en Francia, Suiza, Gran Bretaña y Polonia) y en Asia (China). La empresa se ha especializado en servicios de alta tecnología en el sector de la salud con el fin de elevar el progreso y la calidad en los suministros médicos.

VANGUARD ofrece a los hospitales soluciones de software modulares e integradas para una amplia planificación y documentación de todos los procesos y flujos de producción relevantes para el bloque quirúrgico. Unos componentes perfectamente adaptados entre sí, tecnologías innovadoras que proporcionan a sus operaciones de suministro un respaldo informático preparado para el futuro.

VANGUARD Optim OTM ofrece un sistema de planificación y documentación en tiempo real para el bloque quirúrgico que es fácil de integrar. Proporciona así las bases 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: todas las operaciones en torno al bloque quirúrgico, incluidas las administrativas y logísticas como, por ejemplo, la provisión de recursos necesarios, la planificación del

volumen de personal y la administración de almacén, pueden registrarse y controlarse de forma centralizada.(*VANGUARD Optim OTM*)

Este sistema esta implementado en el gestor de base de datos, SQL Server.

CHCA

CHCA, esta empresa dedica sus esfuerzos a esta tarea desde 1992, está comprometida en la búsqueda de soluciones a las necesidades de sus clientes. La idoneidad y la experiencia del equipo de colaboradores de **CHCA**, la alta calidad de sus servicios y las inversiones continuas en investigación y desarrollo, garantizan que sus productos respondan con eficacia y eficiencia a las necesidades de una clientela exigente. Su objetivo principal es el establecimiento de un lazo de confianza entre **CHCA** y sus consumidores.

Opera, Sistema de Gestión de Cirugía, es un sistema que brinda a los profesionales de la salud y administradores todas las informaciones clínicas y financieras de cada intervención quirúrgica, planificada, real o realizada. Opera soporta la toma de decisiones y la administración general del bloque operatorio, gracias a la producción de una información real y precisa.

El paquete informático de **Opera** incluye los módulos para la planeación de las intervenciones, la administración del material, los exámenes de preadmisión, la documentación del tiempo de procedimientos por cada intervención (Pre-operatorio, Operatorio, Post-operatorio con sala de recuperación), las notas clínicas de la intervención y el resumen de todo procedimiento. Opera es una fórmula perfecta en la administración del bloque operatorio, ya que permite una información eficaz a un costo mínimo.

- Asegura una utilización óptima de las salas de cirugía y del tiempo disponible
- Optimiza la administración de recursos
- Planea el material requerido por cada intervención de acuerdo con la programación de las intervenciones y no por el consumo promedio
- Calcula los costos de funcionamiento por cada procedimiento
- Mejora la disponibilidad de recursos
- Permite un seguimiento eficaz de todo procedimiento en la sala de cirugía(*Opera, Sistema de Gestión de la Cirugía*)

NIS

NIS, Nursering Information System en español Sistema de Información de Enfermería es un sistema de gran actividad e interacción con otros sistemas desarrollados por la misma empresa, no solo con el Clínico, sino también con el Administrativo y con el de Laboratorios. Integra al personal de enfermería a la sistematización de procesos, agilizando y liberando la carga de trabajo para la atención con calidad al paciente.

Este sistema permite agilizar y controlar las funciones operativas, además de automatizar la gestión de la información en los diferentes módulos administrativos que apoyan a la organización, optimizando la productividad y el uso de recursos para apoyar la toma de decisiones.

Se encarga de la administración de los quirófanos, programación de cirugías, realización de cirugías, generación de cargos, modificación e impresión de reportes de cirugías realizadas, cirugías programadas y globales.

Como gestor de base de datos utiliza; SQL Server 7. (*Sistema de Información Integral Hospitalario áreas de negocio 2004*)

Care 2X

Care 2X es 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 (SO) es el encargado de documentar las operaciones quirúrgicas (cirugía, anestesia, enfermería, material, medicinas, etc.), planificar servicio en SO, funciones de búsqueda y archivo, sistema de documentación CIE (Clasificador Internacional de Enfermedades) internacional, planifica operaciones quirúrgicas y documentar los procesos de tras/post operatorio con diagnósticos y procedimientos quirúrgicos. (*Información sobre Care2x*)

Es importante destacar que los software de salud analizados anteriormente tienen importantes funciones para la gestión de los servicios brindados en el Bloque quirúrgico de hospitales a nivel internacional, que

permiten la determinación de los recursos necesarios a emplear en la ejecución y funcionamiento del Salón de Operaciones, tanto humanos especializados, como en medicamentos e instrumental médico que se usa para el trabajo en cada paciente, de igual forma se mide o puede determinarse la cantidad de energía consumida en el mismo. Muchos de estos sistemas permiten determinar el costo o valor de una intervención quirúrgica, que en los países capitalistas se le cobra al paciente.

Sin embargo como en Cuba es gratuita la asistencia médica, en ocasiones se piensa que no es viable hacer estos cálculos, subestimando la importancia de los costos. Estos cálculos no se estaban haciendo hasta el año 2006, que se tomó la medida de incluirlos en el Producto Interno Bruto, al igual que los de Educación, para determinar realmente el crecimiento económico del país por año; cabe recordar que los gastos en el Quirófano, corren a cargo del presupuesto aprobado al hospital en el año. Por lo que se considera que deben quedar recogidos todos estos datos tal como lo hacen los países capitalistas pero no para cobrarle estos servicios al paciente sino para tener una estadística de los gastos del país en este sector.

En este capítulo se ha analizado como el proyecto GEHOS se basa en la política que lleva el país, de desarrollar sus productos informáticos en software libre, es por ello que los arquitectos eligieron, el Gestor de Base de Datos, Postgresql 8.2. La utilización de estas tecnologías, brinda muchas ventajas, se tiene la libertad de hacer modificaciones en el código fuente del programa. De esta forma se enriquece este renglón en la economía del país, que es la producción de software y se apoya la lucha contra la brecha digital, que afecta a los países subdesarrollados. Además se utilizó como herramienta CASE Studio 2.22 para el diseño del Diagrama de Entidad Relación del Módulo.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

En el siguiente capítulo se aborda el tema de las integraciones que presenta el módulo Bloque Quirúrgico con otros sistemas, también el tipo de arquitectura utilizada en el diseño de la base de datos.

Se describe de una forma detallada las principales tablas del Modelo Entidad Relación y las del diseño de Clases Persistentes del desarrollo del software.

2.1 ESTRATEGIA DE INTEGRACIÓN CON OTROS MÓDULOS O SISTEMAS.

La seguridad y confidencialidad de los datos en los hospitales, especialmente en el Bloque Quirúrgico, son aspectos que van de la mano, unidos intrínsecamente a la atención de salud que se brinda a la población, de ahí la importancia y el crédito que otorgan los profesionales de la salud a los registros que disponen, de manera que la seguridad se relaciona con la protección física de la información, incluida la protección contra la pérdida accidental, así como, contra la alteración no autorizada; en tanto que la confidencialidad, se relaciona con la garantía que solo las personas con responsabilidad por las tareas clínicas tengan acceso a la información de los pacientes.(LEDO)

Al Módulo Bloque Quirúrgico le es necesario obtener información de casi todos los módulos que comprende el proyecto GEHOS, para su correcto funcionamiento, porque el área quirúrgica comprende casi todos los procesos que se realizan en el hospital.

Este toma información del Módulo Inscripción Admisión, de la tabla ia_datos_persona, donde se identifican los datos de la persona que son necesarios para el especialista.

También necesita del Módulo Configuración 40 tablas, estas tablas son nomencladores, donde se codifica información referente a los médicos, enfermeros y funcionarios, los datos generales de los usuarios del sistema, a los diagnósticos, a los procedimientos quirúrgicos, las vías de acceso, la clasificación de la operación, la evaluación de la operación, entre otros. Contienen una lista donde se describen cualidades, estados, tipos de clasificaciones y características específicas que puede presentarse al paciente, los cuales son necesarios para establecer una afección, una enfermedad que presenta una persona en dependencia de la consulta que se encuentre.

Otro de los Módulos de los que se toma información es de Farmacia de la tabla fm_producto que identifica el tipo de medicamento.

La integración del sistema, se realizó a través del negocio, para garantizar una correcta protección de los datos, pues por cuestiones de seguridad de la información no está permitido que entre las bases de datos haya ningún tipo de comunicación a no ser a través del negocio.

2.2 DESCRIPCIÓN DE LA ARQUITECTURA Y FUNDAMENTACIÓN.

Una de las principales motivaciones que se toma en cuenta en el desarrollo de sistemas de bases de datos es el deseo de integrar los datos operacionales y proporcionar un acceso controlado a estos. Para lograrlo se rigen de normas que más tarde se complementan para construir la arquitectura de base de datos a utilizar.

La arquitectura de un sistema de base de datos está influenciada en gran medida por el sistema informático en el que se ejecuta. Existen varias formas de diseñar una base de datos, ellas se modelan en dos tipos de categorías: distribuidas y centralizadas.

Como decisión de los arquitectos del proyecto GEHOS se determinó utilizar como arquitectura la centralizada. Posibilitando que desde un servidor central se procesen los datos de todos los módulos del proyecto, existiendo un administrador en este servidor central que tenga un control de todas las base de datos que comprende el sistema, permitiendo acelerar las actividades del sistema de base de datos utilizado, proporcionando a las transacciones respuestas más rápidas, así como la capacidad de ejecutar más transacciones por segundo. Las consultas pueden procesarse de manera que se explote el paralelismo ofrecido por el sistema informático subyacente.

La arquitectura centralizada es la más clásica. En ella, el SGBD está implantado en una sola plataforma u ordenador 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.

Los sistemas de bases de datos centralizados se ejecutan en un único sistema informático sin interaccionar con ninguna otra computadora. Tales sistemas comprenden el rango desde los sistemas de bases de datos monousuario ejecutándose en computadoras personales hasta los sistemas de bases de datos de alto rendimiento ejecutándose en grandes sistemas. (*Arquitectura de Sistemas de Base de Datos*)

El proyecto GEHOS está compuesto por varios módulos que comprenden los procesos de gestión en los hospitales cubanos, cada uno con sus características específicas. Esto trae consigo que se realicen restricciones en cuanto al acceso a los datos que aquí se manejan. Un médico determinado no puede tener acceso a toda la información del software sino al módulo en el cual trabaja, o al que se le ha permitido su entrada.

De la forma que se modelará la arquitectura centralizada en el proyecto es la siguiente; se tendrá un servidor central que contenga todas las bases de datos de los Módulos del proyecto GEHOS, incluyendo la del Bloque Quirúrgico. Existirá un administrador que es el encargado de gestionar toda la información que se manipule en este servidor y es el responsable de establecer las restricciones de seguridad. Donde cada trabajador desde su puesto de trabajo accederá solo a la aplicación o a la instancia del software, al que se le ha permitido su entrada. De tal forma que la información que se entre a la base de datos sea verídica y confidencial.

El sistema presenta un estricto control de acceso que permitirá a cada usuario acceder sólo a los datos autorizados. Para ello, el usuario debe autenticarse, y en dependencia del grupo al que pertenece, dispondrá de acceso a la información. De esta forma, los datos de los pacientes quedan protegidos y se limita sólo a aquellos especialistas autorizados a manejarla.

2.3 ANÁLISIS Y OPTIMIZACIÓN DE CONSULTAS.

Un aspecto importante en el rendimiento de las aplicaciones está muy relacionado con un buen diseño de las tablas y consultas que ésta contiene y la selección adecuada de los tipos de datos lo que conllevará a ocupar menos espacio y un rendimiento mayor debido al ahorro de tiempo en realizar conversiones entre los diferentes tipos de datos.

En bases de datos relacionales el lenguaje de consultas SQL es el más utilizado por los programadores y desarrolladores a nivel internacional para obtener información desde la base de datos. La complejidad que pueden alcanzar algunas consultas puede ser tal, que el diseño de una de ellas puede tomar un tiempo considerable, obteniendo no siempre una respuesta óptima.

Por tanto, la optimización de consultas es el proceso en el cual se pretende mejorar los tiempos de respuesta en un sistema de gestión de bases de datos relacional. Es por ello que hay que tener en cuenta que los lenguajes de consulta relacionales son no procedimentales, es decir, que el usuario solo indica cual es el resultado que desea obtener y no el camino de acceso en la base de datos para llegar a dicho

resultado. Esta navegación automática permite el desarrollo de sistemas que evalúen y mejoren las sentencias de consulta realizadas por los usuarios.

A continuación se presentan algunos aspectos específicos donde se puede mejorar el rendimiento de una consulta:

- La cláusula `DISTINCT` es costosa de ejecutar porque generalmente involucra un ordenamiento de las tuplas resultantes para eliminar duplicados. Es necesario analizar si realmente hace falta usar esa cláusula.
- 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.
- Las tablas temporales pueden bajar el rendimiento de la BD, pero en ocasiones tienen 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.
- Otro uso beneficioso de las tablas temporales es para evitar el uso de la cláusula `ORDER BY` que puede ser costosa.
- Las condiciones de `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.
- Es preferible no usar la cláusula `HAVING` si la condición deseada se puede expresar en la cláusula `WHERE`.
- 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, posiblemente esto es relevante para `joins` que involucran cinco tablas o más.

- El uso de vistas pueden causar una ejecución ineficiente de consultas. Muchas veces la ejecución de consultas sobre las tablas base es más eficiente. (*Entonación de Bases de Datos*)

Para obtener mejoras en el rendimiento de las consultas, se tuvieron en cuenta los aspectos antes citados, al no hacer uso de clausuras que en ocasiones pueden resultar costosas y obtener resultados no deseados. Las consultas que utiliza el Módulo no contienen las restricciones antes mencionadas, las utilizadas son sencillas, y satisfacen los requisitos funcionales del sistema que se plantearán posteriormente.

2.4 REQUISITOS FUNCIONALES Y NO FUNCIONALES DEL SISTEMA.

Un requerimiento es una necesidad documentada sobre el contenido del software, es una forma o funcionalidad de un producto o servicio. En la ingeniería de sistemas o la ingeniería de software se usan los requerimientos en un sentido formal, se utilizan como datos de entrada en la etapa de diseño del producto.

