

Universidad de las Ciencias Informáticas

Facultad 6



Título: “Sistema Informático de Genética Médica: Diseño de la Base de Datos.”

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

Autores:

Carlos Luis Hernández Hernández
Liandy Izquierdo Padrón

Tutores:

Ing. Alfonso Claro Arceo
Ing. Yusdenis Sánchez Perodín

Ciudad de la Habana. Cuba

Junio del 2008

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del 2008.

Firma de Liandy Izquierdo (Autor)

Firma de Carlos L. Hernández (Autor)

Firma de Alfonso Claro (Tutor)

Firma de Yusdenis Sánchez (Tutor)

“La ciencia es la expresión de una necesidad inherente al ser humano y, en todo caso, está ligada a la función superior de su naturaleza inteligente: la capacidad de crear.”

René Gerónimo Favaloro

Agradecimientos

De Liandy:

Especialmente a mi amigo Marcel ¿Qué nos hubiéramos hecho sin usted?

A mi novia Sisely por haber contribuido al desarrollo de esta tesis y por la paciencia y comprensión que me ha brindado en estos años.

A mis amigos: Alejandro, Daumel, Angel, Yoandrys y el Tony por haberme apoyado y ayudado en todas las asignaturas, y por lograr que estos 5 años hayan sido lo más divertido posible.

A mis padres por la preocupación, apoyo y dedicación que me han brindado.

A Franklís por apoyarnos durante el desarrollo de la tesis.

A Martha Nieves por la preocupación que tuvo durante el desarrollo de la tesis.

A nuestros tutores por el apoyo científico que nos dieron.

A todos los profes que tuve durante los 5 años por los conocimientos que me brindaron.

A todos los que de alguna forma u otra tuvieron que ver con el desarrollo de la tesis.

A todos mis compañeros tanto del grupo como los que no son del grupo, por haber compartido estos cinco años.

De Carlos:

Quiero agradecer a todas aquellas personas que con su apoyo incondicional han hecho posible la realización exitosa de este trabajo:

A mis padres Elia y Luis por su dedicación y amor incondicional.

Agradecimientos

A mis tíos: Nieve, Aleida, Manuel y Eduardo por ser mis guías y por su preocupación y ayuda desinteresada, ellos a parte de ser mis tíos yo los considero mis padres.

A mi abuela Ofelia por estar siempre pendiente de mis estudios y de mis resultados. Por ayudarme a enfrentar momentos difíciles y por siempre darme ese amor y ese cariño desinteresado que la caracteriza.

A mi novia por aguantarme durante tres años, en los momentos buenos y malos y por siempre estar a mi lado brindándome amor y cariño.

A mi hermana Elianys y a mis primos Luvia y Carlos Eduardo por brindarme alegría.

A mi familia y mis vecinos por su preocupación y estar siempre pendiente de mis estudios.

A todos mis compañeros del grupo por haber compartido conmigo durante cinco años.

A mi amigo Marcel por su gran ayuda en la elaboración de este trabajo, de verdad es un gran amigo.

A mis tutores por su aporte científico en este trabajo.

A las profes Martha Nieves y Aislein Blanco por su ayuda desinteresada y por su dedicación.

A todos los profes que tuve durante los cinco años de la carrera, de verdad fueron maravillosos.

Los que de manera involuntaria he dejado de citar mis disculpas y mis más sinceros agradecimientos.

A todos muchísimas gracias.

Dedicatoria

“A mis padres Nancy y Pablo por ser mis mayores guías en esta tarea.”

“A mis abuelos queridos Leonor y Tebelio por el cariño que me brindan en todo momento.”

Liandy:

“A mis padres por tomarme en su corazón y llevarme de su mano.”

“A mis tíos: Nieve, Aleida, Eduardo y Manuel por brindarme siempre su apoyo y su cariño.”

“A mi abuela Ofelia por quererme y apoyarme en todo momento.”

“A mi novia por brindarme amor y estar siempre a mi lado.”

“Especialmente le dedico esta tesis a mi abuelo Calixto Hernández Corrales, que aunque no esta presente físicamente, siempre estará en mi pensamiento y en mi corazón. El ha sido mi guía y mi ejemplo a seguir como hombre y como persona, además porque él ha sido mi padre, mi hermano, mi tío, mi maestro, lo ha sido todo y a él le debo todo lo que soy hoy.”

Carlos.

Resumen

El presente trabajo describe la propuesta del modelo lógico y físico de la base de datos correspondiente al Sistema Informático de Genética Médica (SIGM). Para hacer posible los objetivos de este trabajo, previamente se realizó un estudio de las herramientas más utilizadas para diseñar bases de datos y los principales gestores de bases de datos que existen. Basado en el estudio anterior y en la decisión de la dirección del proyecto seleccionamos las herramientas que más se ajustaban a las características del trabajo que se iba a desarrollar. Luego de realizar la elección se describe la solución propuesta y finalmente se hace una valoración del trabajo realizado. Seguidamente se enuncian las conclusiones que se obtuvieron en este trabajo así como las recomendaciones del mismo.

Índice

INTRODUCCIÓN	1
Capítulo 1: Fundamentación teórica.....	4
1.1. La genética médica.	4
1.2. Metodologías de desarrollo.....	5
1.2.1. Rational Unified Process (RUP).	5
1.3. Surgimiento y desarrollo de las bases de datos.	7
1.4. Tipos de bases de datos.	9
1.5. Modelos de bases de datos.	10
1.6. Sistemas gestores de bases de datos.	12
1.6.1. Oracle.	14
1.6.2. PostgreSQL.	15
1.6.3. Microsoft SQL Server.....	16
1.6.4. MySQL.....	17
1.7. Herramientas CASE.....	18
1.7.1. ER/Studio.....	19
1.7.2. DBDesigner.	20
1.7.3. Rational Rose.....	21
1.7.4. Visual Paradigm.....	21
1.8. Herramientas de Desarrollo.	22
1.8.1. EMS Data Generator for MySQL.....	22
Conclusiones.....	22
Capítulo 2.....	24
2.1. Requisitos no funcionales.	24
2.2. Estrategia de integración de la solución con otros módulos y sistemas.....	25
2.3. Descripción de la arquitectura y fundamentación.....	30
2.4. Modelo de datos.	31
2.5. Descripción de las tablas.....	38

Conclusiones	87
Capítulo 3: Validación del diseño realizado.....	88
3.1. Validación teórica del diseño.....	88
3.1.1. Integridad.....	88
3.1.2. Normalización de la base de datos.....	89
3.1.3. Análisis de la redundancia de la información.....	92
3.1.4. Seguridad de la base de datos del SIGM.....	92
3.2. Validación funcional.....	93
3.2.1. Llenado automático de los datos y pruebas de carga a la base datos.....	93
Conclusiones	95
CONCLUSIONES	96
RECOMENDACIONES.....	97
REFERENCIAS BIBLIOGRÁFICAS	98
BIBLIOGRAFÍA	100
ANEXOS	102
GLOSARIO DE TÉRMINOS	109

INTRODUCCIÓN

La genética es una rama de la medicina que se encarga de estudiar el papel que desempeñan los factores hereditarios y genéticos respecto de una enfermedad, un defecto congénito o la susceptibilidad heredada para ciertos problemas de salud [1].

Con el triunfo de la Revolución Cubana y el esfuerzo de la dirección del país por mejorar la calidad de la vida y elevar el bienestar del pueblo cubano se han producido en las últimas décadas grandes avances en la genética y la tecnología biomédicas, que abren nuevos y vastos espacios a la investigación y brindan novedosas herramientas en esta rama.

Como resultado de estos avances surge el Centro Nacional de Genética Médica (CNGM) como una propuesta del Departamento de Genética Médica "Victoria de Girón", fundado en 1970. Su creación fue aprobada en julio de 1981 con la introducción del programa para el diagnóstico y prevención de enfermedades genéticas, dentro del Programa Materno-Infantil del Ministerio de Salud Pública (MINSAP). Esta institución se ha expandido por todo el país a través de la Red Nacional de Genética Médica, la cual está compuesta por 184 centros ubicados en todos los municipios y provincias del país, que coordinados por el CNGM, conducen al programa nacional para el diagnóstico, manejo y prevención de enfermedades genéticas y defectos congénitos. Esta red de instituciones desarrolla acciones asistenciales, docentes y de investigación en el campo de los problemas de salud de carácter genético en la población cubana.

El objetivo fundamental del CNGM es desarrollar proyectos de investigación e innovación, en el campo de la genética médica, inmunología y disciplinas afines. Evaluar e introducir nuevas tecnologías para el diagnóstico, tratamiento y asesoramiento en relación con las enfermedades genéticas. Algunos de los proyectos ramales de salud del CNGM aprobados por la dirección de ciencia y técnica del MINSAP son:

Comportamiento clínico epidemiológico de los defectos congénitos en Cuba.

Prevalencia al nacimiento de las hemoglobinopatías en Cuba.

Al calor de la batalla de ideas surgió la Universidad de las Ciencias Informáticas (UCI), centro que tiene como objetivo informatizar el país y exportar software para el desarrollo económico del mismo. La UCI dando cumplimiento a los objetivos planteados anteriormente, en coordinación con el CNGM, desarrollaron siete proyectos llamados:

Introducción

- Sistema Informatizado para la Gestión del Registro Cubano de Gemelos (RECUGEM).
- Sistema Informatizado para la Gestión del Registro Cubano de Retraso Mental (RECIUREM).
- Sistema Informatizado para la Gestión de Teleconsultas entre Genetistas (Teleconsulta).
- Sistema Informatizado para la Gestión del Registro Cubano de Malformaciones Congénitas (RECUMAC).
- Sistema Informatizado para la Gestión del Registro Cubano de Historias Clínicas (RECIUHCLIN).
- Sistema Informatizado para la Gestión del Registro Cubano de Discapacitados (RECIUDIS).
- Sistema Informatizado para la Gestión del Registro Cubano de las Enfermedades Genéticas (RECIUEGEN).

Posteriormente el CNGM y la UCI decidieron que estos proyectos se unirían para crear lo que hoy es el SIGM, pasando cada proyecto como módulos del mismo. Como eran proyectos individuales, y además de que no estaban integrados a la red nacional de salud, cada uno tenía una base de datos encargada de almacenar la información, trayendo consigo un gran cúmulo de información duplicada innecesariamente, surgiendo como **problema científico de la investigación**: ¿Cómo almacenar de forma única los datos del Sistema Informático de Genética Médica?

Luego se define que el **objeto de estudio** es el proceso de diseño de bases de datos.

El **campo de acción** de la investigación se enmarca en el diseño de la base de datos para el estudio genético poblacional.

Para dar solución a este problema se define como **objetivo general**: diseñar una base de datos que almacene la información de los módulos del SIGM.

De este objetivo se derivan los siguientes **objetivos específicos**:

Diseñar la base de datos.

Realizar pruebas a la base de datos.

Para dar cumplimiento a estos objetivos se trazaron las siguientes **tareas**:

Entrevistas a líderes e integrantes de los diferentes módulos del SIGM.

Revisión de las bases de datos de los módulos del SIGM.

Investigación y selección de las tendencias y tecnologías para el desarrollo de bases de datos.

Introducción

Análisis de la integridad de los datos en la base de datos.

Estudio de la normalización de la base de datos.

Análisis de la redundancia de la información en la base de datos.

Análisis de la seguridad de la base de datos.

Revisión y selección de herramientas para pruebas de carga intensiva.

Pruebas a la base de datos.

Capítulo 1: En este capítulo se brinda información sobre la genética médica, así como una panorámica del surgimiento y desarrollo de las bases de datos, tipos y modelos de bases de datos, gestores de bases de datos, una breve reseña de algunas herramientas CASE, herramientas de desarrollo y de algunas metodologías de desarrollo de software.

Capítulo 2: En este capítulo se describe la solución propuesta, concentrándose en la estrategia de integración de la solución con otros módulos, la descripción de la arquitectura, y en la descripción de las principales tablas de cada módulo del SIGM.

Capítulo 3: En este capítulo se ofrece información sobre la validación del diseño descrito en el capítulo 2, detallando las reglas a tener en cuenta para un buen diseño, además de la descripción de una validación funcional a través de la generación de códigos para un llenado voluminoso e inteligente de la base de datos.

Capítulo 1: Fundamentación teórica.

En este capítulo se brinda información sobre la genética médica, así como una panorámica del surgimiento y desarrollo de las bases de datos, tipos y modelos de base datos, gestores de bases de datos, una breve reseña de algunas herramientas CASE, herramientas de desarrollo y de algunas metodologías de desarrollo de software.

1.1. La genética médica.

Durante la última década del siglo XX, los avances de la ingeniería genética y las nuevas tecnologías de la información, condicionaron el surgimiento de una disciplina que creó vínculos indisolubles entre la informática y las ciencias biológicas: la Bioinformática. Las especialidades médicas que han recibido una mayor influencia de la Bioinformática son la Genética Médica, la Bioquímica Clínica, la Farmacología, las Neurociencias, la Estadística Médica, la Inmunología, la Fisiología y la Oncología. Entre los principales factores que han favorecido el desarrollo de la Bioinformática, se encuentran el impresionante volumen de datos sobre secuencias generadas por los distintos proyectos genoma, tanto el humano como el de otros organismos, y la necesidad de almacenar esta información en bases de datos.

Hace unos años, los resultados de los experimentos podían interpretarse sobre el cuaderno de laboratorio, hoy se necesitan bases de datos y técnicas de visualización sólo para almacenarlos y comenzar a estudiarlos.

Un ejemplo de bases de datos aplicadas a la Bioinformática es la base de datos Medline, que desde el año 1960 hasta el 29 de enero del 2003, fecha en que se cerró la recogida de datos, procesó un total de 281 572 registros que abordan desde diferentes puntos de vista el campo de la Informática Médica (IM). Este estudio recogió datos de casi todas las especialidades médicas, y arrojó que la Genética Médica constituye la especialidad que ha recibido mayor influencia de la Bioinformática.

Hoy en día un grupo de científicos está trabajando en una base de datos sin precedentes conocida como "Proyecto de los 1000 Genomas", donde se analiza el material genético de mil individuos anónimos en todo el mundo. El objetivo de esta inmensa base de datos, que estará disponible para la comunidad

Capítulo 1

científica de manera gratuita a través de Internet, es lograr una fotografía lo más detallada posible de la variación genética del ser humano.

Otro de los proyectos que se están desarrollando en el campo de la genética médica es el proyecto Genoma Humano, que comenzó en 1990, es un proyecto para secuenciar los 3 mil millones de pares de bases del genoma humano y para realizar un mapa de todos los genes humanos (en total 30.000 a 35.000) en los 46 cromosomas, y almacenar esta información en bases de datos para realizar un estudio y comprender cómo se desarrolla el cáncer en primer lugar y por qué algunas personas con cáncer responden bien al tratamiento y nunca tienen una recurrencia del cáncer, mientras que otras desarrollan cáncer nuevamente, a pesar del tratamiento.

En Cuba, los especialistas han realizado algunos estudios en el campo de la Genética Médica, la información de los mismos se ha almacenado manualmente ya que no existe un sistema automatizado capaz de hacer esta función.

1.2. Metodologías de desarrollo.

Una metodología de desarrollo de software permite definir quién debe hacer qué, cuándo y cómo debe hacerlo. Una metodología es un proceso que puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales.

1.2.1. Rational Unified Process (RUP).

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

Inicio: el objetivo en esta etapa es determinar la visión del proyecto.

Elaboración: en esta etapa el objetivo es determinar la arquitectura óptima.

Construcción: en esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

Transmisión: el objetivo es llegar a obtener el release del proyecto.

Capítulo 1

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

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:
Disciplina de Desarrollo:

Modelo de Negocios: entendiendo las necesidades del negocio.

Requerimientos: trasladando las necesidades del negocio a un sistema automatizado.

Análisis y Diseño: trasladando los requerimientos dentro de la arquitectura de software.

Implementación: creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.

Pruebas: asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado esta presente.

Disciplina de Soporte:

Configuración y administración del cambio: guardando todas las versiones del proyecto.

Administrando el proyecto: administrando horarios y recursos.

Ambiente: administrando el ambiente de desarrollo.

Distribución: hacer todo lo necesario para la salida del proyecto.

El ciclo de vida de RUP se caracteriza por:

Dirigido por casos de uso: los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo

Capítulo 1

que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

Como RUP es un proceso, en su modelación define como sus principales elementos:

Trabajadores: vienen hacer las personas o entes involucrados en cada proceso.