Requerimiento Funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. Son complementados por los requerimientos no funcionales, que se enfocan en cambio en el diseño o la implementación. (*Introducción*)

Para el buen funcionamiento del Software, en el Bloque Quirúrgico de los hospitales cubanos es ineludible cumplir una serie de requisitos funcionales que el sistema necesita para tener un correcto desarrollo del mismo, los cuales se describirán a continuación.

Requerimientos Funcionales

RF1 Crear Anuncio Operatorio.

RF1.1 Registrar datos del paciente.

RF1.2 Registrar Datos del Acto Quirúrgico.

RF1.3 Registrar datos del personal.

RF1.4 Registrar posible consumo de la operación.

RF 2 Modificar Anuncio Operatorio

RF 3 Buscar Anuncio Operatorio.

RF3.1 Visualizar Anuncio Operatorio.

RF3.2 Listar Anuncio Operatorio.

RF 4 Cambiar estado del Anuncio Operatorio.

RF4.1 Suspender Anuncio Operatorio.

RF4.2 Registrar causa de suspensión.

RF4.3 Posponer Anuncio Operatorio.

RF 5 Gestionar antecedentes del paciente.

RF5.1 Modificar antecedentes del paciente.

RF5.2 Visualizar antecedentes del paciente.

RF5.3 Buscar antecedentes del paciente.

RF 6 Gestionar Hoja del especialista

RF6.1 Crear Hoja del especialista

RF6.2 Modificar Hoja del especialista

RF6.3 Buscar Hoja del especialista

RF 7 Gestionar Hoja de Anestesia.

RF7.1 Crear Hoja de Anestesia.

RF7.2 Modificar Hoja de Anestesia

RF7.3 Visualizar Hoja de Anestesia.

RF 8 Gestionar planilla de Informe Operatorio.

RF8.1 Crear Informe Operatorio.

RF8.2 Listar Informe Operatorio.

RF8.3 Visualizar Informe Operatorio.

RF 9 Gestionar Planificación Quirúrgica.

RF9.1 Crear Planificación Quirúrgica.

RF9.2 Modificar Planificación Quirúrgica.

RF9.3 Eliminar Planificación Quirúrgica.

RF 9.4 Visualizar Planificación Quirúrgica.

RF 10 Buscar Planificación

RF 11 Buscar Consultas

RF 12 Buscar oportunidad

RF 13 Gestionar hojas de consultas de Aprobación de Cirugía

RF13.1 Crear hojas de consultas de Aprobación de Cirugía

RF13.2 Modificar hojas de consultas de Aprobación de Cirugía

RF 14 Realizar Acto Quirúrgico

RF14.1 Registrar datos del paciente en el pre_operatorio.

RF14.2 Registrar datos del paciente en el trans_operatorio.

RF14.3 Registrar datos del paciente en el post_operatorio

RF 15 Realizar remisión al paciente

Requerimientos no funcionales que debe cumplir el sistema.

RNF 1- Requerimientos de rendimiento

El tiempo de respuesta de una petición al servidor debe ser rápido para la toma de decisiones.

RNF 2- Requerimientos de soporte

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

RNF 3-Requerimientos políticos, culturales y legales.

Las herramientas propuestas para la BD deberán responder a los intereses de la Constitución de la Republica de Cuba, asimismo no existirán prioridades en el servicio según el nivel social, cultural o étnico. No se permitirán la divulgación de los datos. Todos los procesos responderán a las resoluciones establecidas por el Ministerio de Salud Publica cumpliendo las normas instituidas en el Código Penal.

RNF 4-Requerimientos de confiabilidad

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

RNF 5-Requerimientos de hardware

- Requerimientos para un servidor: 512Mb RAM (Recomendado 1Gb RAM o superior), 1GHz o superior, 60Gb HDD.
- Periféricos: Mouse y Teclado.
- Tarjeta de Red.
- 1GB MB de RAM
- Capacidad de Almacenamiento que garantice soporte para la BD.

RNF 6-Requerimientos de software

La BD debe corre en sistemas operativos Windows 98 o superior y sistemas Unix, GNU-Linux. Para sistemas Windows se deber tener instalado Microsoft Framework 2.0 y en sistemas Linux la plataforma Mono 1.2 o superior.

Postgresql 8.2, Slony-I 1.2.9

RNF 7-Restricciones en el diseño y la implementación

- Se utilizará Case Studio para el diseño y modelado de la BD.
- La BD será implementada utilizando como SGBD Postgresql 8.2.
- La comunicación de las terminales clientes con el servidor será a través de conexiones de fibra óptica.

RNF 8-Requisitos de Independencia:

- La independencia de los datos 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.
- Hay que obtener un buen diseño de la base de datos para lograr evitar la aparición de información repetida o redundante por lo que seria muy factible lograr una redundancia nula.

RNF 9-Requisitos de Consistencia:

En aquellos casos en los que no se ha logrado esta redundancia nula, es necesario vigilar y validar por códigos de programación que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

RNF 10-Requisitos de Seguridad y Privacidad

La información almacenada en la BD tendrá un gran valor por lo que se debe garantizar que esta información almacenada en la BD se encuentre 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.

Se deben restringir las funcionalidades mediante roles de usuarios garantizando que la información sea accesible al usuario autorizado.

- Los usuarios de la BD solamente tendrán acceso a las tablas y BD a las cuales se les designe según el rol que desempeña, no se utilizarán usuarios con privilegios administrativos para realizar las conexiones entre servidores.
- Se registrarán y auditarán las trazas dejadas por los usuarios al realizar las diferentes operaciones en el sistema. La programación de las auditorias será programada según corresponda.

RNF 11-Requisitos de Integridad.

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

RNF 12-Requisitos de Respaldo y Recuperación.

La BD deben proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ella, y de restaurar a partir de estas copias los datos que se hayan podido perder. Se le debe dar instrucciones a los motores para que estos generen respaldos o Backup automáticos.

RNF 13-Requisitos de Tiempo de respuesta.

Se debe lograr el menor tiempo de respuesta posible minimizando el tiempo que el SGBD tardará en darnos la información solicitada y en almacenar los cambios realizados, porque así se evita la ocupación de recursos del Servidor de BD y los mismos tiempos de respuestas de los Usuarios.

2.5.1 DESCRIPCIÓN DE LAS CLASES PERSISTENTES.

DESCRIPCIÓN DE LAS TABLAS DE CLASES PERSISTENTES

Tabla 2.5.1 Anestesiaconsulta

Nombre: AnestesiaConsulta	
Tipo de clase: Entidad	
Atributo	Tipo
persona	string
tipoConsulta	string
fechaConsulta	string
imprecisionDiagnostica	string
motivoConsulta	string
historiaEnfermedad	string
sensibilidadMedicamentosa	string
consultaPadre	string
listHbitoToxico	string
Para cada responsabilidad:	
Nombre:	Consulta
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase AnestesiaConsulta, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase AnestesiaConsulta, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.2 AgenteInduccionTipo

Nombre: AgenteInduccionTipo	
Tipo de clase: Entidad	
Atributo	Tipo
medicamento	string

tipoAgenteInduccion	string
agenteInduccion	string
Para cada responsabilidad:	
Nombre:	AgenteInduccionTipo
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase AgenteInduccionTipo, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase AgenteInduccionTipo, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.3 AnestesiaDosificar

Nombre: AnestesiaDosificar	
Tipo de clase: Entidad	
Atributo	Tipo
via	string
medicamento	string
dosis	string
fechaMedicacion	Date time
Para cada responsabilidad:	
Nombre:	AnestesiaDosificar
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase AnestesiaDosificar, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase AnestesiaDosificar, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.4 AnestesiaPosoperatoria

Nombre: AnestesiaPosoperatoria
Tipo de clase: Entidad

Atributo	Tipo
fechaSalida	Date time
observaciones	integer
Para cada responsabilidad:	
Nombre:	AnestesiaPosoperatoria
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase AnestesiaPosoperatoria, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase AnestesiaPosoperatoria, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.5 AnestesiaPreoperatoria

Nombre: AnestesiaPreoperatoria	
Tipo de clase: Entidad	
Atributo	Tipo
FechaSalida	Date time
funcionarioEnfermera	String
Para cada responsabilidad:	
Nombre:	ExamenFisico
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase Anestesia Pre-operatoria, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase Anestesia Pre-operatoria, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.6 AnestesiaTransoperatoria

Nombre: AnestesiaTransoperatoria	
Tipo de clase: Entidad	
Atributo	Tipo

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

fechaHoraLlegada	Date time
complicaciones	String
intubacion	String
funcionarioCirujano	String
funcionarioAnestesista	String
funcionarioEnfermera	String
funcionarioTecnico	String
salon	String
turnoQuirurgico	String
fechaHoraInicioAnestesia	Date Time
complejidadInduccion	String
fechaHoraInicioCirugia	Date time
fechaHoraFinCirugia	Date time
fechaHoraSalida	Date time
agenteMantenimiento	String
Para cada responsabilidad:	
Nombre:	AnestesiaTransoperatoria
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase Anestesia Transoperatoria, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase Anestesia Transoperatoria, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.7 AntecedentesPatologicos

Nombre: AntecedentesPatologicos	
Tipo de clase: Entidad	
Atributo	Tipo
parentesco	String
Enfermedad	String

patologia	String
Para cada responsabilidad:	
Nombre:	AntecedentesPatologicos
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase AntecedentesPatologicos, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase AntecedentesPatologicos, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.8 AnuncioOperatorio

Nombre: AnuncioOperatorio	
Tipo de clase: Entidad	
Atributo	Tipo
persona	String
salon	String
medicamentos	String
subservicio	String
funcionarioCirujano	String
funcionarioAnestesista	String
funcionarioTecnico	String
reintervencion	Boolean
especificacionCirugia	String
fechaHoraCirugia	Date
turnoQuirurgico	Integer
observacionesGenerales	String
biopsia	Boolean
biopsia	Boolean
sangre	String
plasma	String

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

anestesiaAnuncio	String
diagnosticoAnuncio	String
causaSuspension	String
tecnicasQuirurgicas	String
tipoAnestesia	String
producto	String
Para cada responsabilidad:	
Nombre:	AnuncioOperatorio
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase AnuncioOperatorio, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase AnuncioOperatorio, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.9 CE_Complicaciones

Nombre: CE_Complicaciones	
Tipo de clase: Entidad	
Atributo	Tipo
complicacion	String
listComplicacionDiagnostico	String
Para cada responsabilidad:	
Nombre:	CE_Complicaciones
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase CE_Complicaciones, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase CE_Complicaciones, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.10 Cirugias_Anteriores_Consulta

Nombre: Cirugias_Anteriores_Consulta	
Tipo de clase: Entidad	
Atributo	Tipo
tipoCirugia	String
tiempoTranscurrido	Date
observaciones	String
Para cada responsabilidad:	
Nombre:	Cirugias_Anteriores_Consulta
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase Cirugias_Anteriores_Consulta, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase Cirugias_Anteriores_Consulta, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.11 Consulta

Nombre: Consulta	
Tipo de clase: Entidad	
Atributo	Tipo
persona	String
tipoConsulta	String
fechaConsulta	Date
imprecionDiagnostica	String
motivoConsulta	String
historiaEnfermedad	String
sencibilidadMedicamentosa	String
listHabitoToxico	String

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

consultaPadre	String
Para cada responsabilidad:	
Nombre:	Consulta
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase Consulta, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase Consulta, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.12 ConsultaClinico

Nombre: ConsultaClinico	
Tipo de clase: Entidad	
Atributo	Tipo
persona	String
tipoConsulta	String
fechaConsulta	Date
imprecisionDiagnostica	String
motivoConsulta	String
historiaEnfermedad	String
sensibilidadMedicamentosa	String
listHabitoToxico	String
consultaPadre	String
interConsulta	Boolean
indicaciones	String
tratamientoRealizar	String
apto	Boolean
especificacionesApto	String
conductaSeguir	String
textoInterconsulta	String

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

complementario	String
Para cada responsabilidad: Consulta	
Nombre:	ExamenFisico
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase Consulta, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase Consulta, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.13 ConsultaEspecializadaGeneral

Nombre: ConsultaEspecializadaGeneral	
Tipo de clase: Entidad	
Atributo	Tipo
persona	String
tipoConsulta	String
fechaConsulta	Date
imprecionDiagnostica	String
motivoConsulta	String
historiaEnfermedad	String
sensibilidadMedicamentosa	String
listHabitoToxico	String
consultaPadre	String
cirugiasAnteriores	String
Para cada responsabilidad:	
Nombre:	Consulta
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase Consulta, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de	Un método <i>set</i> para cada atributo privado de la clase Consulta I, que permite

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

métodos set	<i>cambiar</i> el estado de cada uno de estos atributos.
-------------	--

Tabla 2.5.14 ConsultaPediatra

Nombre: ConsultaPediatra	
Tipo de clase: Entidad	
Atributo	Tipo
Persona	String
tipoConsulta	Boolean
fechaConsulta	Date
imprecisionDiagnostica	String
motivoConsulta	String
historiaEnfermedad	String
sensibilidadMedicamentosa	String
listHabitoToxico	String
consultaPadre	String
interConsulta	Boolean
Indicaciones	String
tratamientoRealizar	String
Apto	Boolean
especificacionesApto	String
conductaSeguir	String
textoInterconsulta	String
complementario	String
Apgar	String
pesoNacer	Double
vacunaActualizada	Boolean
listAnPatPosnatales	String
listAnPatEmbarazo	String
Para cada responsabilidad:	

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Nombre:	Consulta
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase Consulta, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase Consulta, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.15 ExamenFisico

Nombre: ExamenFisico	
Tipo de clase: Entidad	
Atributo	Tipo
persona	String
mucosas	String
caractTejidoSubcutaneo	String
traquea	String
piel	String
piel	String
caractCara	String
patologiaCara	String
caractNasales	String
ruidoCardiaco	String
cuello	String
murmulloVesicular	String
talla	String
peso	Double
frecuenciaRespiratoria	String
soplos	String
frecuenciaCardiaca	String
tensionArtMin	String

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

tensionArtMax	String
signosFocalizacion	String
signosMeningeos	String
caractBoca	String
caractBoca	String
caractDigestivo	String
estertores	String
Para cada responsabilidad:	
Nombre:	ExamenFisico
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase ExamenFisico, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase ExamenFisico, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.16 InformeAnestesia

Nombre: InformeAnestesia	
Tipo de clase: Entidad	
Atributo	Tipo
fechaInforme	Date
persona	String
Para cada responsabilidad:	
Nombre:	InformeAnestesia
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase InformeAnestesia, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase InformeAnestesia I, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.17 InformeOperatorio

Nombre: InformeOperatorio	
Tipo de clase: Entidad	
Atributo	Tipo
tiposIncisiones	String
electiva	Boolean
tipoSutura	String
clasificacionCirugia	String
salon	String
funcionarioAnestesia	String
funcionarioCirujano	String
funcionarioEnfermera	String
funcionarioTecnico	String
persona	String
turnoQuirurgico	Integer
especificacion	String
biopsia	Boolean
indicacionesPosoperatorias	String
observaciones	String
conteoCompreses	Integer
instrumentalCompleto	Booelan
cirugiaTrabajosa	Booelan
causas	String
anestesia	String
Para cada responsabilidad:	
Nombre:	InformeOperatorioGeneral
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase InformeOperatorio, que permite <i>obtener</i> el estado de cada uno de estos atributos.

Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase InformeOperatorio, que permite <i>cambiar</i> el estado de cada uno de estos atributos.
------------------------------------	---

Tabla 2.5.18 Interrogatorio

Nombre: Interrogatorio	
Tipo de clase: Entidad	
Atributo	Tipo
sistemaOrgano	Date
Afeccion	String
Para cada responsabilidad:	
Nombre:	Interrogatorio
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos get	Un método <i>get</i> para cada atributo privado de la clase Interrogatorio, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos set	Un método <i>set</i> para cada atributo privado de la clase Interrogatorio, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.19 PlanificacionQuirurgica

Nombre: PlanificacionQuirurgica	
Tipo de clase: Entidad	
Atributo	Tipo
salon	String
subservicio	String
cantCirugias	String
tiempoQuirurgico	String
fechaPlanificacion	Date
Para cada responsabilidad:	
Nombre:	PlanificacionQuirurgica
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Descripción general de métodos <i>get</i>	Un método <i>get</i> para cada atributo privado de la clase PlanificacionQuirurgica, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos <i>set</i>	Un método <i>set</i> para cada atributo privado de la clase PlanificacionQuirurgica, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.20 PlanificacionQuirurgicaPersonal

Nombre: PlanificacionQuirurgicaPersonal	
Tipo de clase: Entidad	
Atributo	Tipo
funcionarioCirujano	String
cantPersonal	Integer
Para cada responsabilidad:	
Nombre:	PlanificacionPorCirujano
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos <i>get</i>	Un método <i>get</i> para cada atributo privado de la clase PlanificacionQuirurgicaPersonal, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos <i>set</i>	Un método <i>set</i> para cada atributo privado de la clase InformeAnestesia I, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.21 Reanimacion

Nombre: Reanimacion	
Tipo de clase: Entidad	
Atributo	Tipo
descripcion	String
valor	String
Para cada responsabilidad:	
Nombre:	Reanimacion
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase

Descripción general de métodos <i>get</i>	Un método <i>get</i> para cada atributo privado de la clase Reanimacion, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos <i>set</i>	Un método <i>set</i> para cada atributo privado de la clase Reanimacion, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

Tabla 2.5.22 RegistroTemporalParametros

Nombre: RegistroTemporalParametros	
Tipo de clase: Entidad	
Atributo	Tipo
tipoRegistro	String
presionArtMin	String
presionArtMax	String
frecuenciaCardiacaMin	String
frecuenciaCardiacaMax	String
saturacionParcialOxigeno	String
Para cada responsabilidad:	
Nombre:	RegistroTemporalParametros
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Descripción general de métodos <i>get</i>	Un método <i>get</i> para cada atributo privado de la clase RegistroTemporalParametros, que permite <i>obtener</i> el estado de cada uno de estos atributos.
Descripción general de métodos <i>set</i>	Un método <i>set</i> para cada atributo privado de la clase RegistroTemporalParametros, que permite <i>cambiar</i> el estado de cada uno de estos atributos.

2.6 DIAGRAMAS ENTIDAD RELACIÓN DE LA BASE DE DATOS.

El Modelo Entidad Relación se ha convertido en estándar para el diseño de bases de datos relacionales, es el utilizado para describir el diseño de la base de datos del Bloque Quirúrgico.

El Modelo Relacional es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Cada relación es una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante consultas que ofrecen una amplia flexibilidad y poder para administrar la información. Una BD relacional es un conjunto de dos o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave. A esta manera de construir bases de datos se le denomina modelo relacional. (*Sistema de Información Integral Hospitalario áreas de negocio 2004*)

De forma general el modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información que está contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos.

Este modelo de datos tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario casual de la BD. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

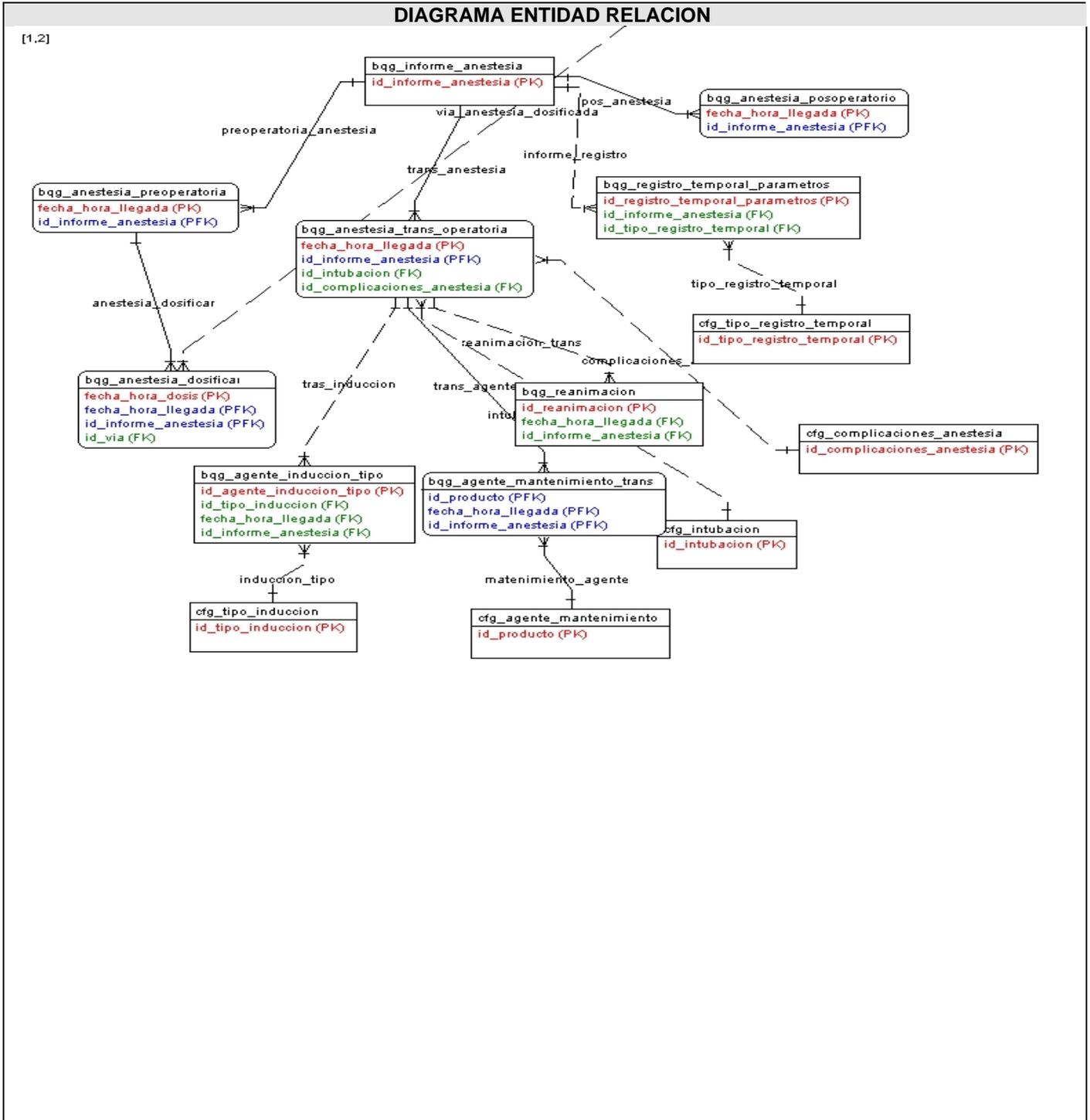
A continuación se detalla del Módulo Bloque Quirúrgico las descripciones de algunas tablas del diagrama entidad relación y sus atributos que se generaron de las Clases Persistentes, luego de haber elaborado el Diagrama de Clases de Diseño del software.

Este diagrama está compuesto por 85 tablas, 44 de ellas guardan información relacionada con informes que se manipulan en el área quirúrgica del hospital entre otros documentos que ayudan a la gestión de los procesos que allí se realizan. Las 41 restantes tablas son nomencladores guardan información relacionada

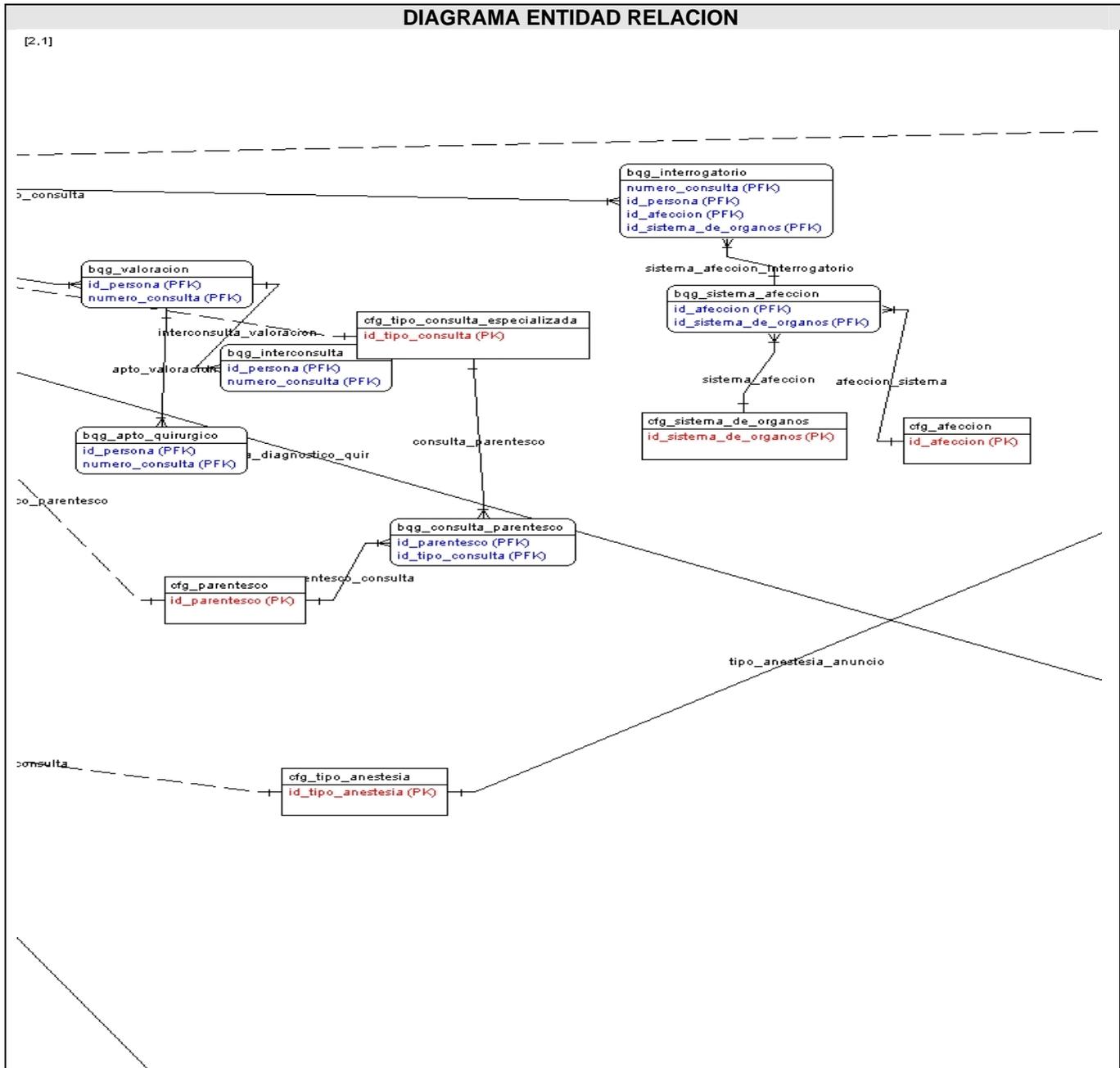
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

con el tipo de anestesia a utilizar, tipo de consulta, tipo de incisión entre otras características imprescindibles que están estrechamente relacionadas con este proceso.