Artefactos: un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Actividades: son los procesos que se llegan a determinar en cada iteración.

Flujo de actividades: secuencia de actividades realizadas por los trabajadores y que producen un resultado de valor observable.

Se asume la metodología RUP, ya que hace exigente el uso de artefactos obteniendo para el rol diseñador de base de datos como principal artefacto: el modelo de datos. Es una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software, y además es la que propone la arquitectura del proyecto, ya que el mismo estará integrado a la red de salud.

1.3. Surgimiento y desarrollo de las bases de datos.

Una base de datos es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquinas accesibles en tiempo real y compatible con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo [2].

El concepto de base de datos es moderno, pero tiene sus antecedentes desde los primeros intentos humanos por archivar y retener la información, ya que ésta se mantenía muy fugazmente en la memoria de los hombres. El primer paso hacia las bases de datos modernas lo constituye la representación de la información en forma escrita. Hasta muy recientemente, las representaciones de la información consistían

Capítulo 1

en marcas visibles hechas a objetos físicos, tales como pictogramas, jeroglíficos, alfabetos, libros, impresiones, fotocopias y otras formas.

Los primeros sistemas computacionales, llamados sistemas de procesamiento de datos, ejecutaban las funciones habituales de tratamiento de los registros de una organización, tales como las funciones de contabilidad, nóminas y otras, de manera similar a como se realizaban manualmente. De este modo, en los años sesenta los archivos o ficheros en la computadora se correspondían con los archivos de papel y el acceso a los registros de los ficheros se realizaba secuencialmente. Sin embargo, estos sistemas son insuficientes para ejecutar muchas de las tareas que resultan rutinarias en los negocios.

Estos problemas se resolvieron parcialmente con los sistemas orientados a ficheros, con la introducción de los ficheros de acceso directo, en particular, los ficheros secuenciales indizados.

Pronto se hizo obvio que estos sistemas tenían un conjunto de limitaciones que restringían las posibilidades de los sistemas computacionales de la época. Entre tales limitaciones se distinguen:

Gran redundancia de los datos: como los ficheros y los programas de aplicación se creaban por diferentes programadores a largo de un período de tiempo, los ficheros presentaban frecuentemente formatos diferentes y los programas se escribían en diversos lenguajes de programación. Por lo que muchas aplicaciones utilizaban sus propios ficheros de datos, de modo que los elementos comunes a varias aplicaciones se repetían en diferentes ficheros. Esta redundancia, que se agravaba por el hecho de que en numerosas ocasiones se utilizaban nombres o tipos de datos diferentes para los campos en que se representaban los mismos elementos de datos, generaba falta de correspondencia entre las diversas versiones de los datos comunes.

Pobre control sobre los datos: como no había control centralizado de los elementos de los datos, la existencia de ficheros aumenta la posibilidad de la utilización indiscriminada de un mismo término con diferentes significados en diferentes contextos (nombre) y términos que significan lo mismo (identificador, código), lo que puede provocar falta de claridad con relación a la manipulación de los datos.

Capacidades inadecuadas de manipulación de los datos: los sistemas de ficheros no proporcionan conexiones sólidas entre los datos contenidos en ficheros diferentes, sino que se caracterizan por representar datos aislados. Debido a esto resultaban ineficientes las respuestas a solicitudes que

requerían el tratamiento de conjuntos de registros que representaban computacionalmente al fenómeno en cuestión y que de hecho se encontraban interrelacionados en la realidad.

Entre otras insuficiencias de los sistemas de gestión de ficheros también se encuentran las anomalías que pueden ocurrir ante el acceso concurrente (simultáneo) de múltiples usuarios a los datos y las dificultades para garantizar restricciones de seguridad (autorización de acceso a los datos). Resulta casi imposible determinar y supervisar los programas de aplicación que se añaden al sistema por lo que se dificulta el control del acceso a los datos garantizando la consistencia de la base de datos.

Los sistemas de bases de datos superaron las limitaciones de los sistemas orientados a los ficheros al desarrollarse sobre una estructura de datos centralizada e integrada, de modo que todas las aplicaciones comparten un fondo común de datos, minimizando con ello los problemas de redundancia y de falta de control sobre los datos. Asimismo, los sistemas de bases de datos facilitaron la interrelación estrecha entre los datos de ficheros diferentes, lo que contribuyó a aumentar las capacidades de manipulación de los datos así como incrementó las posibilidades de automatización con respecto a la verificación de la consistencia y la integridad de los datos. Por consiguiente, el desarrollo de los sistemas de bases de datos se convirtió en crucial para proporcionar información correcta y oportuna.

Los sistemas de procesamiento de datos, en sus inicios, ejecutaron las tareas administrativas para reducir la manipulación de documentos en papel. Posteriormente, se han utilizado dichos sistemas para la producción y la gestión de la información. De tal modo, en la actualidad, los sistemas de bases de datos se han convertido en el fundamento que garantiza el funcionamiento correcto de los sistemas de información, especialmente para la gestión corporativa.

1.4. Tipos de bases de datos.

En el mundo existen varios tipos de bases de datos que brindan una gran ventaja a la hora de manejar la información dentro de los cuales encontramos:

- Según la variabilidad de los datos almacenados:

Capítulo 1

Bases de datos estáticas: éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo [3].

Bases de datos dinámicas: éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. [4].

➤ Según el contenido:

Bases de datos bibliográficas: solo contienen un representante de la fuente primaria, que permite localizarla [5].

Bases de datos de texto completo: almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas [5].

Después del análisis realizado de los tipos de base de datos, se seleccionó las bases de datos dinámicas, porque a diferencia de los demás tipos de bases de datos la información esta en constante cambio, ya que sobre ella se pueden realizar operaciones tales como: adicionar, actualizar, eliminar, etc. Además de que son ideales para almacenar grandes volúmenes de información.

1.5. Modelos de bases de datos.

Un modelo de datos es una "descripción" de un contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos [6].

Algunos de los modelos con frecuencia utilizados en las bases de datos son:

Bases de datos jerárquicas.

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado

Capítulo 1

raíz, y a los nodos que no tienen hijos se los conoce como hojas. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos [7].

Base de datos de red.

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres, posibilidad no permitida en el modelo jerárquico. Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales [7].

Base de datos relacional.

En este modelo, todos los datos son almacenados en relaciones, y debido a que cada relación es un conjunto, entonces el orden en que se almacenen los datos no tiene mayor 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 casual de la base de datos. La información puede ser recuperada o almacenada por medio de «consultas» que ofrecen una amplia flexibilidad y poder para administrar la información [8].

Bases de datos orientadas a objetos.

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos o sea estado y comportamiento. Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

Encapsulación: propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.

Herencia: propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

Polimorfismo: propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos [7].

Bases de datos documentales.

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesauro es un sistema de índices optimizado para este tipo de bases de datos [7].

Base de datos deductivas.

Es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática [7].

Después del análisis de los modelos de bases de datos realizado anteriormente se elige el modelo relacional, ya que aporta altos niveles de fiabilidad e integridad en el manejo de grandes cantidades de datos, además de que lo que marca la diferencia entre el modelo relacional de los demás modelos es que basa expresamente su funcionamiento sobre un modelo matemático muy específico: el álgebra y el cálculo relacional, así como la progresiva adopción de un número de reglas de integridad relacional y de formas normales.

1.6. Sistemas gestores de bases de datos.

Un sistema de gestión de la base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma [9]. Es la aplicación que interacciona con los usuarios de los programas de aplicación y la base de datos.

Un SGBD proporciona los siguientes servicios:

Capítulo 1

Permite la definición de la base de datos mediante el lenguaje de definición de datos. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos. Todo esto se almacenará en la base de datos.

Permite la inserción, actualización, eliminación y consulta de datos mediante el lenguaje de manejo de datos. El hecho de disponer de un lenguaje para realizar consultas reduce el problema de los sistemas de ficheros.

A diferencia de los sistemas de ficheros, el SGBD gestiona la estructura física de los datos y su almacenamiento. Con esta funcionalidad, el SGBD se convierte en una herramienta de gran utilidad. Sin embargo, desde el punto de vista del usuario, se podría discutir que los SGBD han hecho las cosas más complicadas, ya que ahora los usuarios ven más datos de los que realmente quieren o necesitan, puesto que ven la base de datos completa. Consientes de este problema, los SGBD proporcionan un mecanismo de vistas que permite que cada usuario tenga su propia vista o visión de la base de datos. El lenguaje de definición de datos permite definir vistas como subconjuntos de la base de datos.

Las vistas, además de reducir la complejidad permitiendo que cada usuario vea sólo la parte de la base de datos que necesita, tienen otras ventajas:

Proporcionan un nivel de seguridad, ya que permiten excluir datos para que ciertos usuarios no los vean.

Proporcionan un mecanismo para que los usuarios vean los datos en el formato que deseen, representa una imagen consistente y permanente de la base de datos, incluso si la base de datos cambia su estructura.

Ventajas de los SGBD:

Facilidad de manejo de grandes volúmenes de información.

Gran velocidad en muy poco tiempo.

Independencia del tratamiento de información.

Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.

No hay duplicidad de información, comprobación de información en el momento de introducir la misma.

Integridad referencial al terminar los registros.

Desventajas de los SGBD:

El costo de actualización del hardware y software son muy elevados.

El costo (salario o remuneración) del administrador de la base de datos es grande.

El mal diseño de esta puede originar problemas a futuro.

Un mal adiestramiento a los usuarios puede originar problemas a futuro.

Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.

Generan campos vacíos en exceso.

El mal diseño de seguridad genera problemas en esta.

1.6.1. Oracle.

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es el manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Oracle corre en PCs, microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Fue fabricado por Oracle Corporation [8].

Ventajas de SGBD Oracle:

Es una suite de productos que ofrece una gran variedad de herramientas.

Soporte de transacciones: es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica. Un SGBD se dice transaccional si es capaz de mantener la integridad de los datos, haciendo que estas transacciones no puedan finalizar en un estado intermedio.

Estabilidad: un sistema es estable cuando su nivel de fallos disminuye por debajo de un determinado umbral, que varía dependiendo de la estabilidad que se requiera.

Escalabilidad: es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

Multiplataforma: puede funcionar en diversas plataformas. Por ejemplo, una aplicación multiplataforma podría ejecutarse en Windows, en Linux, etc.

Desventajas:

El costo de mantenimiento es alto.

Es un software propietario, por lo que sólo lo pueden adquirir empresas con grandes presupuestos.

Otro aspecto a criticar es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, que incrementan el nivel de exposición de los usuarios.

1.6.2. PostgreSQL.

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD [10]. Pertenece al grupo de licencias de software Libre.

Su desarrollo no es manejado por una sola compañía, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Poco a poco muchos desarrolladores entusiastas de los motores de base de datos se unieron al proyecto y entre todos comenzaron a incorporar muchas características al motor, dentro de las que se encuentran:

Alta concurrencia: PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para:

Números de precisión arbitraria.

Texto de largo ilimitado.

Figuras geométricas (con una variedad de funciones asociadas).

Vistas

Herencia de tablas.

Ventajas:

Instalación ilimitada: con frecuencia las bases de datos comerciales son instaladas en más servidores de lo que permite la licencia, siendo esta la principal fuente de incumplimiento de licencia. PostgreSQL no tiene costo asociado a la licencia del software, por lo que al trabajar con el mismo no se corre el riesgo de ser demandado por violar acuerdos de licencia.

Cuenta con herramientas gráficas de diseño y administración de bases de datos.

Capítulo 1

Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows.

Extensible: el código fuente está disponible para todos sin costo.

Desventajas:

Es lento a la hora de importar datos desde archivos de texto y al gestionar grandes o pequeñas bases de datos.

La sintaxis de algunos de sus comandos no es nada intuitiva, por lo que en un principio la comunicación con el usuario se hace más difícil.

1.6.3. Microsoft SQL Server.

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea [11].

Ventajas:

Incluye también un potente entorno gráfico de administración.

Permite trabajar en modo cliente-servidor.

Permite administrar información de otros servidores de datos.

Permite exportar o importar bases de datos completas, por lo que no es necesaria la creación de una nueva base de datos si se desea cambiar, por ejemplo, de Access a SQL Server.

Desventajas:

Tiempo de respuesta largo.

Utiliza gran cantidad de memoria RAM.

Una mala implementación de los tipos de datos variables.

No es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

1.6.4. MySQL.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento GPL.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollo interno, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

Amplio subconjunto del lenguaje SQL.

Transacciones y claves foráneas.

Conectividad segura.

Replicación.

Ventajas:

Multiplataforma: puede funcionar en diversas plataformas, ya sea Linux, Windows, etc.

MySQL es software de fuente abierta: fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar.

Presenta motores de almacenamiento independientes, MyISAM para lecturas rápidas e InnoDB para transacciones e integridad referencial.

Soporte de transacciones: es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica. Un SGBD se dice transaccional si es capaz de mantener la integridad de los datos, haciendo que estas transacciones no puedan finalizar en un estado intermedio.

Estabilidad: un sistema es estable cuando su nivel de fallos disminuye por debajo de un determinado umbral, que varía dependiendo de la estabilidad que se requiera.

Capítulo 1

Escalabilidad: es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

Tiene completo soporte para operadores y funciones en cláusulas select y where.

Es relativamente sencilla de manejar, para un usuario medio-avanzado.

Desventajas:

Carece de soporte para transacciones.

No soporta vistas.

Después de haber analizado las características, ventajas y desventajas de los SGBD mencionados anteriormente, se decide utilizar el gestor de bases de datos MySQL para dar cumplimiento a la línea base de la arquitectura del proyecto, debido a que el SIGM estará integrado al sistema SISalud y este utiliza software libre. Además de que la extensiva reutilización del código dentro del software produce características altamente funcionales, ha dado lugar a un sistema de administración de base de datos de alta velocidad, compactación, estabilidad y facilidad de despliegue.

1.7. Herramientas CASE.

Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software (Investigación Preliminar, Análisis, Diseño, Implementación e Instalación.). También son un conjunto de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases [12].

Estas herramientas pueden proveer muchos beneficios en todas las etapas del proceso de desarrollo de software, algunas de ellas son:

Verificar el uso de todos los elementos en el sistema diseñado. .

Automatizar el dibujo de diagramas.

Ayudar en la documentación del sistema.

Ayudar en la creación de relaciones en la Base de Datos. .

Generar estructuras de código.

Mejora de la calidad de los desarrollos realizados.

El aumento de la productividad.

Las herramientas CASE también permiten a los analistas tener más tiempo para el análisis y diseño y minimizar el tiempo para codificar y probar. Algunas herramientas CASE son sólo para la fase de diseño. Otras, son sólo generadoras de código. Además existen herramientas de análisis y diseño que tienen una visión de desarrollo orientada a procesos sin la capacidad de modelamiento.

1.7.1. ER/Studio.

ER/Studio es una herramienta para el diseño de base de datos, que brinda productividad en su diseño, generación, y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada, permite visualizar la estructura, los elementos importantes, y optimizar el diseño de la base de datos. Además genera tablas automáticamente y miles de líneas de procedimientos almacenados y código trigger para las principales bases de datos.

Con ER/Studio se puede hacer también la ingeniería inversa, así como generar automáticamente tablas, vistas, índices, reglas de integridad referencial (llaves primarias, llaves foráneas), valores por defecto y restricciones de campos y dominios.

Esta herramienta mejora la productividad entre los desarrolladores cuando los diseños de la base de datos son divididos, compartidos, y reutilizados. También asegura consistencia, rehúso, e integración de los datos del proyecto al proporcionar el bosquejo que las IT (Information Technology, por sus siglas en ingles) necesitan para entender, analizar y comunicar la estructura de la base de datos.