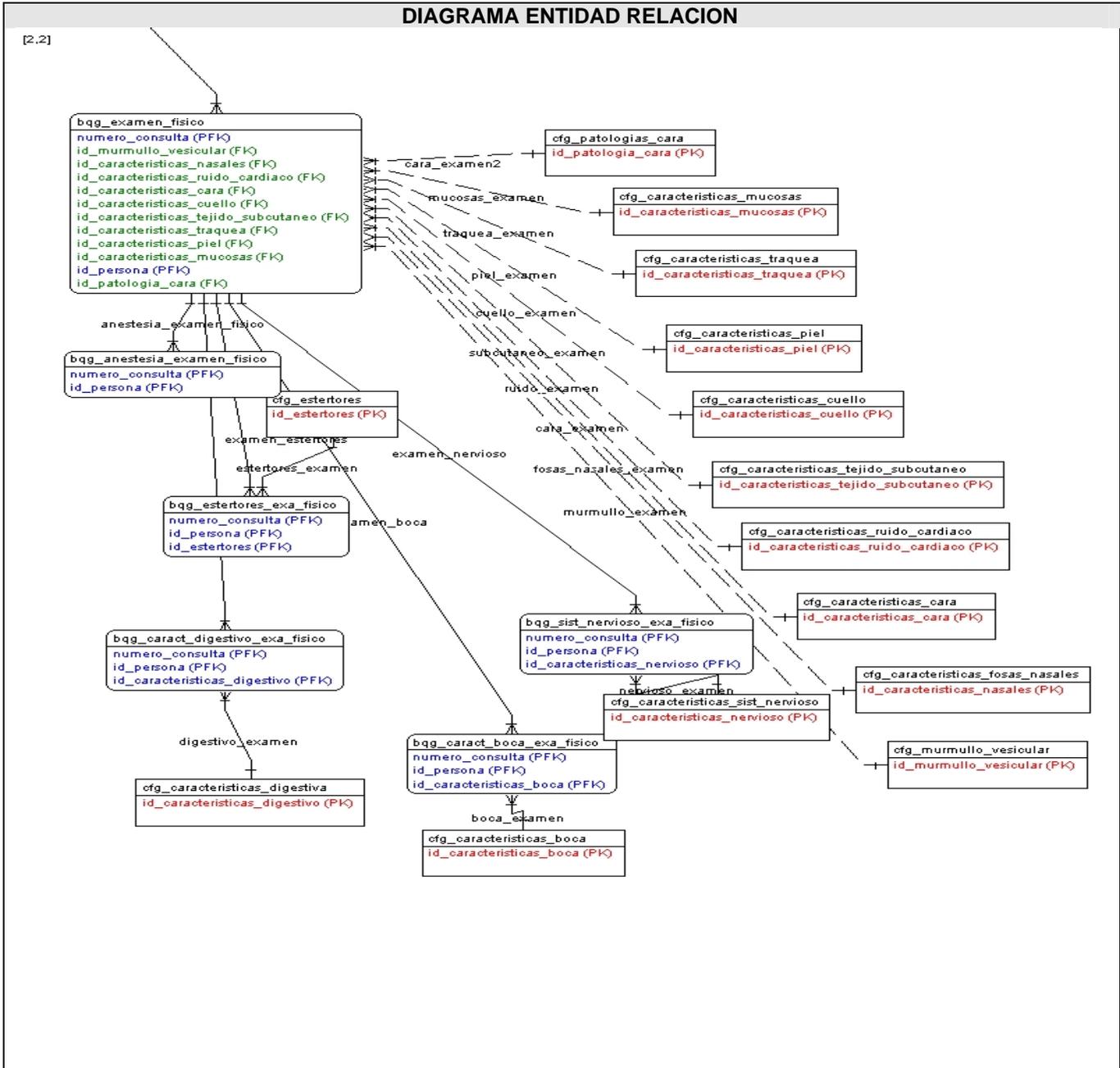
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA



2.6.1 DESCRIPCIÓN DE LAS TABLAS.

DESCRIPCIÓN DE LAS TABLAS DEL MODELO ENTIDAD RELACIÓN

TABLA 2.6.1 bqg_agente_induccion_tipo

Nombre: bqg_agente_induccion_tipo		
Descripción: Esta tabla se genera de la relación de mucho a mucho entre las tablas cfg_tipo_induccion con bqo_anestesia_trans_operatoria y contiene el tipo de inducción que fue realizado al paciente cuando que se le suministra la anestesia.		
Atributo	Tipo	Descripción
id_agente_induccion_tipo (PK)	Integer	Es la llave primaria de esta tabla, su Identificador único.
id_tipo_induccion(FK)	Integer	Llave foránea que proviene de la tabla cfg_tipo_induccion que identifica el tipo de inducción que se utiliza.
fecha_hora_llegada (FK)	Timestamp	Es llave foránea proviene de la tabla bqg_anestesia_trans_operatoria que contiene la fecha y hora en la que se le suministra al paciente la anestesia trans-operatoria.
id_informe_anestesia (FK)	Integer	Llave foránea proviene de la tabla bqg_anestesia_trans_operatoria.
id_producto(FK)	Integer	Llave foránea que proviene del Módulo de Farmacia, de la tabla fm_producto, que contiene una lista con todos los medicamentos existentes.

TABLA 2.6.2 bqq_agente_mantenimiento_trans

Nombre: bqq_agente_mantenimiento_trans		
Descripción: Es una tabla que se generó de la relación de mucho a mucho entre las tablas bqq_anestesia_trans_operatoria y cfg_agente_mantenimiento.		
Atributo	Tipo	Descripción
id_agente_mantenimiento (PFK)	Integer	Llave primaria que proviene de la tabla cfg_agente_mantenimiento que especifica el tipo de agente mantenimiento a utilizar.
fecha_hora_llegada(PFK)	Timestamp	Es llave primaria que proviene de la tabla bqq_anestesia_trans_operatoria.
id_informe_anestesia(PFK)	Integer	Esta llave proviene de la tabla bqq_anestesia_trans_operatoria

TABLA 2.6.3 bqq_anestesia_consulta

Nombre: bqq_anestesia_consulta		
Descripción: Contiene datos que recoge el especialista en anestesia a cada paciente relacionado con la anestesia.		
Atributo	Tipo	Descripción
id_persona (PFK)	integer	Es llave primaria que proviene de la tabla ia_persona del módulo de Inscripción Admisión que identifica a la persona que se está tratando
numero_consulta (PFK)	Integer	Es llave primaria que proviene de la tabla

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		bqg_datos_consulta
id_riesgo_quirurgico(FK)	Integer	Es llave foránea que proviene de la tabla cfg_riesgo_quirurgico que contiene una lista de riesgos quirúrgicos existentes durante una operación
historia_anestesia	varchar	Guarda información relacionada con la anestesia y su seguimiento en esta consulta.
datos_importantes	varchar	Guarda los datos importantes que el médico estima importantes reflejar en esta consulta.
estado_fisico	integer	Es una valoración que realiza el especialista y se establece por una clasificación de 1 al 7.

TABLA 2.6.4 bqg_anestesia_dosificar

Nombre: bqg_anestesia_dosificar		
Descripción: Es una tabla que se generó de la relación de mucho a mucho entre las tablas bqg_anestesia_pre_operatoria y cfg_via, que contiene los datos referente a la dosis de los medicamentos que se le proporciona al paciente en el Pre_Operatorio y por las vías que se les suministrará.		
Atributo	Tipo	Descripción
fecha_hora_dosis(PK)	timestamp	Es la llave primaria, el identificador único de esta tabla.
fecha_hora_llegada (PFK)	timestamp	Es llave primaria que proviene de la tabla bqg_datos_consulta.
id_informe_anestesia (PFK)	integer	Es llave primaria que proviene de la tabla bqg_informe_anestesia que describe el comportamiento de la anestesia en el paciente durante los procesos de pre-operatorio, trans-operatorio y post-

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		operatorio.
id_via (FK)	integer	Es llave foránea que proviene de la tabla cfg_via, que identifica los tipos de vías por los que se puede suministrar determinado medicamento.
id_producto(FK)	integer	Es llave foránea que proviene de la tabla fm_producto del Módulo Farmacia el cual identifica los diferentes medicamentos existentes.
dosis	varchar	Es la dosis en la que se distribuirá el medicamento al paciente.

TABLA 2.6.5 bqq_anestesia_examen_fisico

Nombre: bqq_anestesia_examen_fisico		
Descripción: Contiene los datos de la valoración que realiza el especialista en su consulta referente a el examen físico del paciente.		
Atributo	Tipo	Descripción
numero_consulta (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos_consulta que contiene los datos de la consulta especializada que realiza cada médico, ya sea el cirujano, pediatra etc.
id_persona (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos_consulta.
malampy	varchar	Guarda el tipo de clarificación existente que se le da a la cirugía del 1 al 4.
asa	varchar	Guarda el riesgo quirúrgico que presenta un paciente y se da por un tipo de clasificación del 1 al 5.

TABLA 2.6.6 bqq_anestesia_indicada

Nombre: bqq_anestesia_indicada		
Descripción: Esta tabla se generó de la relación de mucho a mucho entre las tablas bqq_anestesia_consulta y cfg_tipo_anestesia que contiene los datos relacionados con el tipo de anestesia que propone el anestesista para la operación del paciente.		
Atributo	Tipo	Descripción
id_anestesia_indicada(PK)	varchar	Es la llave primaria, el identificador único de esta tabla.
numero_consulta (FK)	varchar	Es llave foránea que proviene de la tabla bqq_datos_consulta que contiene los datos de la consulta especializada que realiza cada médico, ya sea el cirujano, pediatra etc.
id_persona (FK)	integer	Es llave foránea que proviene de la tabla ia_persona del Módulo Inscripción Admisión que identifica a la persona que se está consultando.
id_tipo_anestesia (FK)	integer	Es llave foránea que proviene de la tabla cfg_tipo_anestesia que contiene una lista de los tipos de anestesias, las cuales son suministradas al paciente antes y durante la operación.

TABLA 2.6.7 bqq_anestesia_informe

Nombre: bqq_anestesia_informe
Descripción: Contiene los datos referente al informe final de anestesia de cada paciente.

Atributo	Tipo	Descripción
id_anestesia_informe (PK)	integer	Es la llave primaria, el identificador único de esta tabla.
id_informe_operatorio (FK)	integer	Esta llave proviene de la tabla bqq_informe_operatorio
id_producto		Es llave foránea que proviene de la tabla fm_producto del Módulo Farmacia la cual identifica los diferentes medicamentos existentes.

TABLA 2.6.8 bqq_anestesia_posoperatorio

Nombre: bqq_anestesia_posoperatorio		
Descripción: Contiene información del paciente referente a la anestesia suministrada durante el proceso de posoperatorio.		
Atributo	Tipo	Descripción
fecha_hora_llegada (PK)	timestamp	Es la llave primaria, el identificador único de esta tabla.
id_informe_anestesia (PFK)	integer	Es llave foránea que proviene de la tabla bqq_informe_anestesia que contiene un informe estadístico de lo relacionado con la anestesia pre-operatoria, trans-operatoria y post-operatoria.
hora_salida	timestamp	Es la hora en la que el paciente pasa al post-operatorio.
observaciones	varchar	Estas observaciones las anota el médico presente, donde plasma el estado en que se encuentra el paciente durante el post-operatorio.

TABLA 2.6.9 bqq_anestesia_pre_operatoria

Nombre: bqq_anestesia_pre_operatoria		
Descripción: Contiene los datos referentes a la anestesia que se le suministra al paciente en el pre-operatorio.		
Atributo	Tipo	Descripción
fecha_hora_llegada (PK)	timestamp	Es la llave primaria, el identificador único de esta tabla. Donde guarda la fecha y la hora en la que se le suministra la anestesia al paciente en el pre-operatorio.
id_informe_anestesia (PFK)	integer	Es llave primaria foránea que proviene de la tabla bqq_informe_anestesia.
Id_funcionario (FK)	integer	Es llave foránea que proviene de la Tabla cfg_funcionario del Módulo Configuración, que contiene una lista de todos los trabajadores del hospital.
hora_salida	timestamp	Es la hora en la que el paciente sale del pre-operatorio.

TABLA 2.6.10 bqq_anestesia_trans_operatoria

Nombre: bqq_anestesia_trans_operatoria		
Descripción: Contiene los datos de la reacción del paciente con respecto a la anestesia aplicada durante la operación.		
Atributo	Tipo	Descripción

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

fecha_hora_llegada (PK)	timestamp	Es la llave primaria, el identificador único de esta tabla que contiene la fecha y hora de la llegada al paciente al salón operatorio.
id_informe_anestesia (PFK)	integer	Es llave primaria que proviene de la tabla bqg_informe_anestesia.
id_complicaciones_anestesia (FK)	integer	Es llave foránea que proviene de la tabla cfg_complicaciones_anestesia que contiene los diferentes tipos de complicaciones que puede presentar el paciente con respecto a la anestesia durante la operación
id_intubacion (FK)	integer	Es llave foránea que proviene de la tabla cfg_intubacion que identifica tipos de intubación existentes.
id_funcionario_enfermera (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_enfermera del Módulo Configuración que contiene una lista de todas las enfermeras del hospital.
id_funcionario_cirujano (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_cirujano del Módulo Configuración que contiene una lista con todos los cirujanos del hospital.
id_funcionario_tecnico (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_tecnico del Módulo Configuración que identifica los funcionarios técnicos que trabajan en el hospital.
id_funcionario_anestesista	integer	Es llave foránea que proviene de la tabla cfg_funcionario_anestesista del Módulo

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		Configuración que identifica los anestesiólogos existentes en el hospital.
numero_salon	integer	Guarda el identificador que se le da a cada salón de operación.
turno_quirurgico	integer	Guarda el turno quirúrgico, en el cual se realiza la operación
hora_comienzo_anestesia	timestamp	Guarda la hora en la que se le suministra al paciente la anestesia.
complejidad_induccion	double precision	Guarda si hay o no complejidad en el tipo de inducción suministrada.
hora_inicio_cirugia	timestamp	Guarda la hora en la que se realiza la cirugía.
hora_fin_cirugia	timestamp	Guarda la hora en la que se culmina la cirugía.
hora_salida	timestamp	Hora en la que sale el paciente del salón de operación.

TABLA 2.6.11 bqq_antecedentes_patologico

Nombre: bqq_antecedentes_patologico		
Descripción: Esta tabla que se generó de la relación de mucho a mucho entre cfg_parentesco y cfg_patologia, que contiene los antecedentes patológicos que presenta el paciente y a que familiar corresponde.		
Atributo	Tipo	Descripción
id_parentesco_enfermedades (PK)	integer	Es la llave primaria, el identificador único de esta tabla.
id_parentesco (FK)	integer	Es llave primaria que proviene de la tabla cfg_parentesco que identifica el tipo de

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		parentesco. Ej. madre, abuelo
numero_consulta (FK)	integer	Es llave foránea que proviene de la tabla bqq_datos_consulta.
id_patología (FK)		Es llave foránea que proviene de la tabla cfg_patologia que identifica la patología que presenta el parentesco del paciente.
id_persona (FK)	integer	Es llave foránea que proviene de la tabla bqq_datos_consulta.

TABLA 2.6.12 bqq_antecedentes_no_patologico

Nombre: bqq_antecedentes_no_patologicos		
Descripción: Contiene los datos referente a los tipos de antecedentes no patológicos del paciente.		
Atributo	Tipo	Descripción
id_persona (PFK)	Integer	Es llave primaria que proviene de la tabla bqq_datos_consulta.
numero_consulta (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos_consulta.
id_antecedentes_no_patologicos_parto (FK)	integer	Es llave foránea que proviene de la tabla cfg_antecedentes_no_patologicos_parto que identifica el tipo de antecedente no patológico que se establece durante el parto.
apgar	varchar	Es el examen que se realiza a los recién nacidos.
peso_al_nacer	double precision	Guarda el valor del peso que tuvo el paciente al nacer.
vacuna_actualizada	double	Si esta vacunada o no.

	precision	
--	-----------	--

TABLA 2.6.123 bqq_anuncio_operatorio

Nombre: bqq_anuncio_operatorio		
Descripción: Contiene datos que recoge el cirujano en el Anuncio Operatorio, donde establece la fecha de operación del paciente y otros datos importantes a tener en cuenta en la intervención quirúrgica.		
Atributo	Tipo	Descripción
id_anuncio_operatorio (PK)	integer	Es la llave primaria, el identificador único de esta tabla.
id_persona (FK)	integer	Es llave foránea que proviene de la tabla ia_persona del Módulo Inscripción Admisión identifica a la persona que se está consultando
numero_consulta (FK)	Integer	Es llave foránea que proviene de la tabla bqq_datos_consulta.
id_salon (FK)	integer	Es llave foránea que proviene de la tabla cfg_salon, que identifica el salón, donde se realiza operaciones.
id_producto(FK)	integer	Es llave foránea que proviene de la tabla fm_producto del Módulo Farmacia el cual identifica los diferentes medicamentos existentes.
id_subservicio (FK)	integer	Es llave foránea que proviene de la tabla cfg_subservicio del Módulo Configuración que identifica el tipo de subservicio.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

id_funcionario_anestesista (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_anestesista del Módulo Configuración que identifica el anestesista que participara en la operación.
id_funcionario_cirujano (FK)	Integer	Es llave foránea que proviene de la Tabla cfg_funcionario_cirujano del Módulo Configuración que identifica al cirujano que participará en la operación.
id_funcionario_tecnico (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_tecnico del Módulo Configuración que identifica al técnico que participará en la operación.
reintervencion	double precision	Si el paciente se ha realizado anteriormente operaciones o no.
especificaciones_cirugia	varchar	Son apuntes que realiza el especialista relacionado con la cirugía propuesta.
fecha_hora_cirugia	timestamp	Se especificará la hora y la fecha en la que se realizará la cirugía.
turno_quirurgico	integer	Es el número del turno en el que se le realizara la operación al paciente.
observaciones_generales	varchar	Son apuntes que escribe el médico en la consulta que serán necesarios para evaluaciones posteriores.
biopsia	double precision	Si el paciente necesita realizarse un examen de biopsia o no.
radiologia	double precision	Si el paciente necesita realizarse un examen de radiología o no.
sangre_necesaria	integer	Guarda la cantidad de sangre que necesita el paciente para la operación
plasma_necesario	integer	Guarda la cantidad de plasma necesario

		para la operación.
--	--	--------------------

TABLA 2.6.14 bqq_apto_quirurgico

Nombre: bqq_apto_quirurgico		
Descripción: Contiene los datos que recoge el especialista en su consulta relacionados si el paciente está en condiciones de ser intervenido quirúrgicamente o no.		
Atributo	Tipo	Descripción
id_persona (PFK)	integer	Es llave primaria que proviene de la tabla ia_datos_persona del Módulo Inscripción Admisión.
numero_consulta (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos_persona.
apto	double precision	Si el paciente está en condiciones de realizarse la intervención quirúrgica o no.
especificaciones_apto	varchar	Se guardan anotaciones que debe escribir el especialista relacionado con el apto quirúrgico.
conducta_a_seguir	varchar	Se guardan las anotaciones que el médico le encomienda al paciente en correspondencia con su diagnostico emitido.

TABLA 2.6.15 bqq_caract_boca_exa_fisico

Nombre: bqq_caract_boca_exa_fisico

Descripción: Es una tabla que se generó de la relación de mucho a mucho entre las tablas <code>cfg_caract_boca</code> y <code>bqg_examen_fisico</code> , que guarda las características del sistema nervioso que presenta el paciente en el examen físico.		
Atributo	Tipo	Descripción
<code>id_caract_boca</code> (PFK)	integer	Es la llave primaria foránea que proviene de la tabla <code>cfg_caract_boca</code> que identifica la característica específica de la boca que tiene el paciente.
<code>numero_consulta</code> (PFK)	integer	Es llave primaria que proviene de la tabla <code>bqg_datos_consulta</code> .
<code>id_persona</code> (PFK)	Integer	Es llave primaria que proviene de la tabla <code>bqg_datos_persona</code> del Módulo Inscripción Admisión.

TABLA 2.6.16 `bqg_caract_digestivo_exa_fisico`

Nombre: <code>bqg_caract_digestivo_exa_fisico</code>		
Descripción: Es una tabla que se generó de la relación de mucho a mucho de las tablas <code>cfg_caract_digestiva</code> y <code>bqg_examen_fisico</code> que guarda las características digestivas que presenta el paciente en el examen físico que le realiza el especialista.		
Atributo	Tipo	Descripción
<code>numero_consulta</code> (PFK)	integer	Es llave primaria foránea que proviene de la tabla <code>bqg_datos_consulta</code> .
<code>id_persona</code> (PFK)	integer	Es llave primaria foránea que proviene de la tabla <code>bqg_datos_persona</code> del Módulo Inscripción Admisión.
<code>id_caract_digestivo</code> (PFK)	Integer	Es llave primaria foránea que proviene de la tabla <code>cfg_caract_digestivo</code> .

TABLA 2.6.17 bqq_complicaciones

Nombre: bqq_complicaciones		
Descripción: Contiene los datos de las complicaciones que se presentaron durante la operación.		
Atributo	Tipo	Descripción
id_complicaciones	integer	Es llave primaria de esta tabla, su identificador único
id_funcionario (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario.
id_informe_operatorio (PFK)	integer	Es llave primaria foránea que proviene de la tabla bqq_informe_operatorio.
causa	varchar	Se describen las causas de las complicaciones que presenta el paciente ocurridas durante la operación.

TABLA 2.6.18 bqq_consulta_parentesco

Nombre: bqq_consulta_parentesco		
Descripción: Es una tabla que se generó de la relación de mucho a mucho entre las tablas cfg_parentesco y cfg_tipo_consulta que guarda los datos de los tipo de parentescos patológicos que presenta el paciente y a cual consulta corresponde.		
Atributo	Tipo	Descripción
id_tipo_consulta (PFK)	integer	Es llave primaria de la tabla cfg_tipo_consulta
id_parentesco (PFK)	integer	Es llave foránea de la tabla cfg_parentesco.

TABLA 2.6.19 bqq_consulta_seguimiento

Nombre: bqq_consulta_seguimiento		
Descripción Contiene los datos referente al historial de la consulta es decir su seguimiento.		
Atributo	Tipo	Descripción
id_numero_consulta (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos_persona
id_persona (PFK)	integer	Es llave primaria que proviene de la tabla ia_datos_persona.
consulta_padre	integer	Es el número de seguimiento que tiene el paciente en una determinada consulta.

TABLA 2.6.20 bqq_datos_consulta

Nombre: bqq_datos_consulta		
Descripción: Contiene los datos de cada persona referente a una consulta especializada a la cual fue remitido, ya sea el pediatra, anestesista o cirujano.		
Atributo	Tipo	Descripción
numero_consulta (PFK)	integer	Es la llave primaria de esta tabla, Su identificador único, la cual guarda el número de la consulta en la se está atendiendo el paciente.
id_persona (PFK)	integer	Es llave primaria foránea que proviene de la tabla ia_datos_persona del Módulo Inscripción Admisión que identifica a la persona que se está

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		tratando en la consulta.
id_tipo_consulta (FK)	integer	Es llave foránea que proviene de la tabla cfg_tipo_consulta identifica la consulta que se está tratando. Ej. Cirujano, Pediatra.
fecha_consulta	timestamp	Guarda la fecha en la que se realiza la consulta.
motivo_consulta	Varchar	Guarda la valoración que pone el Médico donde describe el motivo por el cual el paciente asiste a la consulta.
impresion_diagnostico	Varchar	Guarda la impresión que tiene el Médico relacionado con el Diagnóstico.
historia_enfermedad_actual	Varchar	Guarda los datos referenciados al tipo de enfermedad que presenta el paciente donde el médico escribe una valoración al respecto.
id_sensibilidad_medicamentosa	Integer	Es llave foránea que proviene de la tabla ia_sensibilidad_medicamentosa del Módulo Inscripción Admisión que identifica la alergia a los medicamentos.

TABLA 2.6.21 bqg_embarazo_antecedentes

Nombre: bqg_embarazo_antecedentes
Descripción: Es una tabla que se generó de la relación de mucho a mucho de las tablas bqg_antecedentes_no_patologicos y cfg_antecedente_no_patologico_embarazo.

Atributo	Tipo	Descripción
cfg_antecedente_no_patologico_embarazo (PFK)	integer	Es llave primaria que proviene de la tabla cfg_antecedente_no_patologico_embarazo, la cual identifica el tipo de antecedente no patológico que presentó el paciente durante el embarazo.
numero_consulta (PFK)	integer	Es llave primaria que proviene de la tabla bqg_datos consulta.
id_persona (PFK)	integer	Es llave primaria que proviene de la tabla bqg_datos consulta.

TABLA 2.6.22 bqg_estertores_exa_fisico

Nombre: bqg_estertores_exa_fisico		
Descripción: Es una tabla que se generó de la relación de mucho a mucho entre las tablas bqg_examen_fisico y cfg_estertores.		
Atributo	Tipo	Descripción
numero_consulta (PFK)	integer	Es llave primaria que proviene de la tabla bqg_datos consulta.
id_persona (PFK)	integer	Es llave primaria que proviene de la tabla bqg_datos consulta.
id_estertores (PFK)	integer	Es llave primaria que proviene de la tabla cfg_estertores que identifica que tipo de sonido que se le escucha en el pulmón al paciente.

TABLA 2.6.23 bqq_examen_fisico

Nombre: bqq_examen_fisico		
Descripción: Contiene los datos de del examen físico que se le realiza al paciente en una consulta especializada.		
Atributo	Tipo	Descripción
numero_consulta (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos consulta
id_persona (PFK)	integer	Es llave primaria foránea que proviene de la tabla bqq_datos consulta
id_caract_mucosas (FK)	integer	Es llave que proviene de la tabla cfg_caracteristica_mucosa que identifica el tipo de mucosa que presenta el paciente. Ej. Húmedas, secas.
id_caract_tejido_subcutaneo (FK)	integer	Es llave foránea que proviene de la tabla cfg_caract_tejido_subcutaneo que identifica el tipo de tejido subcutáneo que presenta el paciente.
id_caract_traquea (FK)	integer	Es llave foránea que proviene de la tabla cfg_caract_traquea que identifica el tipo de característica de la tráquea que presenta el paciente.
id_caract_piel (FK)	integer	Es llave foránea que proviene de la tabla cfg_caract_piel que identifica la característica de la piel que presenta el paciente.
id_caract_cara (FK)	integer	Es llave foránea que proviene de la tabla cfg_caract_cara que identifica la característica de la cara que presenta el paciente.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

id_patologia_cara (FK)	integer	Es llave foránea que proviene de la tabla cfg_patologia_cara que identifica la característica patológica de la cara que presenta el paciente.
id_caract_nasales (FK)	integer	Es llave foránea que proviene de la tabla cfg_caract_nasales que identifica el tipo de característica nasal que presenta el paciente.
id_caract_ruido_cardiaco (FK)	integer	Es llave foránea que proviene de la tabla cfg_caract_ruido_cardiaco que identifica el ruido cardiaco que presenta el paciente.
id_caract_cuello (FK)	integer	Es llave foránea que proviene de la tabla cfg_caract_cuello que identifica la característica del cuello que presenta el paciente.
id_murmullo_vesicular (FK)	integer	Es llave foránea que proviene de la tabla cfg_murmullo_vesicular que identifica el tipo de murmullo vesicular que presenta el paciente.
talla	double precision	La talla que tiene el paciente actualmente
peso	double precision	El peso del paciente actualmente
frecuencia_respiratoria	integer	Guarda las veces que respira el paciente (ciclo de respiración: se contraen y se expanden los pulmones) por unidad de tiempo, normalmente en respiraciones por minuto.
soplos	double precision	Si tiene a tenido o no soplos. Son sonidos que produce la sangre al circular por las cavidades o válvulas del corazón o a través de los vasos sanguíneos próximos al corazón.
frecuencia_cardiaca	integer	Guarda la cantidad de latidos que realiza el

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		corazón por minuto.
tension_arterial_min	integer	Guarda el volumen de sangre mínimo, que bombea el corazón hacia el cuerpo en un minuto.
tension_arterial_max	integer	Guarda el volumen de sangre máximo, que bombea el corazón hacia el cuerpo en un minuto.
signos_focalizacion	double precision	Si presenta o no signos de focalización. Los signos de focalización indican la posible ubicación de la lesión.
signos_meningeos	double precision	Si presenta o no signos meníngeos.