ER/Studio tiene las siguientes características:

- ✓ Actualización del diseño de la base de datos física:

- **Gestión de la seguridad.** – gestión de usuarios, permisos y roles en cualquier modelo físico o lógico permitiendo propagar los permisos desde el modelo lógico al físico una vez que se crea éste y sincronizar los usuarios, roles y permisos entre los modelos y una base de datos.
 - **Pronóstico de la capacidad** – pronosticar el número de filas y ratios de crecimiento de filas para las tablas y calcular los requisitos de almacenaje futuros.
 - **Generación de sentencias ALTER SQL** – generar automáticamente código ALTER cuando se compara dos modelos físicos o se compara un modelo físico con un archivo SQL
- ✓ Actualización de la gestión del modelo
- **Herencia de Dominios** – ER/Studio 7.0 soporta la herencia de estructuras de dominios. Deriva nuevos dominios de uno ya existente para una plataforma específica o permite construir conjuntos de dominios relacionados.
 - **Mapeos definidos por el usuario** – ER/Studio 7.0 permite al ingeniero de software definir sus propios mapeos entre un modelo físico y lógico
 - **Añadir nuevos modelos** – pudiendo importar modelos ER/Studio o DT/Studio usando el asistente “Add New Physical Model” , o bien utilizando técnicas de ingeniería inversa importando modelos ya existentes o a partir de un archivo SQL
 - **Utilidades de sincronización de sub-modelos** – sincronizando sub-modelos entre los distintos modelos.
 - **Actualización del sub-modelo** – Ofrece un control detallado sobre la construcción y mantenimiento de sub-modelos, permitiendo añadir y borrar objetos relacionados.

1.7.2. DBDesigner.

DBDesigner es un sistema totalmente visual de diseño de bases de datos, que combina características y funciones profesionales con un diseño simple, muy claro y fácil de usar, a fin de ofrecerle un método efectivo para gestionar sus bases de datos [13]. Además permite administrar las mismas, diseñar tablas, hacer peticiones SQL manuales y otras funcionalidades, como ingeniería inversa en MySQL. El programa dispone además de una interfaz profesional y de detallados manuales de uso. Está disponible tanto para sistemas GNU/Linux como para Microsoft Windows 2000/XP. Ha sido creado para funcionar íntegramente

con MySQL, aunque también puede ser capaz de soportar otros sistemas por medio de la inclusión de plugins, por ahora solo PostgreSQL.

1.7.3. Rational Rose.

Rational Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas [14]. Una de las grandes ventajas de esta herramienta es que utiliza la notación estándar en la arquitectura de Software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común. Otra de sus ventajas es que los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. Además de brindar una serie de beneficios, Rational Rose es un software propietario, lo que hace que tenga desventajas respecto a otras herramientas que son libres.

1.7.4. Visual Paradigm.

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. Esta herramienta ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Para el modelado del diseño de la base de datos se selecciona ER/Studio, ya que genera tablas automáticamente y miles de líneas de procedimientos almacenados y códigos trigger para las principales bases de datos. Permite diferenciar el modelo lógico del modelo físico a diferencia del resto de las herramientas CASE. Proporciona a los desarrolladores una plataforma con interfaz amigable que les permite diseñar un producto con calidad de forma rápida. Además como los diseños de cada módulo, a partir de los cuales se partió para el nuevo diseño de la base de datos del SIGM, estaban diseñados en ER/Studio se decide no migrar para otra herramienta facilitando el trabajo y ahorrando tiempo.

1.8. Herramientas de Desarrollo.

1.8.1. EMS Data Generator for MySQL.

EMS Data Generator for MySQL es un utilitario para generar datos de prueba a tablas de bases de datos MySQL. La aplicación asistente te permite definir tablas y campos para generar datos, configurar valores de rangos, generar campos de tipo carácter por máscara, cargar valores desde archivos para campos de tipo objetos binarios (BLOB), obtener listas de valores desde consultas SQL y muchas otras características más para generar datos de prueba de manera simple y directa.

Dentro de sus características principales podemos encontrar:

El asistente posee una interfaz de usuario amigable.

Generación de datos a varias tablas desde diferentes bases de datos en un solo ordenador anfitrión (host).

Soporte para todos los tipos de datos de MySQL.

Diferentes tipos de generación por cada campo, incluyendo lista, azar, generación incremental de datos y más.

Capacidad para usar resultados de consultas SQL como lista de valores para la generación de datos.

Control automático sobre integridad referencial para la generación de datos a tablas vinculadas.

Amplia variedad de generación de parámetros para cada tipo de campo.

Capacidad para configurar valores Null (nulos) para ciertos casos.

Posibilidad de salvar todos los parámetros de generación, para comenzar la sesión del asistente actual.

Conclusiones.

Después de realizar un análisis del estado de arte de las principales metodologías y tecnologías actuales en el mundo, se decide utilizar MySQL como gestor de base de datos, la herramienta CASE a utilizar ER/Studio, para el diseño de base de datos. Como modelo de base de datos se eligió el relacional. La

Capítulo 1

metodología de desarrollo seleccionada es la metodología RUP, y como herramienta de desarrollo para un llenado inteligente y pruebas de carga intensiva a la base de datos el EMS Data Generator for MySQL.

Capítulo 2

En este capítulo se describe la solución propuesta, concentrándose en la estrategia de integración de la solución con otros módulos o sistemas, la descripción de la arquitectura, y en la descripción de las principales tablas de cada módulo del SIGM.

2.1. Requisitos no funcionales.

La definición de las necesidades del sistema es un proceso complejo mediante el cual se especifican y validan las restricciones sobre las que deberá operar. Los requisitos se pueden clasificar en: requisitos funcionales y requisitos no funcionales, siendo estos últimos los que se analizaron para el desarrollo de esta tesis.

Los requisitos no funcionales (RNF) son propiedades o cualidades que el producto debe tener, debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable [15].

A continuación se referencia una lista de requisitos no funcionales que debe cumplir el SIGM.

1. RNF de rendimiento:

Los tiempos de respuestas deben ser generalmente rápidos al igual que la velocidad de procesamiento de la información.

2. RNF de soporte:

Se debe asegurar el soporte para los usuarios de manera que se puedan satisfacer sus necesidades a partir de mejoras, una vez puesta en marcha la aplicación. Para ello se crearán una serie de manuales de usuarios y videos tutoriales, y se mantendrá la asistencia a los usuarios.

3. RNF de seguridad:

El sistema debe tener un mecanismo propio para gestionar la seguridad a través de niveles de acceso a la información. Los permisos al ejecutar cualquier acción deben estar de acuerdo con el nivel jerárquico de acceso que presente el usuario en cada módulo, el cual es definido por los administradores del sistema.

4. RNF de software:

Se requiere para el funcionamiento del sistema disponer de un servidor que cuente con Sistema Operativo Linux, Apache 2.0 y MySQL 5.0 o versiones superiores. Los usuarios del sistema deberán contar con un navegador Internet Explorer 5.5 o Mozilla Firefox 2.0 o superior, para poder acceder a las opciones que brinda el sistema.

5. RNF de hardware:

Para el desarrollo y ejecución de la aplicación se necesitará:

Para el servidor:

- Microprocesador Pentium IV a 3.0 GHz
- 1GB de RAM o superior.

Para el Cliente:

- Microprocesador Pentium a 233 MHz (Recomendado: Pentium a 500MHz o superior).
- 64 MB de RAM (Recomendado: 128 MB RAM o superior).
- 52 MB de espacio de disco duro. (Recomendado: 120MB para la instalación completa del Internet Explorer 5.5).

6. RNF de persistencia:

La información debe almacenarse en base de datos con carácter permanente con el objetivo de poder realizar análisis de la misma con el transcurso de los años.

2.2. Estrategia de integración de la solución con otros módulos y sistemas.

A continuación se menciona una serie de pasos que se tuvieron en cuenta para realizar la integración de los módulos del SIGM:

1. Analizar los diseños de las bases de datos de cada módulo, teniendo en cuenta varios aspectos como: normalización, integridad de los datos, redundancia de la información, entre otros aspectos.
2. Comparar la información representada en las bases de datos antiguas con la información que necesitaba cada registro en particular.

Capítulo 2

3. Actualizar los datos, agregando la información nueva y eliminando de cada módulo los elementos innecesarios.
4. Rediseñar cada módulo, garantizando su normalización y la integridad de sus datos.
5. Proceder a la integración de todos los módulos, quedando diseñada una base de datos para el SIGM.

El SIGM tiene una estructura de siete módulos. Estos comprenden las funcionalidades que aseguran el desarrollo de la Genética Médica, mediante la recogida de datos que se almacenan en la base de datos, para su posterior análisis y estudio. Para la integración de los módulos del SIGM, se planteó como premisa que la nueva estructura de la base de datos quedaría con un módulo llamado datos generales, que sería el encargado de tener toda la información común entre los módulos. Este nuevo módulo estaría relacionado con los demás módulos, los cuales tendrían solamente la información asociada a ellos.

Antes de la integración existían entre todos los módulos un total de 346 tablas, 1579 columnas 332 llaves y 351 relaciones, actualmente la base de datos del SIGM cuenta con 256 tablas, 1194 columnas, 290 relaciones y 225 llaves.

A partir de un análisis comparativo realizado sobre esta información, se tomó el módulo Registro Cubano de Historias Clínicas (RECUHCLIN) como el módulo de datos generales, ya que este era el que más información común tenía con los demás. Para esto se tomó toda la información que tenía asociada el módulo RECUHCLIN y se eliminó la tabla Persona, todos los datos que tenía dicha tabla se le agregó a la tabla dg_pacientes, ya que esta tabla va a ser la encargada de recoger los datos primarios del paciente, estos datos fueron actualizados según el portal de Softel, quedando el módulo RECUHCLIN con sólo una tabla llamada hc_historia_clínica, que va a estar relacionada con el paciente, ya que cada paciente puede tener una historia clínica, y con otros módulos que necesiten esta información.

A continuación se hará una descripción detallada de las transformaciones que sufrieron cada uno de los módulos en el proceso de integración y de actualización de la información una vez consultado con el CNGM:

Módulo Registro Cubano de Gemelos (RECUGEM).

Capítulo 2

Para la integración de este módulo se tomaron los datos generales de la tabla gemelo (nombre, apellidos, CI, sexo, piel, las tablas correspondientes a la dirección del paciente) y se pasaron para la tabla *dg_pacientes*. Además las tablas *tbNom_EnfermedadEspecificas* y *tbNom_Enfermedad* fueron eliminadas ya que esta información se recoge actualmente en el módulo RECUEGEN. Anteriormente se había realizado un estudio a 15 000 gemelos, quienes tenían errores en sus CI o simplemente no tenían CI. Como es necesario incluirlos en la base de datos de SIGM, y no hay forma de comprobar si este paciente ya se había insertado en la base de datos mediante algún otro módulo, se agregó una tabla *gm_indocumentado*, que almacenará esta información. También se le agregaron varias tablas nomencladoras como: *gm_ncausa_fallecimiento*, *gm_nenseñanza*, *gm_nivel_escolar*. La información de estas tablas se recogía en forma de columnas de la tabla *gm_gemelos*, esto era insuficiente debido a que estas columnas tienen valores constantes. También se agregó una tabla *gm_ultimo_insertado* para recoger el último consecutivo de cada consejo, debido a que el código de pareja se conforma por el número de provincia, municipio, consejo popular y este consecutivo, esto hace más fácil que cuando se vaya a seleccionar el consecutivo de ese consejo popular sólo es ir a dicha tabla, tomar el número y sumarle 1, luego se actualiza ésta con el nuevo valor.

Antes de las actualizaciones el módulo RECUGEM tenía 31 tablas, 328 columnas y 32 relaciones (ver Anexo 1). Después de estas actualizaciones el módulo quedó con 31 tablas, 322 columnas y 30 relaciones

Módulo Registro Cubano de las Enfermedades Genéticas (RECUEGEN).

A la tabla *dg_pacientes* también se le agregó los datos de las tablas persona, consulta y diagnóstico, así como las tablas nomencladoras correspondientes a la dirección, porque esta información es común entre todos los módulos. El módulo RECUEGEN tenía, antes de la integración, 9 tablas, 46 columnas, 8 llaves y 8 relaciones (ver Anexo 2). Ahora sólo cuenta con dos tablas llamadas *eg_enfermedades_geneticas* (con el código de la enfermedad y el nombre) y *eg_clasificacion_enfermedad* (con el código de la clasificación y la clasificación), las cuales van estar relacionadas entre ellas ya que una clasificación tiene muchas enfermedades. La tabla *eg_enfermedades_geneticas* va a estar relacionada con *dg_pacientes* con una relación de muchos a muchos. Por lo que el modulo cuenta con, 2 tablas, 5 columnas y 1 relación.

Módulo Registro Cubano de Malformaciones Congénitas (RECUMAC).

Capítulo 2

Los datos primarios del paciente se pasaron para la tabla dg_paciente, los correspondientes a la dirección se añadieron a datos generales. Estas tablas sólo se relacionaban entre sí, no tenían relación con paciente, lo que no ayudaba a la organización de la información, ya que no estaba normalizado. Los datos de las tablas Padre, y Madre se pasaron a las tablas dg_padre y dg_madre que son especializaciones de dg_familiar.

Las tablas relacionadas con la tabla Factores_Ambientales, o sea: Droga, Fuma, Parto_pret, Alcohol, Rayos_X, Aborto, Hipertermia, Anestesia, se registraron en una sola tabla llamada mc_factores_ambientales, ya que en estas se recogía solamente el trimestre en el que sucedió el factor, por lo que resultó más factible concentrar toda la información en dicha tabla. La misma va a estar relacionada con la tabla mc_madre_embarazada.

Los datos de la tabla consanguin_paternal, que estaba relacionada con la tabla Factores_Genéticos, se adicionaron a la tabla mc_factores_geneticos, y la tabla Hijos que también estaba relacionada a la tabla Factores_Genéticos, se eliminó, porque cuando se hicieron las actualizaciones, se determinó que esta información no era necesaria. La tabla mc_factores_geneticos está relacionada con la tabla mc_madre_embarazada.

Después de las actualizaciones, y debido a una información específica que necesitaban los genetistas, a este módulo se añadieron 4 nuevas tablas nomencladoras: mc_clasificacion_malformacion, mc_simple_compuesta, mc_tipo_sindrome, mc_tipo_monogenica.

De 37 tablas, 160 columnas y 25 relaciones, que tenía anteriormente (ver Anexo 3), el módulo RECUMAC cuenta ahora con 18 tablas, 65 columnas y 17 relaciones.

Módulo Teleconsulta

La tabla consulta se va a llamar tl_sala_conferencia. Se mantiene la tabla tl_historial relacionada con tl_sala_conferencia y además se agrega una tabla tl_mensaje, que son los mensajes emitidos por los médicos que participan en la teleconsulta. La tabla tl_mensaje está relacionada con la tabla dg_medico.

La tabla consulta_usuario pasa a llamarse tl_participante_sala, la misma está relacionada con dg_medico, puesto que un médico puede ser un participante de sala.

Capítulo 2

Como anteriormente la tabla urgencia recogía el nombre y la prioridad de dicha urgencia, y ahora sólo se necesita la prioridad, se decidió que dicha tabla se eliminaría y pasaría como columna de la tabla tl_solicitud.

La tabla usuario_solicitud se llama tl_participante_discucion, y la misma se relaciona con dg_medico, ya que un participante puede ser un médico.

A este módulo se agrega una tabla tl_estado que se relaciona con tl_solicitud. La nueva tabla va a recoger el estado en que se encuentran las solicitudes.

La tabla tl_informe se mantiene, pero se le añade las siguientes columnas: resumen, diagnóstico, diagnóstico definitivo, recomendaciones, y aprobado; se elimina la columna informe, la que recogía lo antes mencionado en un solo campo, lo que no era útil.

La tabla tl_solicitud se relaciona con hc_historia_clinica, puesto que las historias clínicas pueden solicitarse para establecer una teleconsulta.

Las tablas que recogían la información respecto a: formato_solicitud, y objeto_formulario se eliminaron, porque no era necesario para el cliente recoger esta información.

Anteriormente, el módulo Teleconsulta contaba con 18 tablas, 93 columnas y 18 relaciones (ver Anexo 4), actualmente tiene 9 tablas, 50 columnas y 5 relaciones.

Módulo Registro Cubano de Retraso Mental (RECUREM).

Todos los datos primarios del paciente, las tablas nomencladoras correspondiente a la dirección, así como la información respecto a su familia, fueron pasados para el módulo de datos generales.

Se eliminaron las tablas que recogían la información sobre las habilidades funcionales por área de desarrollo, la vivienda, situación laboral, estado nutricional. Dentro de estas tablas se pueden mencionar: TCognoscitiva, TCond_Socializ, TComunicacion, Talimentacion, TRecibe_MINSAP, TVivienda, TAYuda_Tecnica, TServicio_Comunidad, TValoracion_Nutricional, TCausa_Desnutricion, y otras que se pueden apreciar en el anexo correspondiente al módulo. Todos estos datos se actualizaron con el cliente y se eliminaron, ya que se determinó que no era necesario almacenarlo.