TABLA 2.6.24 bqq_habito_toxico_consulta

Nombre: bqq_habito_toxico_consulta		
Descripción: Es una tabla que se generó de la relación de mucho a mucho de las tablas cfg_habito_toxico y bqq_datos_consulta.		
Atributo	Tipo	Descripción
id_habito_toxico (PFK)	integer	Es llave primaria foránea que proviene de la tabla cfg_habito_toxico donde se identifica el tipo de habito tóxico que presenta el paciente.
numero_consulta (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos consulta
id_persona (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos consulta

TABLA 2.6.25 bqq_informe_anestesia

Nombre: bqq_informe_anestesia		
Descripción: Contiene los datos referente a todo el proceso que se le realiza al paciente, manteniendo un control del mismo durante los proceso de pre-operatorio, la operación y post-operatorio.		
Atributo	Tipo	Descripción
id_informe_anestesia	integer	Es llave primaria de esta tabla su identificador único.
fecha_informe_anestesia	timestamp	Es la fecha en la que se realiza el informe de anestesia.
id_persona	integer	Es llave primaria que proviene de la tabla bqq_datos consulta

TABLA 2.6.26 bqq_informe_operatorio

Nombre: bqq_informe_operatorio		
Descripción: Contiene los datos del informe operatorio que se le realiza al paciente, se elabora una descripción de todo el proceso por el cual transitó el paciente durante la operación, recoge los especialista que intervinieron y otros aspectos significativos.		
Atributo	Tipo	Descripción
id_informe_operatorio (PK)	integer	Es llave primaria de esta tabla su identificador único.
id_salones (FK)	integer	Es llave foránea que proviene de la tabla cgf_salones que identifica el salón que se operó el paciente.
id_anuncio_operatorio (FK)	integer	Es llave foránea que proviene de la tabla

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		bqg_anuncio_operatorio
id_tipo_incisiones (FK)	integer	Es llave foránea que proviene de la tabla cfg_tipo_incisiones que identifica el tipo de incisión que fue realizada al paciente.
id_clasificacion_cirugia_limpieza(FK)	integer	Es llave foránea que proviene de la tabla cfg_clasificacion_cirugia_limpieza que identifica la clasificación de la cirugía realizada.
id_tipo_sutura (FK)	integer	Es llave foránea que proviene de la tabla cfg_tipo_sutura que identifica el tipo de sutura que fue realizada.
id_clas_cirugia_limpieza (FK)	integer	Es llave foránea que proviene de la tabla cfg_tipo_sutura que identifica la clasificación de la cirugía en cuanto al tipo de limpieza.
id_funcionario_anestesista (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_anestesista del Módulo Configuración que identifica los funcionarios que participaron en la operación.
id_funcionario_enfermera (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_enfermera del Módulo Configuración que identifica a las enfermeras que intervinieron en la operación.
id_funcionario_cirujano (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_cirujano del Módulo Configuración que identifica a los cirujanos que intervinieron en la operación.
id_funcionario_tecnico (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_tecnico del Módulo Configuración que identifica a los funcionarios técnicos que intervinieron en la

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		operación.
id_persona (FK)	integer	Es llave foránea que proviene de la tabla bqq_datos_consulta.
turno_quirurgico	integer	Guarda el turno en el cual fue operado el paciente.
especificaciones	varchar	Son especificaciones que del especialista determina plasmar en este informe.
biopsia	double precision	Si necesita realizarse biopsia o no. Es un procedimiento realizado con el propósito de obtener tejido o células del cuerpo para examinarlos con el microscopio.
indicaciones_post_operatorias	varchar	Guarda las indicaciones que plantea el especialista para después que se efectúe el acto quirúrgico.
observaciones	varchar	Guarda las observaciones que el especialista plasma en el informe Operatorio.
conteo_compresas	double precision	Si las compresas están completas o no.
instrumental_completo	double precision	Si el instrumental está completo o no.
cirugia_trabajosa	double precision	Si la cirugía fue trabajosa o no.
causas	varchar	Las causas por las que se determinó realizar la intervención quirúrgica.

TABLA 2.6.27 bqq_interconsulta

Nombre: bqq_interconsulta

Descripción: Contiene los datos relacionados con la consulta a la que fue remitido el paciente después de la operación.		
Atributo	Tipo	Descripción
id_persona (PFK)	integer	Es llave primaria foránea que proviene de la tabla bqq_datos consulta
numero_consulta (PFK)	integer	Es llave primaria foránea que proviene de la tabla bqq_datos consulta
texto_interconsulta	varchar	Guarda la valoración que hace el médico en esta consulta.

TABLA 2.6.28 bqq_interrogatorio

Nombre: bqq_interrogatorio		
Descripción: Es una tabla que se generó de la relación de mucho a mucho entre las tablas bqq_sistema_afeccion y bqq_sistema_afeccion que contiene el tipo de sistema de órganos con su afección correspondiente.		
Atributo	Tipo	Descripción
numero_consulta(PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos consulta
id_persona (PFK)	integer	Es llave primaria foránea que proviene de la tabla bqq_datos consulta
id_afeccion (PFK)	integer	Es llave primaria que proviene de la tabla cfg_afeccion que identifica las afecciones que presentan todos los sistemas de órganos.
id_sistema_de_organos (PFK)	integer	Es llave primaria que proviene de la tabla cfg_sistema_de_organos que identifica a todos los sistemas de órganos existentes.

TABLA 2.6.29 bqq_planificacion_por_cirujano

Nombre: bqq_planificacion_por_cirujano		
Descripción: Contiene los datos de la planificación que tiene cada cirujano para la realización de cirugías.		
Atributo	Tipo	Descripción
id_planificacion_por_cirujano (PK)	integer	Es llave primaria de esta tabla su identificador único.
id_funcionario_cirujano (FK)	integer	Es llave foránea que proviene de la tabla cfg_funcionario_cirujano del Módulo Configuración que identifica al cirujano al que se le realiza la planificación.
id_planificacion_quirurgica (FK)	integer	Es llave foránea que proviene de la tabla bqq_planificacion_quirurgica
cantidad_personal	integer	Guarda la cantidad de personal médico que estará en su operación.

TABLA 2.6.30 bqq_planificacion_quirurgica

Nombre: bqq_planificacion_quirurgica		
Descripción: Esta tabla guarda la planificación de las cirugías que realizada el vicedirector quirúrgico del Hospital.		
Atributo	Tipo	Descripción
id_planificacion_quirurgica (PK)	integer	Es llave primaria de esta tabla su identificador único.
id_salon (FK)	integer	Es llave foránea que proviene de la tabla

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		cfg_salon
id_subservicio	integer	Es llave foránea que proviene de la tabla cfg_subservicio del Módulo Configuración que identifica el salón que se reservará para realizar a operación.
cantidad_cirugias	integer	Guarda el número de cirugías planificadas.
fecha_planificacion	timestamp	Guarda la fecha de la planificación de las cirugías.
tiempo_quirurgico	timestamp	Guarda el tiempo quirúrgico.

TABLA 2.6.31 bqq_posnatales_antecedentes

Nombre: bqq_posnatales_antecedentes		
Descripción: Es una tabla que se generó de la relación de mucho a mucho entre las tablas bqq_antecedentes_no_patologicos y cfg_antecedentes_no_patologico_posnatale.		
Atributo	Tipo	Descripción
id_antecedentes_no_patologicos (PFK)	integer	Es llave primaria que proviene de la tabla bqq_antecedentes_no_patologicos
numero_consulta (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos consulta
id_persona (PFK)	integer	Es llave primaria que proviene de la tabla bqq_datos consulta

TABLA 2.6.32 bqq_premedicacion

Nombre: bqq_premedicacion		
<p>Descripción: Es una tabla que se generó de la relación de mucho a mucho entre las tablas bqq_anestesia_consulta cfg_via y guarda los datos de los medicamentos que se le orientan al paciente en la consulta de anestesia previo a la operación.</p>		
Atributo	Tipo	Descripción
id_anestesia_indicada (PK)	integer	Es llave primaria de esta tabla su identificador único.
numero_consulta(FK)	integer	Es llave foránea que proviene de la tabla bqq_datos_consulta.
id_persona (FK)	integer	Es llave foránea que proviene de la tabla bqq_datos_consulta.
id_via (FK)	integer	Es llave foránea que proviene de la tabla cfg_via que identifica la vía por la que se le suministrará el medicamento al paciente.
id_productos (FK)	integer	Es llave foránea que proviene de la tabla fm_producto del Módulo Configuración que identifica los medicamentos existentes.
dosis	double precision	Si necesita una dosis o no.

TABLA 2.6.33 bqq_registro_temporal_parametros

Nombre: bqq_registro_temporal_parametros

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Descripción: Guarda un registro consecutivo del paciente que se realiza durante los procesos pre-operatorio, trans-operatorio y el post-operatorio.		
Atributo	Tipo	Descripción
id_registro_temporal_parametros	integer	Es llave primaria de la tabla. Su identificador único.
hora_medicamento	timestamp	Guarda la hora en la que se le suministra el medicamento al paciente.
presion_arterial_min	integer	Guarda el valor de la presión arterial mínima que presenta el paciente.
presion_arterial_max	integer	Guarda el valor de la presión arterial máxima que presenta el paciente
frecuencia_cardiaca_min	integer	Guarda el valor de la frecuencia cardiaca mínima que presenta el paciente.
frecuencia_cardiaca_max	integer	Guarda el valor de la frecuencia cardiaca máxima que presenta e paciente.
saturacion_parcial_oxigeno	integer	Guarda el valor de la saturación parcial de oxígeno.

TABLA 2.6.34 bqq_tratamiento_actual

Nombre: bqq_tratamiento_actual		
Descripción: Guarda información referente al tratamiento que el paciente está tomando en el momento de la consulta.		
Atributo	Tipo	Descripción
id_tratamiento_actual (PK)	integer	Es la llave primaria de esta tabla. Su

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

		identificador único.
numero_consulta (FK)	integer	Es llave foránea que proviene de la tabla bqq_datos_consulta.
id_persona (FK)	integer	Es llave foránea que proviene de la tabla bqq_datos_consulta.
id_producto (FK)	integer	Es llave foránea que proviene de la tabla fm_producto del Módulo Farmacia que identifica el medicamento que se le suministra al paciente.

TABLA 2.6.35 bqq_valoracion

Nombre: bqq_valoracion		
Descripción: Guarda la valoración del paciente que realiza el especialista su consulta		
Atributo	Tipo	Descripción
id_persona (PFK)	Integer	Es llave primaria foránea que proviene de la tabla bqq_datos_persona
numero_consulta (PFK)	Integer	Es llave primaria foránea que proviene de la tabla bqq_datos_persona
indicaciones	varchar	Guarda las indicaciones que realiza el especialista al paciente
tratamiento_a_realizar	varchar	Guarda el tratamiento que le indica el medico al paciente.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

En este capítulo se ha obtenido el diseño de la base de datos del Bloque Quirúrgico, tanto los datos como su definición. Todos los accesos a la base de datos se realizan a través del SGBD que mediante su lenguaje de manejo de datos permite a los usuarios la inserción, actualización, eliminación y realizar consultas a la misma en la forma más óptima posible. El diseño de base de datos obtenida posteriormente será analizado y se validará teóricamente.

CAPÍTULO 3: VALIDACIÓN DEL DISEÑO REALIZADO

En este capítulo se establece una valoración del estado en que se encuentra la base de datos del Módulo Bloque Quirúrgico. Donde se expondrán conceptos y reglas a seguir en el diseño de la BD para eliminar repeticiones entre los datos y atributos compuestos para concluir con la realización de un modelo eficiente. De una forma referenciada se explicará como se realiza la seguridad de los datos en el proyecto para garantizar una protección de la información convincente.

3.1 INTEGRIDAD

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados, sobre todo si se trata del Módulo Bloque Quirúrgico donde se maneja información de alta importancia. Esta integridad está expresada mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD en este caso, Postgresql 8.2, quien se debe encargarse de mantenerlas.

Aplicando la integridad de los datos se logra adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada. (*Tutorial de PostgreSQL*)

Las reglas de integridad son aquellas que han de ser aplicadas a una base de datos para asegurar que los datos introducidos sean consistentes con la realidad que pretenden modelar. Existen dos reglas generales que le realizan aportes al modelo relacional.

1. Regla de integridad de entidades

La primera regla de integridad 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.

Atributos Nulos

Cuando en una tupla un atributo es desconocido, se dice que es *nulo*. Un nulo no representa el valor cero ni la cadena vacía, éstos son valores que tienen significado. El nulo implica ausencia de

información, bien porque al insertar la tupla se desconocía el valor del atributo, o bien porque para dicha tupla el atributo no tiene sentido.

Ya que los nulos no son valores, deben tratarse de modo diferente, lo que causa problemas de implementación. De hecho, no todos los SGBD relacionales soportan los nulos.

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

2 Regla de integridad referencial

La segunda regla de integridad 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.

La regla de integridad referencial se enmarca en términos de estados de la base de datos: indica lo que es un estado ilegal, pero no dice cómo puede evitarse. La cuestión es ¿qué hacer si estando en un estado legal, llega una petición para realizar una operación que conduce a un estado ilegal? Existen dos opciones: rechazar la operación, o bien aceptar la operación y realizar operaciones adicionales compensatorias que conduzcan a un estado legal.

Por lo tanto, para cada clave ajena de la base de datos habrá que contestar a tres preguntas:

- Regla de los nulos: ¿Tiene sentido que la clave ajena acepte nulos?
- Regla de borrado: ¿Qué ocurre si se intenta borrar la tupla referenciada por la clave ajena?
- Restringir: no se permite borrar la tupla referenciada.
- Propagar: se borra la tupla referenciada y se propaga el borrado a las tuplas que la referencian mediante la clave ajena.
- Anular: se borra la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).
- Regla de modificación: ¿Qué ocurre si se intenta modificar el valor de la clave primaria de la tupla referenciada por la clave ajena?
- Restringir: no se permite modificar el valor de la clave primaria de la tupla referenciada.
- Propagar: se modifica el valor de la clave primaria de la tupla referenciada y se propaga la modificación a las tuplas que la referencian mediante la clave ajena.
- Anular: se modifica la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos). (*Reglas de integridad*)

La integridad permite obtener los datos almacenados en la base de datos del Módulo Bloque Quirúrgico. Pero también existe la integridad de dominio donde a través de ella se puede controlar la información guardada en la base de datos.

Como dominio, se puede entender, un conjunto de normas de negocio que gestionan la disponibilidad de datos en una determinada columna de una tabla. (4.3 *Integridad de dominio*)

Por ejemplo que sólo se pueda introducir en la tabla `bqg_registro_temporal_parametros`, en el atributo `frecuencia_cardiaca_max` solo los datos validados por un dominio de valores.