Capítulo 2

Los datos del parto durante los siete primeros días, así como los de a partir del octavo día, que antes se recogían en las tablas: TParto, TTipo_Nac y TEventos_Ninos, se recogen en una sola tabla llamada rm_datos_del_parto.

Anteriormente, el módulo RECUREM contaba con 124 tablas, 464 columnas y 117 relaciones (ver Anexo 5), actualmente cuenta 72 tablas, 233 columnas y 77 relaciones.

Módulo Registro Cubano de Discapacitados (RECUDIS).

El módulo RECUDIS es el que menos cambios sufrió ya que la mayoría de la información se encontraba actualizada, por lo que solo fue pasado para el modulo datos generales los datos primarios del paciente y los datos de las tablas nomencladoras correspondientes a la dirección. Además las tablas que se encontraban como nomencladoras relacionadas con la tabla vivienda como: Higiene, Acceso_Transporte, Cond_Vivienda y CondVidaRestoFam, fueron eliminadas y pasadas como atributos de la tabla vivienda, ya que se determinó que estos eran campos de textos.

Este módulo contaba anteriormente con 47 tablas, 198 columnas y 45 relaciones (ver Anexo 6), actualmente cuenta con 27 tablas, 149 columnas y 26 relaciones.

2.3. Descripción de la arquitectura y fundamentación.

La base de datos del SIGM posee 256 tablas, donde se almacena la información de los siete módulos: RECUREM, RECUDIS, RECUHCL, Teleconsulta, RECUGEM, RECUEGEN y RECUMAC. Las principales tablas de la base de datos son dg_paciente, dg_his_perinatal, dis_discapacitado, rm_datos_clinico, tl_solicitud, mc_persona_recumac, gm_gemelo y gm_pareja. El sistema cuenta con una capa de Acceso a Datos cuyo objetivo es gestionar la información. La capa Acceso a Datos es generada a partir del Framework Symfony.

El sistema tiene como requisito que debe ser insertado en el sistema SISalud y debe cumplir con la arquitectura establecida para el mismo, debiendo desplegarse sobre Infomed. A continuación la figura 1 muestra el diagrama de despliegue, el cual se explica en detalles en el documento de arquitectura del SIGM.

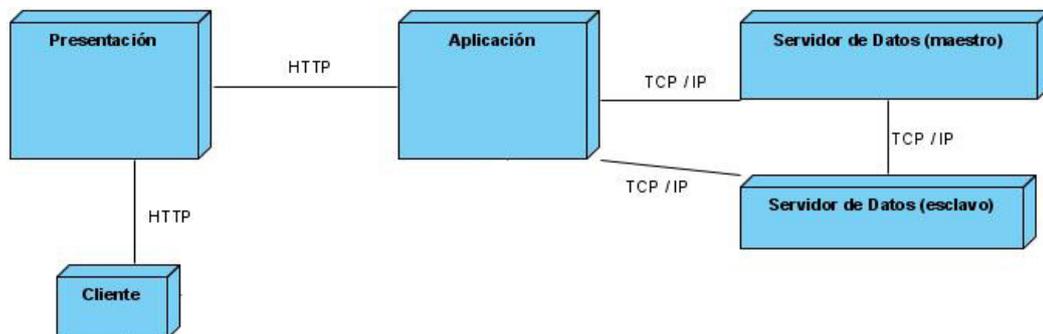


Figura 1 Diagrama de despliegue.

2.4. Modelo de datos.

El modelo de datos es un conjunto de conceptos reglas y convenciones que permiten describir y manipular los datos. [16].

A continuación se muestra el modelo de datos de cada uno de los módulos de la base de datos del SIGM:

Módulo Datos Generales.

Capítulo 2

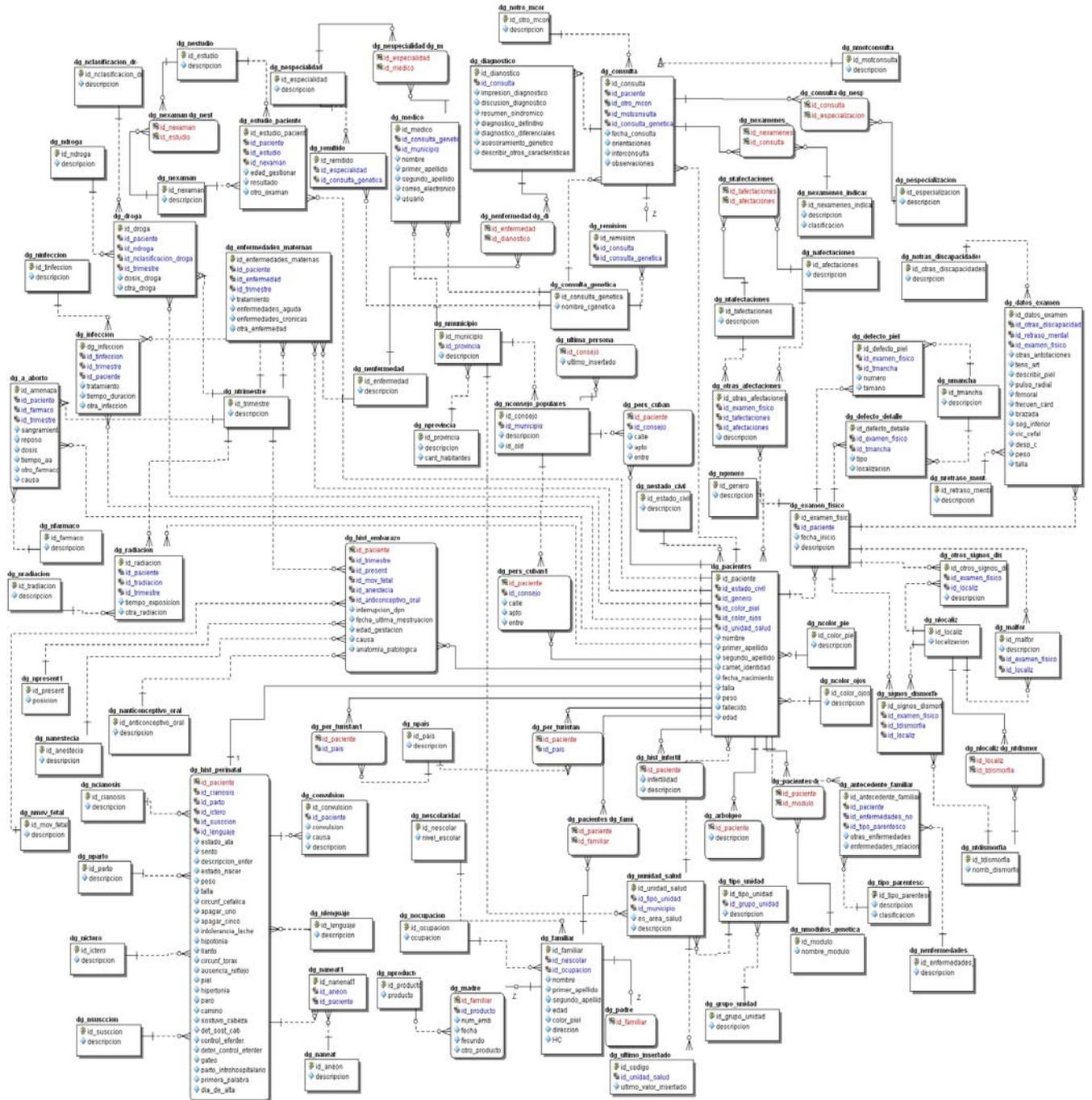


Figura 2. Modelo de datos del módulo Datos Generales.

Módulo RECUDIS.

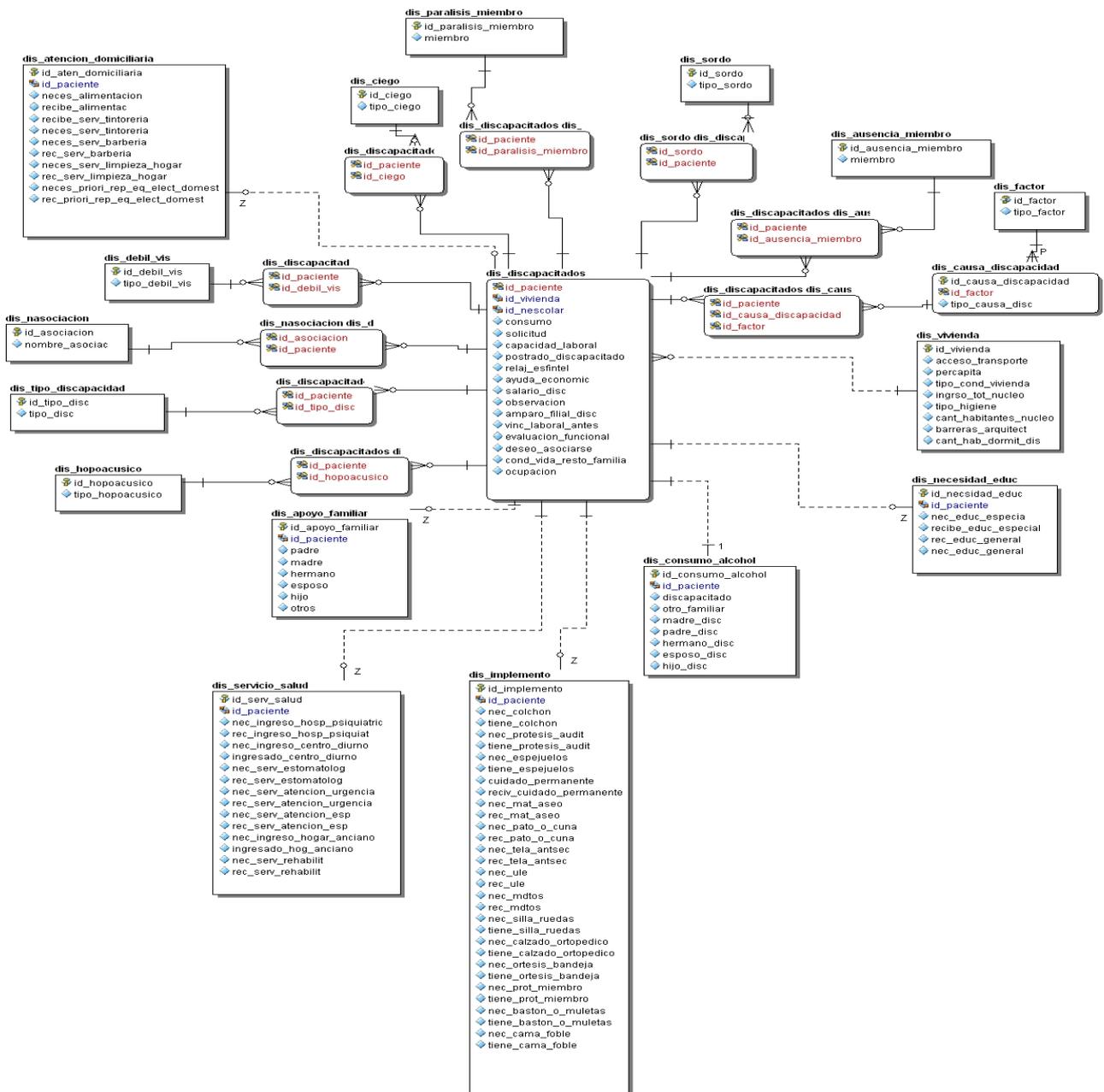


Figura 3. Modelo de datos del módulo RECUDIS.

Módulo RECUEGEN.

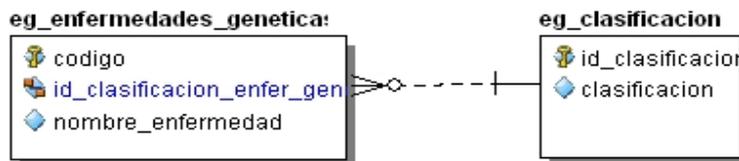


Figura 4. Modelo de datos del módulo RECUEGEN.

Módulo RECUHCLIN.

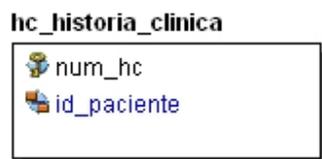


Figura 5. Modelo de datos del módulo RECUHCLIN.

Módulo RECUGEM.

Módulo Teleconsulta.

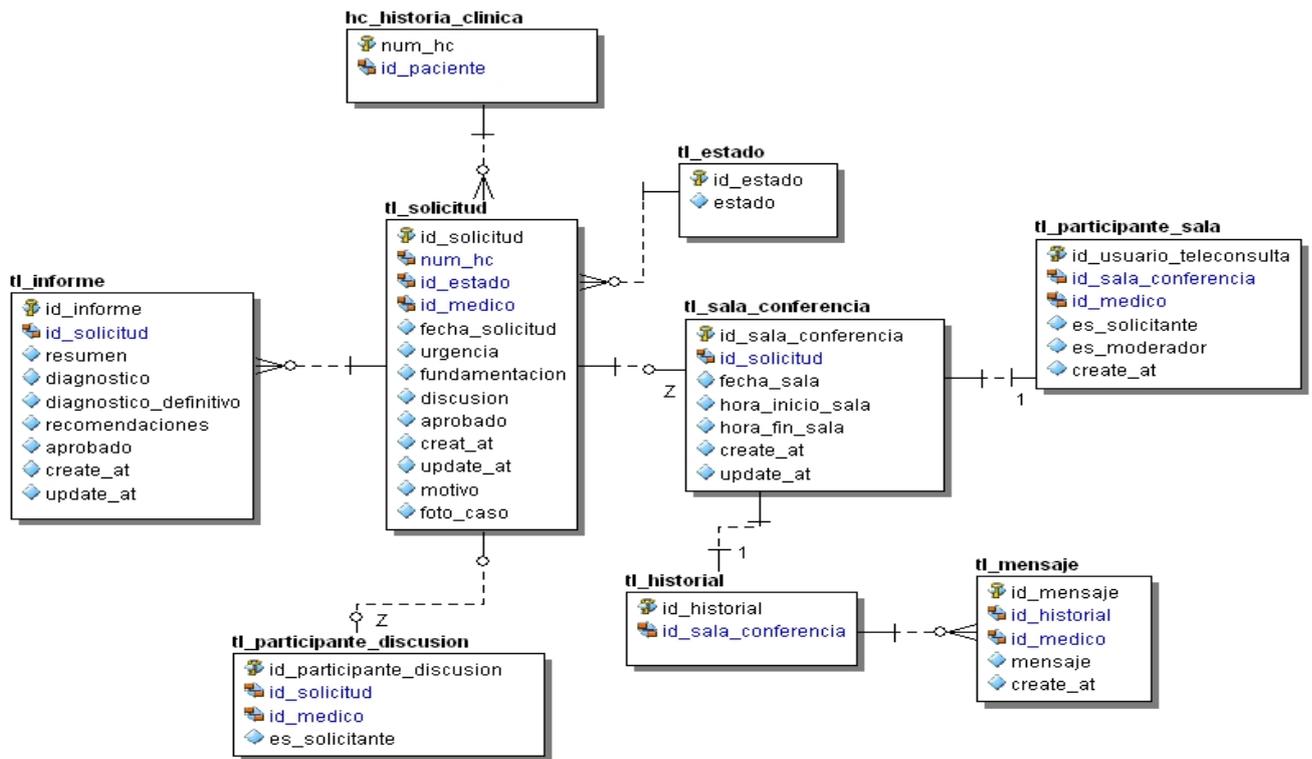


Figura 8. Modelo de datos del módulo Tele-consulta.

Módulo RECUMAC.

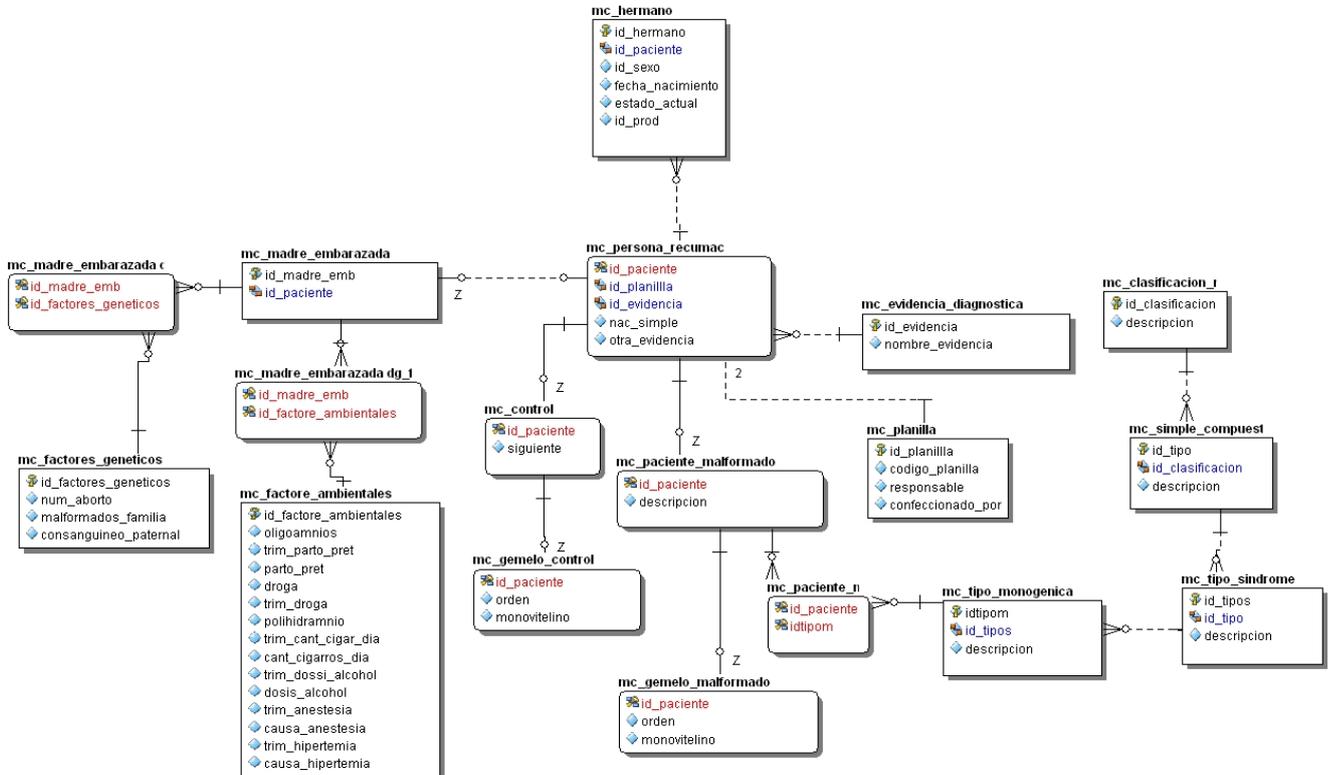


Figura 9. Modelo de datos del módulo RECUMAC.

2.5. Descripción de las tablas.

En este epígrafe se describen las tablas principales de cada módulo, para ver las demás dirigirse al expediente de proyecto.

Módulo Datos Generales.

Nombre: dg_a_aborto		
Descripción: Almacena la información de la amenaza de aborto.		
Atributo	Tipo	Descripción

Capítulo 2

id_amenaza	int	Identificador de la amenaza de aborto
Id_paciente	int	Identificador de la tabla dg_paciente(llave foránea)
Id_farmaco	varchar	Identificador de la tabla dg_nfarmaco(llave foránea)
Id_trimestre	int	Identificador de la tabla dg_ntrimestre(llave foránea)
dosis	float	Dosis que se usó del fármaco
tiempo_aa	varchar	Tiempo que se usó el fármaco
sangramiento	tinyint	Si tuvo sangramiento durante el embarazo
reposo	tinyint	Especifica si tubo reposo
causa	varchar	Especifica la causa de la amenaza

Nombre: dg_antecedente_familiar		
Descripción: Almacena la información de los antecedentes familiares		
Atributo	Tipo	Descripción
id_antecedente_familiar	int	Identificador
id_paciente	int	Identificador de la tabla dg_paciente(llave foránea)
id_enfermedades_no	int	Identificador de la tabla dg_nenfermedades_no (llave foránea)
Id_tipo_parentesco	int	Identificador de la tabla dg_tipo_parentesco (llave foránea)

Capítulo 2

Otras_enfermedades	varchar	Especifica otras enfermedades
Enfermedades_relacionadas	varchar	Especifica las enfermedades relacionadas

Nombre: dg_consulta		
Descripción: En esta tabla se almacenan los datos de las consultas.		
Atributo	Tipo	Descripción
fecha_consulta	Data	Fecha en que se realiza la consulta
orientaciones	text	las orientaciones que se le hacen al paciente
interconsultas	text	si el paciente ha sido remitido desde otra consulta
observación	text	observaciones del paciente que se hacen en la consulta
id_motconsulta	int	Identificador de la tabla dg_nmotconsulta (llave foránea)
id_consulta	int	Identificador
Id_consulta_genetica	int	Identificador de la tabla dg_consulta_genetica (llave foránea)
id_paciente	int	Identificador de la tabla dg_paciente (llave foránea)

Capítulo 2

id_otro_mon	int	Identificador de la tabla dg_notro_mcon (llave foránea)
-------------	-----	---------------------------------------------------------

Nombre: dg_datos_examen		
Descripción: Almacena la información del examen		
Atributo	Tipo	Descripción
id_dato_examen	int	Identificador
id_examen	int	Identificador de la tabla dg_examen_fisico (llave foránea)
id_afec_cabello	int	Identificador de la tabla dg_cabello (llave foránea)
Id_implantacion_cabello	varchar	Identificador de la tabla dg_nimplantacion_cabello (llave foránea)
Id_retraso_mental	varchar	Identificador de la tabla dg_retraso_mental (llave foránea)
Id_otras_discapacidades	varchar	Identificador de la tabla dg_notras_discapacidades (llave foránea)
Otras_antotaciones	varchar	Se describen otros datos del examen no muy relevantes
tens_art	float	Valor de la tensión arterial en esa consulta
Describir_piel	varchar	Descripción de la piel
pulso_radial	float	Valor del pulso radial

Capítulo 2

femoral	varchar	Descripción del femoral
frcuen_card	varchar	Descripción de la frecuencia cardiaca
brazada	float	Tamaño Brazada del paciente
seg_inferior	float	Tamaño del segmento inferior del paciente
cic_fetal	float	Tamaño circunferencia cefálica
desp_c	float	Tamaño de proporción del cráneo
peso	float	El peso del paciente
talla	float	Talla del paciente

Nombre: dg_diagnostico		
Descripción: Almacena la información del diagnostico del paciente		
Atributo	Tipo	Descripción
id_diagnostico	int	identificador
id_consulta	int	Identificador de la tabla dg_consulta (llave foránea)
impresión_diagnostico	text	Impresión diagnóstico
discusión_diagnostico	text	Discusión diagnóstico
resumen_sindromico	text	Resumen sindrómico
diagnostico_definitivo	text	Diagnóstico definitivo

Capítulo 2

diagnostico_diferenciales	text	Especifica los diagnósticos diferenciales
asesoramiento_genetico	text	Asesoramiento genético
describir_otras_caracteristicas	text	Describir otras características

Nombre: dg_enfermedades_maternas		
Descripción: Almacena la información de las enfermedades maternas		
Atributo	Tipo	Descripción
id_enfermedades_maternas	int	Identificador de la enfermedad materna
id_paciente	int	Identificador de la tabla dg_paciente (llave foránea)
id_enfermedad	int	Identificador de la tabla dg_enfermedad (llave foránea)
id_trimestre	int	Identificador de la tabla dg_ntrimestre (llave foránea)
tratamiento	text	Tratamiento que tuvo
enfermedades_agudas	varchar	Las enfermedades agudas
otras_enfermedades	varchar	Especifica otras enfermedades
enfermedades_cronicas	varchar	Las enfermedades crónicas

Capítulo 2

Nombre: dg_examen_fisico		
Descripción: En esta tabla se almacena los datos del examen físico.		
Atributo	Tipo	Descripción
id_examen	int	Identificador del examen físico
id_paciente	int	Identificador de la tabla dg_paciente (llave foránea)
fecha_inicio	date	Fecha de inicio del examen.
descripcion	varchar	Descripción del examen físico

Nombre: dg_familiar		
Descripción: En esta tabla se almacenan los datos comunes de la madre y el padre del paciente.		
Atributo	Tipo	Descripción
Id_familiar	int	Identificador que va a identificar el familiar.
id_nescolar	int	Identificador de la tabla dg_nescolaridad (llave foránea)
id_ocupacion	int	Identificador de la tabla dg_nocupacion (llave foránea)
nombre	Varchar	Almacena el nombre de la persona
primer_apellido	Varchar	Almacena el primer apellido de la persona.

Capítulo 2

segundo_apellido	Varchar	Almacena el segundo apellido de la persona.
edad	int	La edad del familiar.
Color_piel	varchar	El color de la piel del familiar
direccion	varchar	La dirección particular del familiar.

Nombre: dg_hist_embarazo		
Descripción: En esta tabla se almacenan los datos de la Historia del Embarazo.		
Atributo	Tipo	Descripción
id_paciente	int	Identificador de la tabla dg_paciente (llave primaria), identificador de historia del embarazo
id_present	int	Identificador de la tabla dg_npresent1 (llave foránea)
id_mov_fetal	int	Identificador de la tabla dg_mov_fetal (llave foránea)
id_trimestre	int	Identificador de la tabla dg_ntrimestre (llave foránea)
id_anticonceptivo_oral	int	Identificador de la tabla dg_nanticonceptivo_oral (llave foránea)
id_anestecia	int	Identificador de la tabla dg_nanestecia (llave foránea)

Capítulo 2

interrupción_dpn	tinyint	Interrupción durante el embarazo
fecha_ultima_menstruacion	date	Fecha de la ultima menstruación
edad_gestacion	int	Almacena la edad gestacionar
causa	varchar	Causa de la amenaza de aborto
anatomía_patologica	tinyint	Anatomía de la madre

Nombre: dg_hist_perinatal		
Descripción: En esta tabla se almacenan los datos de la historia perinatal del paciente.		
Atributo	Tipo	Descripción
id_paciente	Int	Es el identificador que identifica al paciente.(llave primaria)
id_ictero	int	Es el identificador de la tabla dg_nictero (llave foránea)
id_succion	int	Es el identificador de la tabla dg_nsuccion (llave foránea)
id_parto	int	Es el identificador de la tabla dg_nparto (llave foránea)
id_cianosis	int	Es el identificador de la tabla dg_ncianosis (llave foránea)
Id_lenguaje	int	Es el identificador de la tabla dg_nlenguaje (llave foránea)

Capítulo 2

estado_alta	int	Estado cuando se le da el alta
descripción_enfermedad	text	Descripción de la enfermedad que tuvo
estado_nacer	int	Estado al nacer el paciente
Peso	int	Peso del paciente al nacer
Talla	Int	Talla del paciente al nacer
circunf_cefalica	float	Circunferencia cefálica
apgar_uno	int	Apagar después de 1 min.
apgar_cinco	int	Apagar después de 5 min.
anotaciones_neo	text	Anotaciones de neonatología
intolerancia_leche	tinyint	Intolerancia a la leche (Si o No)
Hipotonia	tinyint	Hipotonía (Si o NO)
ausencia_reflejo	tinyint	Ausencia de reflejos (Si o No)
circunf_torax	varchar	Circunferencia del tórax
hipertonía	tinyint	Hipertonía (Si o No)
Sento	FLOAT	A que tiempo se sentó
Paro	FLOAT	A que tiempo se paró
Camino	FLOAT	A que tiempo caminó
sostuvo_cabeza	FLOAT	A que tiempo sostuvo la cabeza
det_sost_cab	TEXT	Detalles de sostuvo la cabeza

Capítulo 2

control_efinter	FLOAT	Control de esfínter
det_control_efinter	TEXT	Detalles del control de esfínter.
gateo	FLOAT	A que tiempo gateó
Piel	varchar	Descripción de la piel
parto_introhospitalario	tinyint	Especifica si tubo parto intra-hospitalario
Llanto	varchar	Como tuvo el llanto al nacer
otra_nanetat	varchar	Especifica Anotaciones antología
primera_palabra	int	Cantidad de letras de la primera palabra
dia_de_alta	int	Especifica el día de alta

Nombre: dg_medico		
Descripción: En esta tabla se almacenan los datos del médico		
Atributo	Tipo	Descripción
id_medico	int	Identificador del médico
id_consulta_genetica	int	Identificador de la tabla dg_consulta_genetica (llave foránea)
id_municipio	int	Identificador de la tabla dg_nmunicipio (llave foránea)
nombre	varchar	Nombre del médico
primer_apellidos	varchar	Primer apellido del médico

Capítulo 2

segundo_apellidos	varchar	Segundo apellido
correo_electronico	varchar	Correo del médico
usuario	varchar	Usuario del médico

Nombre: dg_pacientes		
Descripción: En esta tabla se almacenan los datos referentes al paciente		
Atributo	Tipo	Descripción
id_paciente	int	Identificador que va a tener el paciente.
id_color_ojos	int	Identificador de la tabla dg_ncolor_ojos (llave foránea)
id_estado_civil	int	Identificador de la tabla dg_nestado_civil (llave foránea)
id_ncolor_piel	int	Identificador de la tabla dg_ncolor_piel (llave foránea)
id_unidad_salud	int	Identificador de la tabla dg_nunidad_salud (llave foránea)
id_genero	int	Identificador de la tabla dg_ngenero (llave foránea)
talla	float	La estatura del paciente.
peso	float	El peso corporal del paciente.
carnet_identidad	Varchar	Carne de identidad del paciente.

Capítulo 2

nombre	Varchar	Almacena el nombre del paciente.
fecha_nacimiento	date	Almacena la fecha de nacimiento del paciente.
fallecido	tinyint	Almacena si el paciente es fallecido.
primer_apellido	Varchar	Almacena el primer apellido del paciente.
segundo_apellido	Varchar	Almacena el segundo apellido del paciente
edad	int	Almacena la edad de la paciente

Para el diseño de la base de datos del SIGM se utilizaron varias tablas nomencladoras, estas son aquellas que poseen datos finitos conocidos.

Las tablas nomencladoras de la base de datos se pueden reconocer por su nombre, debido a que comienzan con el nombre del módulo seguido por una “n”, por ejemplo: dg_ncolor_piel, dg_ncolor_ojos, entre otros. Además están compuestas por un identificador y un campo “descripción”, este campo especifica la información referente a cada tabla. Un ejemplo de esto se muestra a continuación:

Nombre: dg_nafectaciones		
Descripción: En esta tabla hace referencia a las afectaciones		
Atributo	Tipo	Descripción
id_afectaciones	int	Identificador
descripcion	varchar	Especifica la afectaciones

Módulo RECUHCLIN.

Capítulo 2

Nombre: hc_historia_clinica		
Descripción: En esta tabla se recoge el numero de la historia clínica.		
Atributo	Tipo	Descripción
num_hc	varchar	Identificador de la historia clínica
id_paciente	int	Identificador de la tabla dg_paciente (llave foránea)

Módulo RECUDIS.

Nombre: dis_apoyo_familiar		
Descripción: Contiene los datos del apoyo familiar		
Atributo	Tipo	Descripción
Id_apoyofam	int	Identificador de apoyo familiar
Id_Discapacitado	int	Identificador de la tabla dis_discapacitado (llave foránea)
Padre	TINYINT	Especifica si recibe ayuda del padre
Madre	TINYINT	Especifica si recibe ayuda de la madre
Hijo	TINYINT	Especifica si recibe ayuda del hijo
Hermano	TINYINT	Especifica si recibe ayuda del hermano
Esposa	TINYINT	Especifica si recibe ayuda de la esposa
Otro	TINYINT	Especifica si recibe ayuda de otro familiar

Capítulo 2

Nombre: dis_atencion_domiciliaria		
Descripción: Contiene los datos sobre la atención domiciliaria.		
Atributo	Tipo	Descripción
Id_Atenc_Domic	int	Identificador de la atención domiciliaria
Id_Discapacitado	int	Identificador de la tabla dis_discapacitado (llave foránea)
Neces_Alimentacion	TINYINT	Especifica si necesita alimentación
Recibe_Aliment	TINYINT	Especifica si recibe alimentación
Neces_Serv_Tintoreria	TINYINT	Especifica si necesita servicio de tintorería
Recibe_Serv_Tintoreria	TINYINT	Especifica si recibe servicio de tintorería
Neces_Serv_Barberia	TINYINT	Especifica si necesita servicio de barbería
Recibe_Serv_Barberia	TINYINT	Especifica si recibe servicio de barbería
Neces_Serv_Limp_Hogar	TINYINT	Especifica si necesita servicio de limpieza del hogar
Recibe_Serv_Limp_Hogar	TINYINT	Especifica si recibe servicio de limpieza del hogar
Neces_Priori_Rep_Eq_Elect_Domest	TINYINT	Especifica si necesita prioridad en la reparación de equipos electrodomésticos.
Recibe_Priori_Rep_Eq_Elect_Domest	TINYINT	Especifica si recibe prioridad en la reparación de equipos electrodomésticos.