A mayor número de limitaciones, mejor se asegura el correcto funcionamiento de nuestra base de datos. Estas normas o reglas de integridad de dominio pueden indicar que campos son necesarios tener obligatoriamente con valores, en la base de datos.

A continuación se describen algunas Reglas de Dominio que utiliza la BD del Módulo Bloque Quirúrgico: Se estableció un dominio, `dom_id`, para lograr que las llaves primarias de las tablas que no son autoincrementables no alcancen valor nulo ni negativo. Asegurando que se cumpla la primera regla de integridad.

El siguiente dominio se relaciona con el atributo `frecuencia_cardiaca_max` que se encuentra en la tabla `bqg_registro_temporal_parametros`.

- `frec_cardiaca_max`, verifica que la frecuencia cardiaca del paciente no alcance valores negativos ni nulos.

Este dominio se relaciona con el atributo `frecuencia_cardiaca_min` que se encuentra en la tabla `bqg_registro_temporal_parametros`.

- `frec_cardiaca_min` verifica que la frecuencia cardiaca del paciente no alcance valores negativos ni nulos.

Este dominio se relaciona con el atributo `tension_arterial_max` que se encuentra en la tabla `bqg_registro_temporal_parametros`

- `tension_arterial_max`, verifica que la tensión del paciente no alcance valores negativos ni nulos.

Este dominio se relaciona con el atributo `tension_arterial_min` que se encuentra en la tabla `bqg_registro_temporal_parametros`.

- `tension_arterial_min`, verifica que la tensión del paciente no alcance valores negativos ni nulos.

Como la base de datos del Módulo está disponible para varios usuarios de forma simultánea, deben existir mecanismos que aseguren que las interrelaciones entre registros se mantienen coherentes, que se respetan las dependencias de existencia y que las claves únicas no se repitan.

Por ejemplo, un usuario no debe poder borrar una entidad de una base de datos, si otro usuario está usando los datos de esa entidad. Este tipo de situaciones son potencialmente peligrosas, ya que

provocan situaciones con frecuencia imprevistas. Ciertos errores de integridad pueden provocar que una base de datos deje de ser usable.

Los problemas de integridad se suelen producir cuando varios usuarios están editando datos de la misma base de datos de forma simultánea. Por ejemplo, un usuario crea un nuevo registro, mientras otro edita uno de los existentes, y un tercero borra otro.

3.2 NORMALIZACIÓN DE LA BASE DE DATOS

El modelo conceptual de datos obtenido mediante la técnica de entidad-relación del Bloque Quirúrgico será refinado y convertido en un modelo lógico relacional, con la utilización de la normalización, lo que ofrecerá como resultado el conjunto de tablas a implementar en la base de datos.

Este proceso tiene como finalidad, reducir las inconsistencias y redundancias de los datos, facilitar el mantenimiento y evitar las anomalías en las manipulaciones de datos que en esta área se efectúan.

El objetivo será obtener un modelo lógico normalizado que represente las entidades normalizadas y las interrelaciones existentes entre ellas.

Para ello, se toma como punto teórico de partida el concepto de dependencia funcional, que dice:

Un atributo B depende funcionalmente de otro atributo A, de la misma entidad si a cada valor de A le corresponde sólo un valor de B. Lo anterior se completa mediante la dependencia funcional total y la dependencia transitiva.

El procedimiento de normalización consiste en someter a las tablas que representan entidades a un análisis formal para ver si cumplen, o no, las restricciones necesarias que aseguren evitar los problemas citados con anterioridad. A mayor nivel de normalización, mayor calidad en la organización de los datos y menor peligro para la integridad de los datos. Este procedimiento consiste en ir alcanzando formas normales. (

2.7. Creación de una base de datos: enfoque E/R y transformación relacional.

)

La teoría de la normalización se ha desarrollado para obtener estructuras de datos eficientes que eviten las anomalías de actualización. El concepto de normalización fue introducido por E.D. Codd y fue pensado para aplicarse a sistemas relacionales. Sin embargo, tiene aplicaciones más amplias.

La normalización es la expresión formal del modo de realizar un buen diseño. Provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información.

Ventajas:

- Evita anomalías en la actualización.

- Mejora la independencia de los datos, permitiendo realizar extensiones de la BD, afectando muy poco, o nada, a los programas de aplicación existentes que accedan la base de datos.

Cuando se está en presencia de una primera relación que contiene enormes problemas de redundancia, consistencia e integridad de los datos, es necesario mejorar estas relaciones. Estas mejoras deben dar como resultado tablas equivalentes y mejores que sus respectivas originales, y poseer siempre tres propiedades: conservación de la información (de atributos y de tuplas), conservación de dependencias y mínima redundancia de los datos. Las mejoras introducidas obligan a plantear hasta que Forma Normal es necesario llegar, es decir, a que nivel de depuración.

La normalización involucra varias fases que se realizan en orden. La realización de la 2da fase supone que se ha concluido la 1ra y así sucesivamente. Tras completar cada fase se dice que la relación está en:

Primera Forma Normal (1FN)

Segunda Forma Normal (2FN)

Tercera Forma Normal (3FN)

Forma Normal de Boyce-Codd (FNBC)

Existen, además, 4FN y 5FN.

Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización. (

2.7. Creación de una base de datos: enfoque E/R y transformación relacional.

)

Pero no siempre es una buena idea tener una base de datos conformada en el nivel más alto de normalización, puede llevar a un nivel de complejidad que pudiera ser evitado si estuviera en un nivel más bajo de normalización.

Primera Forma Normal (1NF)

Una relación está en primera forma normal (1FN) si los valores para cada atributo de la relación son atómicos. Esto quiere decir simplemente que cada atributo sólo puede pertenecer a un dominio, que tiene un valor único para cada fila. Esta prohíbe los atributos multivaluados, compuestos y sus combinaciones. (*Diseño de Base de Datos Relacionales*)

En el primer diseño de la BD del Bloque Quirúrgico se encontraron en ocasiones varios atributos compuestos, al aplicar esta regla se realizaron modificaciones en las tablas para alcanzar este grado de normalización y de esta forma conseguir menor espacio en disco.

Segunda forma normal (2FN):

Una relación está en segunda forma normal si está en la 1ª FN y todos los atributos no clave dependen de la clave completa y no sólo de una parte de esta.

Este paso sólo se aplica a relaciones que tienen claves compuestas, es decir, que están formadas por más de un atributo como llave. Si un esquema de relación no está en 2ªFN, se le puede normalizar a varias relaciones en 2ªFN en las que los atributos que dependen de una parte de la clave formarán una nueva relación que tendrá esa parte de la clave como clave primaria. (*Diseño de Base de Datos Relacionales*)

La segunda forma normal compara todos y cada uno de los campos de la tabla con la clave definida. Si todos los campos dependen directamente de la clave se dice que la tabla está en segunda forma normal. Una vez que ha alcanzado el nivel de la Segunda Forma Normal, se han controlado la mayoría de los problemas de lógica. Se puede insertar un registro sin un exceso de datos en la mayoría de las tablas.

Tercera forma normal (3FN):

Una relación está en tercera forma normal si todos los atributos de la relación dependen funcionalmente sólo de la clave, y no de ningún otro atributo.

Esto significa que en una relación en 3FN, para toda DF: $\rightarrow X \quad Y$, X es una clave.

Aquí se observa que si una relación está en tercera forma normal, está también en segunda forma normal, sin embargo lo inverso no siempre es cierto. (*Diseño de Base de Datos Relacionales*)

La aplicación de estas Formas de Normalización incorpora a la BD del Módulo funcionalidades óptimas que ayudan a obtener correctos resultados cuando se realizan consultas.

Otra ventaja de la normalización es el consumo de espacio. Una base de datos normalizada puede ocupar menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un menor uso de espacio en disco.

Merece la pena destacar que los diseñadores de BD no tienen que normalizar hasta la forma normal más alta posible. Las relaciones pueden dejarse en formas normales inferiores por razones de rendimiento.

Es por ello que el Modelo Entidad Relación del Bloque Quirúrgico se modeló aplicando las reglas de la normalización hasta lograr que se encontrara en 3FN. No se modeló hasta otra forma normal más avanzada porque puede conducir a tener una base de datos ineficiente y construir un esquema demasiado complejo para trabajar. Un balance apropiado de sentido común y práctico puede ayudar a decidir hasta que nivel se normalizará.

3.3 ANÁLISIS DE LA REDUNDANCIA DE LA INFORMACIÓN

Para que una base de datos sea efectiva hace falta eliminar en la medida de lo posible las redundancias, es decir, las repeticiones de los datos que puedan llevar a error, como se da el caso al llamar a un mismo campo de distinta manera en varios archivos, ya que existe el riesgo de inconsistencia entre las distintas versiones de los mismos datos. Además requiere un mayor espacio de almacenamiento, que influye en mayor coste y mayor tiempo de acceso a los datos.

La inconsistencia es un estado en el que dos o más datos repetidos en una BD contienen diferentes valores. Generalmente es consecuencia de la actualización dispareja o desordenada de datos redundantes. (*Otras ventajas de las bases de datos*)

Se analizó minuciosamente este aspecto en la base de datos y se trató de eliminar todos los casos donde existían tablas donde los valores de los atributos se repetían en otras entidades. Ver figura 1 y 2.

bqg_anestesia_trans_operatoria
fecha_hora_llegada (PK)
id_informe_anestesia (PK)
id_funcionario_enfermera
id_funcionario_cirujano
id_funcionario_tecnico
id_funcionario_anestesista
numero_salon
turno_quirurgico
hora_comienzo_anestesia
complejidad_induccion
hora_inicio_cirugia
hora_fin_cirugia
hora_salida
id_intubacion
id_complicaciones_anestesia
frecuencia_cardiaca

Figura 1.

bqg_registro_temporal_parametros
id_registro_temporal_parametros (PK)
id_informe_anestesia
hora_medicationto
presion_arterial_min
presion_arterial_max
frecuencia_cardiaca_min
frecuencia_cardiaca_max
saturacion_parcial_oxigeno
id_tipo_registro_temporal

Figura 2.

En la tabla bqg_anestesia_trans_operatoria y bqg_registro_temporal_parametros existían valores repetidos en el atributo frecuencia_cardiaca, frecuencia_cardiaca_min y frecuencia_cardiaca_max respectivamente, que mostraban el mismo resultado. La descripción de este atributo muestra la frecuencia cardiaca que presenta un paciente en un determinado estado. Al excluirse esta redundancia se eliminó el atributo de la tabla anestesia_tran_operatoria y se mantuvieron los de bqg_registro_temporal_parametros, ya que se guardaba la misma información en dos lados diferentes de la base de datos, lo que dificultaba la tarea de modificación.

La duplicación de estos datos genera a su vez una duplicación del trabajo a la hora de mantenerlos y actualizarlos. Por tanto las Bases de Datos al reducir la duplicación de datos, disminuyen el trabajo.

3.4 ANÁLISIS DE LA SEGURIDAD DE LOS DATOS

Unas de las características importantes que debe cumplir una base de datos que esté instalada en un Hospital es evitar que frente a fallos de hardware existan fugas de datos y que existan accesos no autorizados que puedan romper la integridad de los datos, porque en este sector, se maneja mucha información sensible a cambios, y se convierte de gran importancia asegurarlos.

Además, es imprescindible garantizar que los datos sólo serán presentados a quien esté autorizado, ya que sino se puede comprometer el sistema al permitir la entrada a personal no autorizado en la entidad. En gran parte, los sistemas de bases de datos ofrecen múltiples características que permiten asegurarlas.

El Gestor de Base de Datos del Bloque Quirúrgico se encarga de realizar las restricciones en la seguridad, donde cada usuario tiene acceso a un conjunto de acciones que fueron definidas por el administrador del sistema, garantizando la confiabilidad de los datos.

Se crean diferentes usuarios de la base de datos para cada aspecto de su aplicación con derechos muy limitados sobre los objetos de la base de datos. Tan solo deben otorgarse los privilegios estrictamente necesarios, y evitar que el mismo usuario pueda interactuar con la base de datos en diferentes casos. Esto quiere decir que si un intruso gana acceso a la base de datos usando las credenciales de sus aplicaciones, él solo puede efectuar tantos cambios como su aplicación se lo permita.

La seguridad de las BD del Proyecto GEHOS se realizará desde el Módulo de Seguridad que implementará esta protección en la capa de acceso a datos, para no depender solamente de la brindada por el gestor de bases de datos. Este Módulo de forma centralizada se encargará de la seguridad del sistema completo.

3.5 TRAZABILIDAD DE LAS ACCIONES

El chequeo de las acciones realizadas por cada usuario en la base de datos tales como: insertar, eliminar, actualizar y seleccionar determinado archivo se ha convertido en un aspecto gran importancia para los administradores de sistemas tener registrado este control.

Este proceso se conoce como trazabilidad de las acciones el cual representa un conjunto de medidas, acciones y procedimientos que permiten registrar e identificar cada producto desde su origen hasta su destino final. Consiste en la capacidad para reconstruir la historia, recorrido o aplicación de un determinado producto, identificando diversos aspectos como son:

- El origen de sus componentes
- Historia de los procesos aplicados al producto
- Distribución y localización después de su entrega

La utilización de este proceso en el proyecto GEHOS se efectúa mediante el Módulo de Seguridad donde tienen implementado un sistema que garantiza el control de las acciones que realiza determinado usuario en tiempo estipulado. **Ver Anexo 2.**

En el Anexo 2 se representa el diseño de la base de datos del Módulo Seguridad, que registra un control de todos los movimientos efectuados en la base de datos de cada Módulo. El rol que se otorgue a cada personal médico tendrá los permisos necesarios para realizar las modificaciones en la base de datos que se le han permitido. Es decir cada usuario del sistema tendrá un rol predefinido por el administrador, y este es el encargado de conceder los permisos a cada personal que interactúe con la aplicación.