Capítulo 2

Nombre: dis_consumo_alcohol		
Descripción: Almacena la información del consumo de alcohol		
Atributo	Tipo	Descripción
Id_Cons_Alcohol	int	Identificador de consumo de alcohol
Id_Discapacitado	int	Identificador de la tabla dis_discapacitado (llave foránea)
Discapacitado	Varchar	Especifica si el discapacitado consume alcohol
Otros_Famil	Varchar	Especifica si otro familiar consume alcohol
Madre_Disc	Varchar	Especifica si la madre consume alcohol
Padre_Disc	Varchar	Especifica si el padre consume alcohol
Hermano_Disc	Varchar	Especifica si el hermano consume alcohol
Esposo_Disc	Varchar	Especifica si el esposo consume alcohol
Hijo_Disc	Varchar	Especifica si el hijo consume alcohol

Nombre: dis_discapacitado		
Descripción: En esta tabla se almacenan los datos específicos del discapacitado		
Atributo	Tipo	Descripción
Id_Discapacitado	int	Identificador del discapacitado

Capítulo 2

id_paciente	int	Identificador de la tabla dg_paciente (llave foránea)
Id_Vivienda	int	Identificador de la tabla dis_vivienda (llave foránea)
consumo	varchar	Especifica el consumo
id_nescolar	varchar	Identificador que indica la escolaridad de la tabla dg_nescolaridad (llave foránea)
SolicEmpleoAntes	Tinyint	Especifica si el discapacitado ha estado empleado antes
capacidad_laboral	Tinyint	Especifica si trabaja
Postrado_disc	Tinyint	Especifica si el discapacitado está postrado.
Relaj_Esfintel	Tinyint	Especifica si el discapacitado necesita relajación de esfínter
Ayuda_Econ	float	Especifica la ayuda económica que recibe
Salario_Disc	Float	Especifica el salario del discapacitado
observacion	text	Especifica algunas observaciones
Amparo_Filiar_Disc	TINYINT	Especifica si recibe amparo filial
VincLaboralAntes	varchar	Especifica donde estuvo vinculado laboralmente antes
Evaluacion_funcional	varchar	Se describe una evaluación funcional
Deseos_asociarse	varchar	Se describe si tiene deseo de asociarse

Capítulo 2

cond_vida_resto_fmilia	varchar	Se describe la condición de vida del resto de la familia
ocupacion	varchar	Especifica la ocupación

Nombre: dis_implemento		
Descripción: En esta tabla se almacenan los datos referentes a todos los implementos del discapacitado.		
Atributo	Tipo	Descripción
Id_Implemento	int	Identificador de implemento
Id_Discapacitado	int	Identificador de la tabla dis_discapacitado (llave foránea)
Neces_Colchon	Tinyint	Especifica si el discapacitado necesita colchón.
Tiene_Colchon	Tinyint	Especifica si el discapacitado tiene colchón.
Neces_Protesis_Audit	Tinyint	Especifica si el discapacitado necesita prótesis auditiva.
Tiene_Protesis_Audit	Tinyint	Especifica si el discapacitado tiene prótesis auditivas.
Neces_Espejuelos	Tinyint	Especifica si el discapacitado necesita espejuelos.
Tiene_Espejuelos	Tinyint	Especifica si el discapacitado tiene espejuelos

Capítulo 2

Cuidados_Permanentes	Tinyint	Especifica si el discapacitado necesita cuidados permanentes.
Recibe_Cuid_Permat	Tinyint	Especifica si el discapacitado recibe cuidados permanentes.
Neces_Mat_Aseo	Tinyint	Especifica si el discapacitado necesita material de aseo.
Recibe_Mat_Aseo	Tinyint	Especifica si el discapacitado recibe material de aseo.
Neces_PatoOCuna	Tinyint	Especifica si el discapacitado necesita pato o cuña
Recib_PatoOCuna	Tinyint	Especifica si el discapacitado recibe pato o cuña.
Neces_Tela_Antsect	Tinyint	Especifica si el discapacitado necesita el servicio de rehabilitación.
Recibe_Tela_Antsect	Tinyint	Especifica si el discapacitado recibe tela antiséptica.
Necesi_Hule	Tinyint	Especifica si el discapacitado necesita hule.
Recibe_Hule	Tinyint	Especifica si el discapacitado recibe hule.
Neces_Mdtos	Tinyint	Especifica si el discapacitado necesita medicamentos.
Recibe__Mdtos	Tinyint	Especifica si el discapacitado recibe medicamentos.
Neces_Silla_Rueda	Tinyint	Especifica si el discapacitado necesita silla

Capítulo 2

		de ruedas.
Tiene_Silla_Rueda	Tinyint	Especifica si el discapacitado tiene silla de ruedas.
Neces_Calzado_Ortop	Tinyint	Especifica si el discapacitado necesita calzado ortopédico.
Tiene_Calzado_Ortop	Tinyint	Especifica si el discapacitado tiene calzado ortopédico.
Neces_Ortesis_Bandeja	Tinyint	Especifica si el discapacitado necesita ortesis y bandeja
Tiene_Ortesis_Bandeja	Tinyint	Especifica si el discapacitado tiene ortesis y bandeja.
Neces_Protesis_Miembro	Tinyint	Especifica si el discapacitado necesita prótesis de miembro.
Tiene_Protesis_Miembro	Tinyint	Especifica si el discapacitado tiene prótesis de miembro.
Neces_BastonesOMuletas	Tinyint	Especifica si el discapacitado necesita bastones o muletas.
Tiene_BastonesOMuletas	Tinyint	Especifica si el discapacitado tiene Bastones o Muletas.
Neces_Cama_Fowler	Tinyint	Especifica si el discapacitado necesita cama flower.
Tiene_Cama_Fowler	Tinyint	Especifica si el discapacitado tiene cama flower.

Capítulo 2

Nombre: dis_necesidad_educacional		
Descripción: En esta tabla se almacena los datos de las necesidades educacional de los discapacitados.		
Atributo	Tipo	Descripción
Id_Nec_Educ	int	Identificador de necesidad educacional
Id_Discapacitado	int	Identificador de la tabla dis_discapacitado (llave foránea)
Neces_Educ_Esp	Tinyint	Especifica si el discapacitado necesita o no educación especial.
Recibe_Educ_Esp	Tinyint	Especifica si el discapacitado recibe o no educación especial.
Neces_Educ_Gral	Tinyint	Especifica si el discapacitado necesita educación general.
Recibe_Educ_Gral	Tinyint	Especifica si el discapacitado recibe o no educación general.

Nombre: dis_servicio_salud		
Descripción: En esta tabla se almacenan los datos referentes a los servicios de salud relacionados con el discapacitado		
Atributo	Tipo	Descripción
Id_Serv_Salud	int	Identificador del servicio de salud
Id_Discapacitado	int	Identificador de la tabla dis_discapacitado

Capítulo 2

		(llave foránea)
Neces_Ing_Hopt_PSquiat	Tinyint	Especifica si el discapacitado necesita ingreso en hospital psiquiátrico.
Recibe_Ing_Hosp_Psiquiat	Tinyint	Especifica si el discapacitado recibe la ingreso en hospital psiquiátrico.
Neces_Ing_Cent_Diurno	Tinyint	Especifica si el discapacitado necesita ingreso en un centro diurno.
Ingresado_Cent_Diurno	Tinyint	Especifica si el discapacitado recibe el ingreso en un centro diurno.
Neces_Serv_Estomat	Tinyint	Especifica si el discapacitado necesita servicio de estomatología.
Recibe_Serv_Estomat	Tinyint	Especifica si el discapacitado recibe el servicio de estomatología.
Neces_Serv_Atenc_Urgencia	Tinyint	Especifica si el discapacitado necesita servicio de atención de urgencias.
Recibe_Serv_Atenc_Urgencia	Tinyint	Especifica si el discapacitado recibe el servicio de atención de urgencias.
Neces_Serv_Atenc_esp	Tinyint	Especifica si el discapacitado necesita servicio de atención especial.
Recibe_Serv_atencion_Esp	Tinyint	Especifica si el discapacitado recibe el servicio de atención especial.
Neces_Ing_Hog_Anc	Tinyint	Especifica si el discapacitado necesita ingreso en un hogar de ancianos.

Capítulo 2

Ingresado_HA	Tinyint	Especifica si el discapacitado se encuentra ingresado en un hogar de anciano.
Neces_Serv_Rehabilit	Tinyint	Especifica si el discapacitado necesita el servicio de rehabilitación.
Recibe_Serv_Rehabilit	Tinyint	Especifica si el discapacitado recibe el servicio rehabilitación.

Nombre: dis_vivienda		
Descripción: En esta tabla se almacenan todos los datos referentes a la vivienda.		
Atributo	Tipo	Descripción
Id_Vivienda	int	Identificador de la vivienda
acceso_trasp	varchar	Especifica el acceso al transporte
Percapita	Float	Especifica el dinero per cápita del discapacitado.
tipo_condvivienda	varchar	Especifica el tipo de vivienda
Ingreso_Total_Nucleo	Float	Especifica el ingreso total del núcleo.
tipo_higiene	varchar	Especifica el tipo higiene
Cant_Habit_Nucleo	int	Especifica la cantidad de habitantes del núcleo.
Barreras_Arquitect	Tinyint	Especifica si la vivienda posee barreras arquitectónicas o no.
Cant_Habit_Dormit_Disc	int	Especifica la cantidad de personas que duermen en el dormitorio del discapacitado.

Módulo RECUMAC.

Nombre: mc_persona_recumac		
Descripción: Registro de las características específicas del paciente malformado		
Atributo	Tipo	Descripción
id_persona	int	Identificador de la persona
id_planilla	int	Identificador de la planilla
id_evidencia	int	Identificador de la evidencia
nac_simple	Tinyint	Se registra si ese nacimiento es simple o múltiple
otras_evidencias	Varchar	Especifica otras evidencias

Nombre: mc_planilla		
Descripción: Registro de los datos específicos del confeccionador de la planilla.		
Atributo	Tipo	Descripción
Id_planilla	int	Identificador de la planilla
codigo_planilla	Varchar	Se registra el código de la planilla.
responsable	National Varchar	Se registra el nombre del responsable del malformado.
Confeccionado_por	National Varchar	Se registra el nombre del confeccionador de la planilla.

Capítulo 2

Nombre: mc_hermano		
Descripción: Registra los datos de los hermanos del paciente.		
Atributo	Tipo	Descripción
Id_hermano	int	Identificador de la tabla hermano
Id_persona	int	Identificador de la persona
Id_sexo	char	Se registra el sexo del hermano.
Fecha_nacimiento	date	Especifica la fecha de nacimiento
estado_actual	char	Se registra el estado actual del hermano si está vivo o muerto
Id_prod	char	Se registra el estado en que nació el hermano si vivo o muerto.

Nombre: mc_paciente_malformado		
Descripción: Registro de los datos específicos del paciente malformado.		
Atributo	Tipo	Descripción
Id_persona	int	Identificador de la persona
descripcion	text	Se registra la descripción de la malformación que presenta el paciente.

Capítulo 2

Nombre: mc_control		
Descripción: Registra los datos específicos del malformado control		
Atributo	Tipo	Descripción
Id_persona	int	Identificador de la persona
siguiente	Tinyint	Se registra si el control es nacido siguiente o no siguiente del paciente con malformación.

Nombre: mc_gemelo_control		
Descripción: Registro de los datos específicos del control gemelo		
Atributo	Tipo	Descripción
Id_persona	int	Identificador de la persona
orden	int	Se registra el orden del nacimiento de los gemelos.
monovitelino	Tinyint	Se registra se el gemelo es monovitelino y bivitelino.

Nombre: mc_factore_ambientales		
Descripción: Contiene los datos de los factores ambientales de la madre embarazada		
Atributo	Tipo	Descripción
id_factore_ambientales	int	Identificador de los factores ambientales
oligoamnios	tinyint	Si tuvo oligoamnios

Capítulo 2

trim_parto_pret	int	Especifica el trimestre del parto
parto_pret	varchar	Especifica el parto
droga	varchar	Especifica el tipo de droga
trim_droga	int	Especifica el trimestre que tomo droga
polihidramnio	tinyint	Si tuvo polihidramnio
trim_cant_cigar_dia	int	Especifica en el trimestre que se tomo la cantidad de cigarro en el día
cant_cigarros_dia	int	Especifica la cantidad de cigarro en el día
trim_dossi_alcohol	int	Especifica en el trimestre que tomo la dosis de alcohol
dosis_alcohol	float	Especifica la dosis de alcohol que tomo
trim_anestesia	int	Especifica en el trimestre de la causa de la anestesia
causa_anestesia	varchar	Especifica la causa de la anestesia
trim_hipertemia	int	Especifica en el trimestre de la causa de la hipertermia
causa_hipertemia	varchar	Especifica la causa de la hipertermia

Nombre: mc_factores_geneticos		
Descripción: Contiene los datos de los factores genéticos de la madre embarazada		
Atributo	Tipo	Descripción
id_factores_geneticos	int	Identificador de los factores genéticos

Capítulo 2

num_aborto	int	Especifica el numero de abortos
malformados_familia	tinyint	Especifica si algún familiar malformado
consanguineo_paternal	varchar	Especifica la consanguineidad paternal

Módulo Tele - consulta.

Nombre: tl_solicitud		
Descripción: Almacena los datos de la solicitud de una tele consulta		
Atributo	Tipo	Descripción
id_solicitud	int	Identificador de la solicitud
id_estado	int	Identificador de la tabla tl_estado (llave foránea)
id_num_hc	varchar	Identificador de la tabla hc_historia_clinica (llave foránea)
id_medico	int	Identificador de la tabla dg_medico llave foránea)
fecha_solicitud	date	Especifica la fecha en que el médico realiza y envía la solicitud.
urgencia	varchar	Especifica el tipo de urgencia que tiene la solicitud
fundamentacion	text	Especifica el resumen de historia clínica con los hallazgos principales que justifiquen la solicitud de la discusión de un caso
motivo	text	Principal signo o preocupación que genera la solicitud de la discusión de un caso

Capítulo 2

foto_caso	image	Foto del caso que será debatido.
discusion	tinyint	Se activa en 1 cuando el caso esté en el momento de la discusión, es decir del chat.
aprobada	tinyint	Se activa en 1 si la solicitud es aprobada por el administrador para llevar a cabo la discusión.
create_at	date	Para la fecha y hora en que se creó algo en la base de datos.
update_at	date	Para la fecha y hora en que se actualizó algo en la base de datos.

Nombre: tl_sala_conferencia		
Descripción: Contiene lo referente a los datos de las tele consultas creadas		
Atributo	Tipo	Descripción
id_sala_conferencia	int	Identificador de la sala que se planifica.
id_solicitud	int	Identificador de la tabla tl_solicitud (llave foránea)
fecha_sala	date	Fecha en que se realizará la discusión del caso.
hora_inicio_sala	date	Hora en que comenzará la discusión del caso.
hora_fin_sala	date	Hora en que terminará la discusión del caso.
create_at	date	Para la fecha y hora en que se creó algo en la base de datos
update_at	date	Para la fecha y hora en que se actualizó algo en la

Capítulo 2

		base de datos.
--	--	----------------

Nombre: tl_participante_sala		
Descripción: Almacena los usuarios que van a participar en la tele consulta		
Atributo	Tipo	Descripción
id_sala_conferencia	int	Identificador de la tabla tl_sala_conferencia (llave foránea)
id_participante_sala	int	Identificador que se le asigna a cada participante.
id_medico	int	Identificador de la tabla dg_medico (llave foránea)
es_moderador	tinyint	Se activa en 1 para el médico que se desempeñará como moderador del debate
es_solicitante	tinyint	Se activa en 1 para el médico que hizo la solicitud del caso a discutir.
Create_at	date	Para la fecha y hora en que se creó algo en la base de datos

Nombre: tl_informe		
Descripción: Contiene lo referente a los datos de los informes elaborados a partir del historial de las tele-consultas planificadas		
Atributo	Tipo	Descripción
id_informe	int	Identificador del informe de un caso discutido.

Capítulo 2

id_solicitud	int	Identificador de la tabla tl_ solicitud (llave foránea)
resumen	text	Resumen del debate
diagnostico	text	Diagnóstico del caso discutido.
recomendaciones	text	Recomendaciones sobre el diagnóstico.
diagnostico_definitivo	Tinyint-	Se activa en 1 si el diagnóstico arrojado durante el debate es definitivo.
create_at	date	Para la fecha y hora en que se creó algo en la base de datos
aprobado	tinyint	Especifica si esta aprobado
update_at	date	Para la fecha y hora en que se actualizó algo en la base de datos.

Módulo RECUREM.

Nombre: rm_datos_clinico		
Descripción: Contiene los datos clínicos del retrasado mental		
Atributo	Tipo	Descripción
ID_Datos_Clinico	int	Identificador de datos clínicos
Tiene_Diagn_Causa_RM	int	Especifica si tiene diagnóstico de la causa del retrasado mental
Nombre_Pers_Datos	Varchar	Especifica el nombre de la persona que brinda los datos

Capítulo 2

id_paciente	int	Identificador de la tabla dg_paciente (llave foránea)
Evidencia_dirmorfica	tinyint	Especifica si tiene evidencia dismórfica
ID_Madre_Emb	int	Identificador de la tabla rm_madre_emb (llave foránea)
CMF	Varchar	Especifica el nombre del consultorio médico de la familia
id_Parentesco	int	Identificador de la tabla rm_parentesco (llave foránea)
Tiene_otros_serv_genetica	tinyint	Especifica si existen otros servicios

Nombre: rm_datos_del_parto		
Descripción: Contiene los datos referente al parto		
Atributo	Tipo	Descripción
id_datos_del_parto	int	Identificador del parto
ID_Madre_Emb	int	Identificador de la tabla rm_madre_emb (llave foránea)
edad_gestac_parto	varchar	Especifica la edad de la madre en el período del embarazo.
lugar_parto	varchar	Especifica el lugar del parto
dias_nacido_al_alta	int	Especifica los días de nacidos hasta el alta.
tipo_nacimiento	varchar	Especifica el tipo de nacimiento (cesárea o parto)

Capítulo 2

		normal).
--	--	----------

Nombre: rm_examen_fisico		
Descripción: Almacena la información referente al examen físico		
Atributo	Tipo	Descripción
ID_Examen_Fisico	int	Identificador del examen físico
ID_Datos_Clinico	int	Identificador de la tabla rm_datos_clinico (llave foránea)
Otras_malfor_intr	tinyint	Especifica si tiene otras malformaciones internas
Otras_psicois_primaria	tinyint	Especifica si tiene otras psicosis primarias

Módulo RECUEGEN.

Nombre: eg_enfermedades_geneticas		
Descripción: Almacena la información referente a las enfermedades genéticas		
Atributo	Tipo	Descripción
id_codigo	int	Identificador de la enfermedad
Id_clasificacion_enfer_genetica	int	Identificador de la tabla eg_clasificacion_enfer_genetica (llave foránea)
nombre_enfermedad	varchar	Especifica el nombre de la enfermedad

Capítulo 2

Nombre: eg_clasificacion_enfer_geneticas		
Descripción: Almacena la información referente a la clasificación de las enfermedades genéticas		
Atributo	Tipo	Descripción
id_clasificacion_enfer_geneticas	int	Identificador
clasificacion	Varchar	Clasificación de la enfermedad

Módulo RECUGEM.

Nombre: gm_gemelo		
Descripción: Recoge los datos generales de cada gemelo		
Atributo	Tipo	Descripción
cod_gemelo	Varchar	Llave primaria
idpareja	Varchar	Llave primaria
id_paciente	Int	Campo llave extranjera
Id_indocumentado	Int	Campo llave extranjera
centro_trabajo	Varchar	El centro de trabajo
adoptado	Tinyint	Si el gemelo es adoptado o no
fotoFrontal	LongBlob	Foto donde aparecen ambos de frente
fotoLateral	LongBlob	Foto donde aparecen ambos de lado

Capítulo 2

telefono	Varchar	Teléfono de donde vive
Id_nivel_escolar	int	Campo llave extranjera
cant_estudios	int	Para cuantos estudios ha sido el gemelo

Nombre: gm_gemelo_conv_familiar		
Descripción: Almacena información sobre la convivencia familiar.		
Atributo	Tipo	Descripción
idpareja	Varchar	Campo llave primaria
cod_gemelo	Varchar	Campo llave primaria
cfRelacFAMIL	Varchar	Cómo son las relaciones entre los familiares en la casa
cfFuman	Tinyint	Si fuman los familiares
cfBeben	Tinyint	Si Beben los familiares
cvAntisociales	Tinyint	Si hay antisociales
cvPresoPenal	Tinyint	Si hay sanciones penales

Nombre: gm_gemelo_fact_riesgo		
Descripción: Aquí es donde se recogen las preguntas sobre factores de riesgo del gemelo		
Atributo	Tipo	Descripción

Capítulo 2

idpareja	Varchar	Llave primaria
cod_gemelo	Varchar	Llave primaria
frFuma	TinyInt	Si fuma
frBebe	TinyInt	Si bebe
frObesidad	TinyInt	Si es obeso

Nombre: gm_gemelo_padecimientos		
Descripción: Recoge los datos de los padecimientos de los gemelos		
Atributo	Tipo	Descripción
Idpareja	Varchar	Campo llave primaria
cod_gemelo	Varchar	Campo llave primaria
padHipertension	TinyInt	Si padece Hipertensión
padCardioplsquemica	TinyInt	Si padece Cardiopatía Isquémica
padEpilepsia	TinyInt	Si padece Epilepsia
padRetrMental	TinyInt	Si padece Retraso Mental
padGradoRetrMental	Varchar	el Grado de Retraso Mental que padece
padCancerMama	TinyInt	Si padece Cáncer de Mama
padCancerColon	TinyInt	Si padece Cáncer de Colon
padCancerProstata	TinyInt	Si padece Cáncer de Próstata

Capítulo 2

padDiabetesMell	TinyInt	Si padece Diabetes
padAsmaBronq	TinyInt	Si padece Asma Bronquial
padEsquizofrenia	TinyInt	Si padece Esquizofrenia
padPsicosisManiacDepr	TinyInt	Si padece Psicosis Maniaco Depresiva
padAdiccAlcohol	TinyInt	Si padece Adicción al Alcohol
padParkinson	TinyInt	Si padece Parkinson
padDemencia	TinyInt	Si padece Demencia
padDepresion	TinyInt	Si padece Depresión
padOtra1	Varchar	Si padece de otra enfermedad
padOtra2	Varchar	Si padece de otra enfermedad
padOtra3	Varchar	Si padece de otra enfermedad
padOtra4	Varchar	Si padece de otra enfermedad

Nombre: gm_gemelo_exploCancer		
Descripción: Almacena la información sobre cáncer que ha padecido los gemelos		
Atributo	Tipo	Descripción
ldpareja	Varchar	Campo llave primaria
cod_gemelo	Varchar	Campo llave primaria
cancColon	TinyInt	Si ha padecido cáncer de Colon

Capítulo 2

cancMama	TinyInt	Si ha padecido cáncer de Mama
cancPulmon	TinyInt	Si ha padecido cáncer de Pulmón
cancProstata	TinyInt	Si ha padecido cáncer de Próstata
cancVejiga	TinyInt	Si ha padecido cáncer de Vejiga
cancLinfoma	TinyInt	Si ha padecido cáncer de Linfoma
cancUtero	TinyInt	Si ha padecido cáncer de Útero
cancLeucemia	TinyInt	Si ha padecido cáncer de Leucemia
cancLaringe	TinyInt	Si ha padecido cáncer de Laringe
cancOtro	TinyInt	Si ha padecido otro cáncer
cancOtroCual	Varchar	Descripción del otro cáncer
cancEdadDiagAnos	Int	Años del diagnóstico
cancEdadDiagMeses	Int	Meses del diagnóstico
cancTipoPadFam	Varchar	Tipo de cáncer padecido en algún antecedente familiar

Nombre: gm_gemelo_exploDepresion

Descripción: Información sobre las preguntas exploratorias para la depresión

Atributo	Tipo	Descripción
idpareja	Varchar	Campo llave primaria

Capítulo 2

cod_gemelo	Varchar	Campo llave primaria
depInfeliz	Varchar	Si se siente Infeliz
depCansado	Varchar	Si se siente Cansado
depInquieto	Varchar	Si se siente Inquieto
depInsomnio	Varchar	Si tiene Insomnio
depDisfrute	Varchar	Si no disfruta ninguna actividad
depSexo	Varchar	Si perdió interés en el sexo
depDecisiones	Varchar	Si tiene dificultad en tomar decisiones
depFracasado	Varchar	Si se siente fracasado
depCulpable	Varchar	Si se siente culpable por algo
depCosasMal	Varchar	Siente que las cosas siempre irán mal

Nombre: gm_gemelo_exploCerebro_Vascular		
Descripción: Aquí es donde se recogen las preguntas exploratorias de cerebro Vascular		
Atributo	Tipo	Descripción
ldpareja	Varchar	Llave primaria
cod_gemelo	Varchar	Llave primaria
cvAdormecimiento	TinyInt	Ha presentado alguna vez, de forma repentina, adormecimiento o debilidad de un lado del cuerpo o cara

Capítulo 2

cvDificHabla	TinyInt	Ha presentado alguna vez, de forma repentina dificultad para hablar o hacerse entender
cvPerdVision	TinyInt	Ha presentado alguna vez, de forma repentina pérdida de la visión de uno o ambos ojos
cvEquilibrio	TinyInt	Ha presentado alguna vez, de forma repentina trastorno para la coordinación y el equilibrio
cvDiagnosticado	Varchar(10)	El médico le ha diagnosticado un accidente cerebro vascular

Nombre: gm_gemelo_edad_escolar		
Descripción: Aquí es donde se recogen las preguntas sobre Edad Escolar		
Atributo	Tipo	Descripción
ldpareja	Varchar	Llave primaria
cod_gemelo	Varchar	Llave primaria
ld_tipo_escuela	Int	Llave extranjera
nombMaestro	Varchar	Nombre del maestro
nombre_escuela	Varchar	Nombre de la escuela
relacFamEsc	Varchar	Relaciones familiares con la escuela
controlComport	Varchar	Como se controla el comportamiento del gemelo en el hogar
caPegaInsulta	Varchar	Si le pega, insulta a los demás niños

Capítulo 2

caRompeAdrede	Varchar	Si rompe cosas adrede
caCruelAnimales	Varchar	Si es cruel con los animales
cafacilEnojo	Varchar	Si se enoja fácilmente
caGritaxTodo	Varchar	Si le grita a todos
caJuegosMov	Varchar	Si le gusta jugar juegos en que esté en movimiento
caJuegosMesa	Varchar	Si le gustan los juegos de mesa
caMuyInquieto	Varchar	Si es muy inquieto
caTranquiloTc	Varchar	Si esta tranquilo mientras ve la tele clase
caCorreReceso	Varchar	Si corre sin parar a la hora del receso
caJuegaSolo	Varchar	Si es capaz de jugar juega solo por un rato
caRuido	Varchar	Hace ruido donde quiera que se encuentre
caBrincaSentado	Varchar	Brinca y salta mientras está sentado
caMania	Varchar	Tiene algún tic o manía
caTorpeMov	Varchar	Es un niño torpe en sus movimientos
caTermTareas	Varchar	Deja por terminar las tareas que comienza
caDificAtender	Varchar	Tiene dificultades para atender a un cuento o narración
caConcentrado	Varchar	Se concentra en las actividades que realiza
caComoAusente	Varchar	Parece ausente por momentos

Capítulo 2

caPierdeCosas	Varchar	Pierde o deja las cosas olvidadas
caTareaBien	Varchar	Realiza las tareas correctamente
caDificEntTareas	Varchar	Tiene dificultades para comprender las tareas
caDemoraTarea	Varchar	Demora mucho tiempo en la ejecución de las tareas
caCambiaActiv	Varchar	Cambia constantemente de actividad
caDesobediente	Varchar	Es un niño desobediente
caEmociones	Varchar	No expresa sentimientos o emociones
caAyuda	Varchar	Ayuda cuando alguien está herido, se siente mal o se siente enfermo
caSentimCulpa	Varchar	Se siente mal o culpable cuando él o ella hace algo erróneo
caAmigoCercano	Varchar	Tiene al menos un amigo cercano
caConsiderado	Varchar	Es considerado ante los sentimientos de otra persona
caAmable	Varchar	Es amable con los niños más pequeños
caPreocupado	Varchar	Es preocupado por sus deberes escolares
caMiente	Varchar	menudo miente o hace trampas
caroba	Varchar	Roba en la casa, escuela o cualquier otro lugar
caAgresivo	Varchar	Si le fastidian responde agresivamente

Capítulo 2

caAcusa	Varchar	Culpa a los demás de haber comenzado los problemas
caSobreReacciona	Varchar	Sobre reacciona con disgusto a los accidentes
caPandillero	Varchar	Logra que otros “le caigan en pandilla” para agredir a un tercero
caIntimida	Varchar	Amenaza o intimida
caUsaFuerza	Varchar	Usa la fuerza física para intimidar a otros
lecLento	TinyInt	Lee lento y con errores frecuentes
lecConfLetras	TinyInt	Cuando lee confunde las letras parecidas
lecPalabDific	TinyInt	Lee correctamente palabras desconocidas o difíciles
lecUsaDedos	TinyInt	Usa los dedos para seguir un texto
lecRapidAdiv	TinyInt	Lee muy rápido y adivinando las palabras
lecComprende	TinyInt	Comprende lo que lee sin dificultad
escLento	TinyInt	Escribe lento
escLegible	TinyInt	Tiene escritura legible
escPocoErr	TinyInt	Escribe con pocos errores ortográficos
escUneSepPalab	TinyInt	Une palabras que se escriben separadas y separa palabras que se escriben unidas
escConfLetrParec	TinyInt	Al escribir confunde letras parecidas como p, b, d, q

Capítulo 2

escBuenaRedac	TinyInt	Tiene buena redacción de textos (se ajusta al tema, mantiene un orden lógico, las ideas tienen calidad)
matNumMal	TinyInt	Escribe números al revés o rotados
matConfAntSuc	TinyInt	Conoce el antecesor y sucesor de cualquier número
matConfDigSim	TinyInt	Confunde dígitos similares (por ej. 6/9, 3/8)
matMemMultip	TinyInt	Tiene buena memoria para las tablas de multiplicar
matCifrasCorrect	TinyInt	Dice y escribe correctamente las cifras
matConfSignosOp	TinyInt	Al calcular confunde los signos operativos
matDifAprendProc	TinyInt	Tiene dificultad en aprender los procedimientos para multiplicar, dividir y/o reducir números fraccionarios
matDifTermConc	TinyInt	Tiene problemas para denominar los términos, las operaciones y/o los conceptos matemáticos

Nombre: gm_gemelo_edad_laboral		
Descripción: Esta tabla recoge algunos datos de los gemelos en edad laboral		
Atributo	Tipo	Descripción
idpareja	Varchar	Llave primaria
cod_gemelo	Varchar	Llave primaria

Capítulo 2

centro_trabajo	Varchar	Especifica el nombre del centro de trabajo
----------------	---------	--------------------------------------------

Nombre: gm_pareja		
Descripción: La tabla recoge la información sobre las parejas de gemelos.		
Atributo	Tipo	Descripción
ldpareja	varchar	Llave primaria
lugar_nacimiento	Varchar	El lugar donde nacieron
nombre_madre	Varchar	El nombre y apellidos de la madre
edad_madre	Int	La edad de la madre
nombre_padre	Varchar	El nombre y apellidos del padre
edad_padre	Int	La edad de la padre
separadosDesde	Int	Desde que edad están separados
separadosHasta	Int	Hasta cuando han estado separados
otroGemFamilia	TinyInt	Si existen otros gemelos en la familia
nombTutor	Varchar	Nombre y apellidos del tutor
dirTutor	Varchar	Dirección del tutor
telefTutor	Varchar	Teléfono del tutor
cant_estudios	Int	Para cuantos estudios ha sido utilizada la pareja
ld_consejo	varchar	Llave extranjera

Capítulo 2

Nombre: gm_pareja_cigosidad		
Descripción: Recoge los datos sobre la cigosidad de los gemelos es decir si es monocigótico o dicigótico.		
Atributo	Tipo	Descripción
ldpareja	varchar	Llave primaria
cigosidad	Varchar	Tipo de cigosidad según el algoritmo de cigosidad
caraFrente	Varchar	Parecido en cuanto a cara de frente
caraPerfil	Varchar	Parecido en cuanto a cara de perfil
formaMenton	Varchar	Parecido en cuanto a forma del mentón
colorOjos	Varchar	Parecido en cuanto a color de los ojos
bocaLabios	Varchar	Parecido en cuanto a la boca y los labios
colorFormaPelo	Varchar	Parecido en cuanto a color, forma y pelo
complexCorporal	Varchar	Parecido en cuanto a complemento corporal
dosGotasAgua	TinyInt	Si son exactamente iguales
unaPlacenta	Varchar	Si son nacidos de la misma placenta
igcolorNatCabello	TinyInt	Si tiene el mismo color natural de cabello
igTipoCabello	TinyInt	Si tiene el mismo tipo de cabello
igFormaOjos	TinyInt	Si tiene la misma forma de los ojos
igcolorOjos	TinyInt	Si tienen el mismo color de los ojos

Capítulo 2

madreConf	Varchar	Si la madre los confunde
padreConf	Varchar	Si el padre los confunde
miemFamConf	Varchar	Si los miembros de la familia los confunde
madreConfLact	Varchar	Si la madre los confunde durante la lactancia
ajenosConf	Varchar	Si personas ajenas a la familia los confunde
madreConf1Anno	Varchar	Si la madre los confunde a 1 año de edad
ajenosConfSepar	Varchar	Si las personas ajenas los confunden cuando están separados
madreConfDesp1Anno	Varchar	Si la madre los confunde después de 1 año de edad
miemFamConfJuntos	Varchar	Si miembros de la familia los confunde cuando están juntos
madreConfSepar	Varchar	Si la madre los confunde cuando están separados
miemFamConfSepar	Varchar	Si miembros de la familia los confunde cuando están separados

Nombre: gm_pareja_gem_parientes_igualsexo

Descripción: Aquí es donde se recogen los datos sobre los gemelos que tienen parientes gemelos de igual sexo.