Este capítulo concluyó con la validación teórica del diseño de la base de datos del Módulo Bloque Quirúrgico, se trató de eliminar las redundancias existentes, así como las reglas de integridad para validar los datos que se entran al sistema. Se explica como se realiza la seguridad de los datos que será implementado por el Módulo de Seguridad del Proyecto. Todo lo anterior permitió diseñar una base de datos normalizada hasta alcanzar la tercera forma normal.

CONCLUSIONES

Con el desarrollo de este trabajo se arriba a las siguientes conclusiones:

- Se efectuó un estudio de los procesos vinculados con las intervenciones quirúrgicas en los hospitales cubanos y de los sistemas similares existentes, nacionales e internacionales.
- Se obtuvo como artefacto de la Metodología RUP, el Diagrama de Clases Persistentes elaborado a partir del Diagrama de Clases del Diseño.
- Se realizó una validación teórica del diseño para garantizar la integridad de los datos, eliminar redundancia en la información y certificar que la base de datos estuviera correctamente normalizada.

Se realizó un diseño de la base de datos que interrelaciona los procesos que en su conjunto son vitales para el buen funcionamiento del Bloque Quirúrgico y que deben contribuir, en parte al éxito de cualquier intervención quirúrgica que se realice.

Con lo antes planteado se da cumplimiento a los objetivos trazados para la realización del trabajo de diploma obteniendo el diseño de la base de datos que mejora la gestión de la información referente a los procesos vinculados con las intervenciones quirúrgicas que tienen lugar en los hospitales del país.

RECOMENDACIONES

- Se recomienda realizar un estudio del sistema para lograr mejoras en la realización de versiones futuras de la base de datos del Módulo.
- Incorporar funcionalidades y servicios al diseño para un mayor perfeccionamiento del mismo.

REFERENCIA BIBLIOGRÁFICA

2.7. *Creación de una base de datos: enfoque E/R y transformación relacional.*

. Disponible en: <http://www.tramullas.com/documatica/2-7.html>

4.3 *Integridad de dominio.* Disponible en:

<http://www.adrformacion.com/cursos/sqlserver/leccion2/tutorial3.html>

Arquitectura de Sistemas de Base de Datos. Disponible en: http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T9900_OGonzalez.pdf

Bienvenido al Hospital. Disponible en: <http://ciberhabitat.gob.mx/hospital/>

Características de PostgreSQL. Disponible en: <http://www.usabilidadweb.com.ar/postgre.php>

Diseño de Base de Datos Relacionales. 15/05/07]. Disponible en: <http://usuarios.lycos.es/cursosgbd/UD4.htm>

Entonación de Bases de Datos. Disponible en: <http://www.bd.cesma.usb.ve/ci5851/apuntes3.pdf>

Gestión de Datos. Disponible en: <http://www.jorgesanchez.net/bd/docs2006/sgbd.pdf>

Información sobre Care2x. Disponible en: <http://care2x.ourproject.org/>

Introducción. Disponible en: <http://www.cecam.sld.cu/pages/docencia/introduccion.htm>

LEDO, L. M. V. *INFORMACIÓN, TECNOLOGÍAS Y ÉTICA EN LA SALUD.* Disponible en: http://www.cecam.sld.cu/pages/rcim/revista_9/articulos_htm/eticaensalud.htm

Opera, Sistema de Gestión de la Cirugía. Disponible en: http://www.chca.ca/opera/index_spa.html

Oracle. Disponible en: <http://es.wikipedia.org/wiki/Oracle>

Otras ventajas de las bases de datos. 13/03/07]. Disponible en: http://html.rincondelvago.com/base-de-datos_3.html

Postgres vs MySQL. Disponible en: http://www.netpecos.org/docs/mysql_postgres/index.html

Producto

Disponible en: http://www.sivsa.com/site/espanha/produto_hosix.asp

¿Que es Postgres?. Disponible en: <http://www.mmug.cl/articulos.php?id=293&tod=1>

Reglas de integridad. Disponible en: http://www.aulaclie.es/sql/b_8_1_1.htm

Sistema de gestión de base de datos. Disponible en:

http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos

Sistema de Información Hospitalaria. Disponible en:

<http://educacion.salud.gob.mx/cursos/informatica/HIS/his.pdf>

Sistema de Información Integral Hospitalario áreas de negocio 2004 [Disponible en:

<http://cim.uag.mx/his.html#nis>

TESIS. Quirófano. Disponible en: http://www.sisinf.com/quiroyfans_es.htm

Tutorial de PostgreSQL. Disponible en: <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html>

VANGUARD Optim OTM Disponible en: <http://www.vanguard.de/dynasite.cfm?dssid=4669>

BIBLIOGRAFÍA

Arquitectura de Sistemas de Base de Datos. Disponible en: http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T9900_OGonzalez.pdf

Características de PostgreSQL. Disponible en: <http://www.usabilidadweb.com.ar/postgre.php>

Conceptos básicos de integridad referencial. Disponible en: http://www.aulaclie.es/sql/b_8_1_1.htm

5.2 Esquema conceptual de la base de datos. Disponible en: <http://elies.rediris.es/elies9/5-2.htm>

Gestión de Datos. Disponible en: <http://www.jorgesanchez.net/bd/docs2006/sgbd.pdf>

PostgreSQL vs MySQL. Disponible en: http://www.netpecos.org/docs/mysql_Postgresql/index.html

Sistema de gestión de base de datos. Disponible en:

http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos

Sistemas de información hospitalaria. Disponible en:

<http://www.virtual.epm.br/material/healthcare/spanish/E0403.pdf>

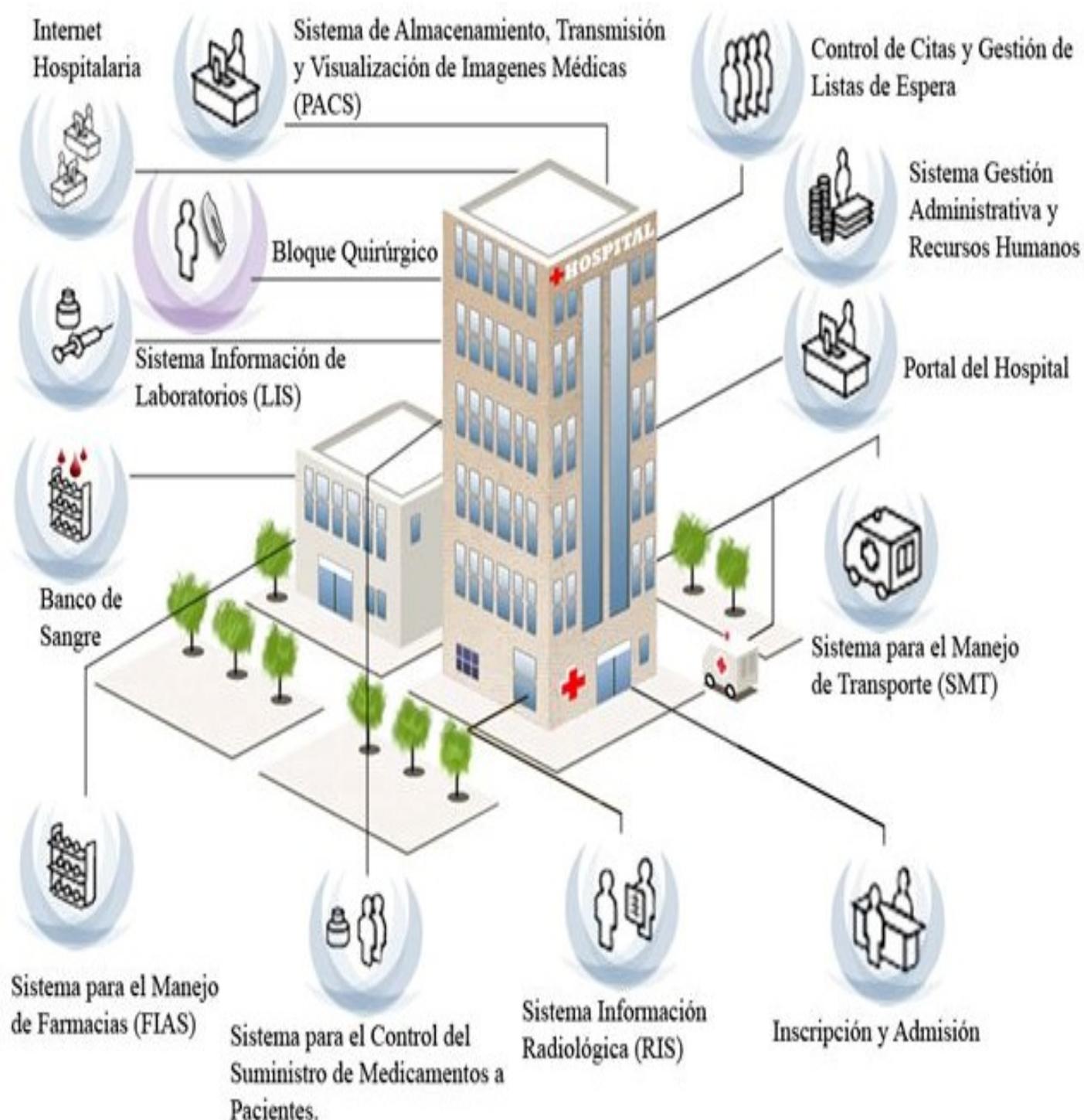
GARCIA, L. R. M. M. DISEÑO de BASES DE DATOS. OCTUBRE 1999.

ROBERTO ANDRADE FONSECA. ABL CONSULTORES, S. A. D. C. V. *Programación de funciones en PL/pgSQL para PostgreSQL*, 8 de febrero de 2002. [Disponible en:

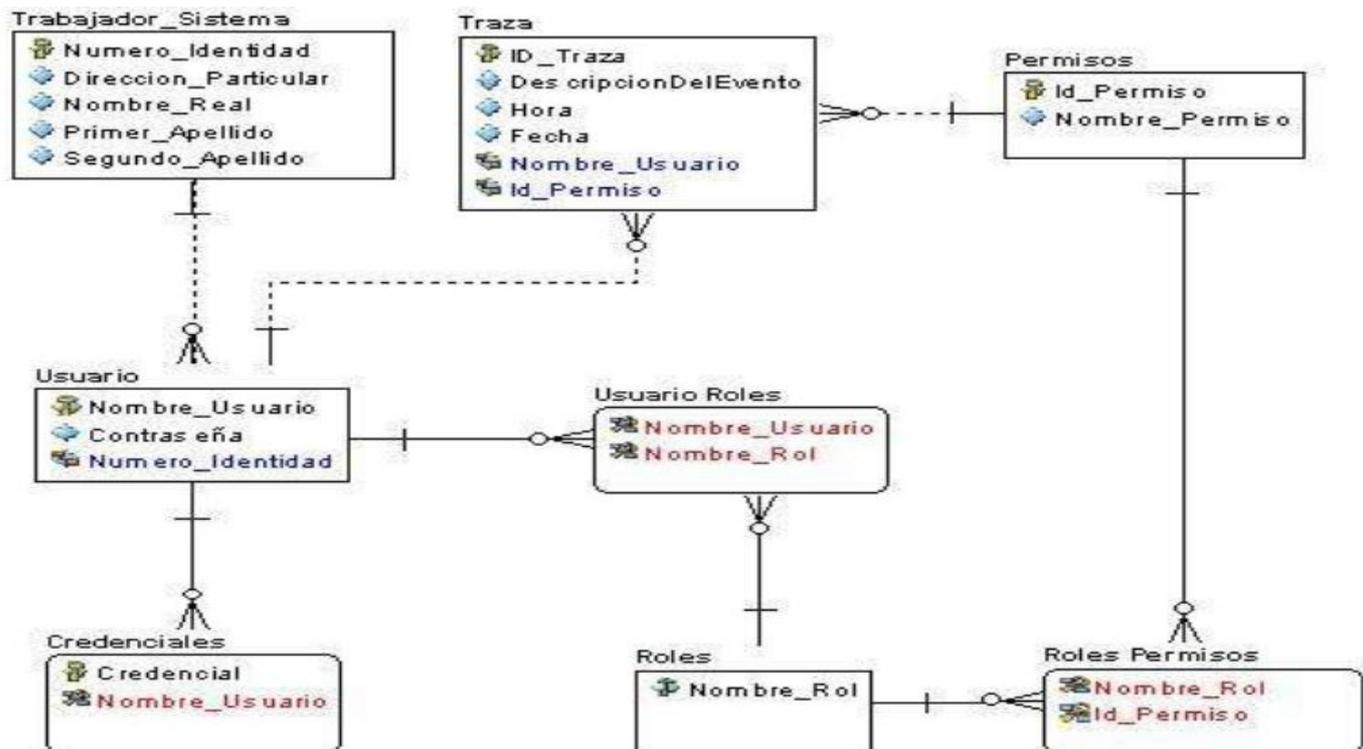
http://foobar.cl/~chepito/software/PL_pgSQL.pdf

ANEXOS

1. Infraestructura de un SIH.



2. Diseño de la base de datos del Módulo Seguridad.



GLOSARIO

Cirugía: Especialidad médica cuyo fin es curar las enfermedades o malformaciones mediante intervenciones quirúrgicas.

Datos: Técnicamente, los datos son hechos y cifras en bruto, tales como órdenes y pagos, los cuales se procesan para obtener información, por ejemplo el saldo deudor y el monto disponible. Sin embargo, en el uso común, los términos datos e información se toman como sinónimos. Cualquier forma de información, ya sea en forma electrónica o sobre papel. En forma electrónica, "datos" se refiere a archivos, bases de datos, documentos de texto, imágenes y, voz y video codificados en forma digital.

Información: Es la comunicación o adquisición de conocimientos que permiten ampliar o precisar los que se poseen sobre una materia determinada.

Salud: es definida por la Constitución de 1996 de la Organización Mundial de la Salud como el estado de completo bienestar físico, mental y social, y no solamente la ausencia de afecciones o enfermedades. También puede definirse como el nivel de eficacia funcional y/o metabólica de un organismo tanto a nivel micro (celular) como en el macro (social).

Sistema Operativo (OS) Programa especial el cual se carga en una computadora tras ser encendida y cuya función es gestionar los demás programas, o aplicaciones, que se ejecutan en dicha computadora.

Tupla: Una tupla es una lista inmutable. Una tupla no puede modificarse de ningún modo después de su creación.