Atributo	Tipo	Descripción
----------	------	-------------

Capítulo 2

idpareja	varchar	Llave primaria
hermanos	TinyInt	Si tiene un hermano gemelo de igual sexo
hijos	TinyInt	Si tiene hijos gemelos de igual sexo
madre	TinyInt	Si la madre es gemela de igual sexo
padre	TinyInt	Si el padre es gemelo de igual sexo
tiosMater	TinyInt	Si tiene tíos maternos gemelos de igual sexo
tiosPater	TinyInt	Si tiene tíos paternos gemelos de igual sexo
sobrinos	TinyInt	Si tiene sobrinos gemelos de igual sexo

Nombre: gm_ant_patol_familiar_pareja		
Descripción: Recoge los antecedentes familiares de la pareja		
Atributo	Tipo	Descripción
idpareja	varchar	Campo llave primaria
idnomfam	int	Campo llave primaria (identificador del familiar)
hta	TinyInt	Si padece Hipertensión
cardiopatía	TinyInt	Si padece cardiopatía
asma	TinyInt	Si padece asma
demencia	TinyInt	Si padece demencia
cancer	TinyInt	Si padece cáncer

Capítulo 2

tmHospitalizado	TinyInt	Si a estado hospitalizado por trastornos mentales
tmGentePersigue	TinyInt	Si cree que la gente lo persigue
tmVocesVisiones	TinyInt	Si oye voces o ve visiones
tmSensacExtranas	TinyInt	Si tiene sensaciones extrañas
tmContactRealidad	TinyInt	Si ha perdido contacto con la realidad

Nombre: gm_gemelo_antPerinatales		
Descripción: Almacena los datos de los antecedentes perinatal del gemelo		
Atributo	Tipo	Descripción
cod_gemelo	Varchar	Campo llave primaria
ldpareja	Varchar	Campo llave primaria
appeso	Float	Peso con que nació
aphipoxia	TinyInt	Si tuvo hipoxia al nacer
apconvulsiones	TinyInt	Si tuvo convulsiones al nacer
apinfecciones	TinyInt	Si tuvo infecciones al nacer
aptrauma	TinyInt	Si tuvo traumas al nacer
apotros	Varchar	Si tuvo otros al nacer
Apparto_cesarea	TinyInt	Especifica si el parto fue cesárea
apamenaza_aborto	TinyInt	Si fue amenaza de aborto

Capítulo 2

apcantidad_semanas_parto	Int	Cantidad de semanas de parto
apcomplicacion_parto	TinyInt	Si tuvo complicaciones en el parto
apcesarea_urgencia	Varchar	En caso de cesárea, si esta fue por urgencia
Apparto_normal	TinyInt	Especifica si el parto fue normal

Nombre: gm_indocumentado

Descripción: En esta tabla se van a recoger los datos primarios de los pacientes que tengan problema en la documentación, o sea los mismos atributos de la tabla dg_pacientes.

Conclusiones

En este capítulo se ha descrito en detalle la estrategia de integración de los módulos del SIGM, así como la arquitectura en la que se desplegará la aplicación. Además se ha mostrado el diagrama entidad-relación, así como la descripción de cada una de las tablas que lo forman, para un mejor entendimiento del mismo y de cada uno de sus atributos y tipos de atributos.

Capítulo 3: Validación del diseño realizado.

En este capítulo se ofrece información sobre la validación del diseño descrito en el capítulo 2, detallando las reglas a tener en cuenta para un buen diseño, además de la descripción de una validación funcional a través de la generación de códigos para un llenado voluminoso e inteligente de la base de datos.

3.1. Validación teórica del diseño.

Para lograr un buen diseño de una base de datos hay que tener en cuenta un número de aspectos que son importantes como: integridad de los datos, normalización de la base de datos, análisis de la redundancia de la información, trazabilidad de las acciones, así como la seguridad de la base de datos, lo cual es de gran importancia para lograr bajo toda circunstancia que los recursos del sistema sean utilizados para los fines previstos.

3.1.1. Integridad.

La integridad de los datos se refiere a la corrección y completitud de los datos en una base de datos. Cuando los contenidos de una base de datos son modificados con sentencias INSERT, UPDATE o DELETE, la integridad de los datos puede perderse al añadirse datos no válidos a la base de datos, al modificarse datos existentes tomando un valor incorrecto o al eliminar datos que violan alguna regla. Para garantizar la integridad de los datos en bases de datos relacionales, existen varias restricciones como: integridad de dominio, de entidad, referencial, entre otras.

Integridad de dominio: se restringen los valores que puede tomar un atributo respecto a su dominio, se logra controlar los valores que se guardan en la base de datos. En el caso de la base de datos del SIGM esta integridad se garantiza a nivel de aplicación, o sea los programadores son los que velan que ello se cumpla.

Integridad de entidad: se establece que la llave primaria de una tabla debe tener un valor único para cada fila de la tabla, siendo esto un ejemplo de las buenas prácticas, o sea la determinación del atributo que será la llave primaria partiendo de la obligación de que todos los valores del atributo son únicos y no nulos.

Capítulo 3

Chequeo de validez: cuando se crea una tabla cada columna tiene un tipo de datos y el SGBD garantiza que solamente los datos del tipo especificado sean ingresados en la tabla. Por ejemplo el caso de la edad que solo se permite que sea un entero, o sea de tipo integer.

Integridad referencial: es una propiedad deseable en toda base de datos, ya que asegura la integridad entre las claves ajenas y primarias. Por ejemplo en la base de datos del SIGM, con las entidades *Historia Clínica* y *Paciente*. Toda *Historia Clínica* corresponde a un paciente y solamente uno. Esto implica que en todo momento dichos datos sean correctos, sin repeticiones innecesarias, datos perdidos y relaciones mal resueltas. Un paciente se identifica por su atributo *id_paciente*. También tendrá otros atributos como el nombre, la dirección, y otros. La entidad *Historia Clínica* debe tener un atributo *num_hc* que identifique a la *Historia Clínica* y entonces tendría *Historia Clínica* el atributo *id_paciente* como llave foránea. Por sentido común es evidente que todo valor de *num_hc* debe corresponder con algún valor existente del atributo *id_paciente* de la entidad *Paciente*. Cuando se crea una nueva instancia de *Historia Clínica*, la integridad referencial exige que la nueva fila creada posea en su campo *id_paciente* un valor asignado que ya existe en la entidad *Paciente*. En caso contrario, no se permite la operación. Cuando se intenta eliminar una instancia de *Paciente*, la integridad referencial exige que no exista ninguna *Historia Clínica* asociada, es decir, se comprueba que no existe ninguna instancia de *Historia Clínica* cuyo atributo *id_paciente* coincida con el atributo *id_paciente* de la instancia a borrar. En caso contrario, no se permite la operación. Hay que decir que MySQL garantiza que se cumpla la integridad referencial, haciendo uso de disparadores (Triggers) para el tratamiento de errores que permite validar la existencia previa de las llaves foráneas e insertar información válida en la base de datos.

Datos requeridos: establece que una columna tenga un valor no *NULL*. Se define efectuando la declaración de una columna es *NOT NULL* cuando la tabla que contiene la columna se crea por primera vez. Por ejemplo en la tabla *dg_paciente* el nombre del paciente es un dato requerido y se declara *NOT NULL*, así como otros datos de esa y otras tablas, validando que no existan tuplas con esas columnas vacías, o sea *NULL*.

3.1.2. Normalización de la base de datos.

La normalización es un conjunto de reglas que sirven para ayudar a los diseñadores a desarrollar un esquema que minimice los problemas de lógica. Las guías que la normalización provee crean el marco de

referencia para simplificar una estructura de datos compleja. Otra ventaja de la normalización de base de datos es el consumo de espacio [17]. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco. Existen varios niveles de normalización como:

Primera Forma Normal (1NF).

Segunda Forma Normal (2NF).

Tercera Forma Normal (3NF).

Forma Normal Boyce-Codd.

Cuarta Forma Normal.

Quinta Forma Normal o Forma Normal de Proyección-Unión.

Forma Normal de Proyección-Unión Fuerte.

Forma Normal de Proyección-Unión Extra Fuerte.

Forma Normal de Clave de Dominio.

Cada regla está basada en la que le antecede. A continuación se explicará solo las tres primeras formas normales ya que es el nivel al que se ha normalizado la base de datos del SIGM.

Primera Forma Normal (1FN).

Definición formal: una relación R se encuentra en 1FN si y solo si por cada renglón columna contiene valores atómicos [18].

O sea se considera que está en la primera forma normal cuando cumple lo siguiente:

Las celdas de las tablas poseen valores simples y no se permiten grupos ni arreglos repetidos como valores, es decir, contienen un solo valor por cada celda.

Todos los ingresos en cualquier columna (atributo) deben ser del mismo tipo.

Capítulo 3

Cada columna debe tener un nombre único, el orden de las columnas en la tabla no es importante.

Dos filas o renglones de una misma tabla no deben ser idénticas, aunque el orden de las filas no es importante.

Segunda Forma Normal (2FN).

Definición Formal: Una relación R está en 2FN si y solo si está en 1FN y los atributos no primos dependen funcionalmente de la llave primaria [18].

O sea se considera que está en segunda forma normal cuando cumple con las reglas de la primera forma normal y todos sus atributos que no son claves (llaves) dependen por completo de la clave. De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, esta en segunda forma normal.

Tercera Forma Normal (3FN).

Definición Formal: Una relación R está en 3FN si y solo si esta en 2FN y todos sus atributos no primos dependen no transitivamente de la llave primaria [18].

Consiste en eliminar la dependencia transitiva que queda en una segunda forma normal, en pocas palabras una relación esta en tercera forma normal si está en segunda forma normal y no existen dependencias transitivas entre los atributos, nos referimos a dependencias transitivas cuando existe más de una forma de llegar a referencias a un atributo de una relación.

La base de datos del SIGM está normalizada hasta la tercera forma normal, ya que, estos son los tres niveles básicos en que debe estar una base de datos, teniendo en cuenta que cuando estas son muy complejas se hace difícil llevarla a otros niveles de normalización. Dicha base de datos está en 3FN, pues cumple con los requerimientos para la 1FN y la 2FN, y además se garantiza que no existen dependencias transitivas entre los atributos. Con el objetivo de lograr un mayor rendimiento a nivel de aplicación y de optimizar el tiempo de ejecución de las consultas, algunas tablas están normalizadas solamente hasta la 1FN y 2FN; como por ejemplo: la tabla *dg_familiar* que tiene una columna multivaluada (*dirección*), por ello esta tabla no esta normalizada; otro ejemplo es: la tabla *gm_nfamiliar* que se encuentra en 2FN pues presenta dependencia transitiva entre sus atributos, *grado* y *descripción*.

3.1.3. Análisis de la redundancia de la información.

La redundancia de los datos es un factor negativo en las bases de datos, pues trae inconsistencia a los datos y supone un consumo de recursos de almacenamiento que puede llevar a situaciones en las que un dato se actualice en una de sus ubicaciones y en otra no y se pierda la integridad de la BD, por tanto debe evitarse. Aunque en bases de datos grandes es casi imposible eliminar la redundancia al 100%, a veces hay que emplearla por cuestiones de rendimiento. Hay que decir que antes con el almacenamiento de la información en bases de datos separadas había un cúmulo de información repetida innecesariamente.

Eliminar la redundancia de la información es uno de los objetivos principales de la normalización, lo cual no es más que eliminar, o por lo menos tener una mínima redundancia de información en la BD. En el caso de la base de datos del SIGM este parámetro se minimizó en gran medida al normalizar la base de datos hasta la 3FN. Aunque en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias. En ocasiones resulta mejor perder en redundancia por ganar en simplicidad de las consultas y lograr un mejor tiempo de respuesta; tal es el caso de *dg_familiar* donde el campo *color_piel* aparece como texto plano y no como un nomenclador. De igual forma en las tablas *gm_pareja* y *gm_entrevista* existen los campos *dir_tutor* y *dir_entrevistado* respectivamente; en ambos casos estos campos almacenan una dirección como texto plano y no como relaciones a las tablas que gestionan la dirección ya diseñadas, pues el número de JOIN aumentaría considerablemente para cada consulta, y con ellos el rendimiento de la base de datos, teniendo en cuenta que estas tablas son utilizadas frecuentemente.

3.1.4. Seguridad de la base de datos del SIGM.

La seguridad de los datos se refiere a la protección de estos contra el acceso por parte de las personas no autorizadas y contra su indebida destrucción o alteración. Los métodos de seguridad en una base de datos se enfocan hacia la prevención del acceso a la base de datos de usuarios no autorizados. Debido a que las características de los SGBD que facilitan el acceso y la manipulación de las bases de datos también abren sus puertas a intrusos, la mayoría de los SGBD incluyen herramientas de seguridad que permiten el acceso a los datos únicamente a los programas o a las personas autorizadas, y además, restringen los tipos de procesamiento que se puedan relacionar una vez que se tenga el acceso.

Capítulo 3

En el caso de la base de datos de datos del SIGM, el sistema debe integrarse como un componente más de SISalud, para lo cual deberá ser registrado por el mismo e interactuar con otros componentes consumiendo sus servicios, o brindando los propios, utilizando para ello el componente de seguridad SAAA. Además el sistema debe tener un mecanismo propio para gestionar la seguridad a través de niveles de acceso a la información. Los permisos al ejecutar cualquier acción debe estar de acuerdo con el nivel jerárquico de acceso que presente el usuario en cada módulo, el cual es definido por los administradores del sistema. Para controlar los accesos no deseados a la base de datos se tienen en cuenta algunos aspectos:

Cualquier acción antes pasan por un filtro de seguridad de ser ejecutadas. Este filtro verifica que el usuario tiene el nivel de acceso y los permisos requeridos para acceder a la acción. Los privilegios en Symfony se componen por dos aspectos fundamentales:

- Las acciones seguras requieren que los usuarios estén autenticados.
- Las credenciales son privilegios de seguridad agrupados bajo un nombre y que permiten organizar la seguridad en grupos.

Para lograr lo anterior se hace necesario crear el archivo “security.yml” en la carpeta “/config” del proyecto o del módulo. En el cual se deben especificar las restricciones que los usuarios deberán cumplir para poder acceder a las acciones.

Por otro lado es importante lograr que MySQL sea seguro para ello se llevó a cabo las siguientes hechas:

Se crean usuarios a los que se asignan permisos en función de su rol y las tareas que deben realizar; por ejemplo para eliminar tablas de la base de datos solo está permitido a los diseñadores y a los del módulo de administración; mientras el resto de los usuarios solo puede operar sobre la información que existe en las tablas, y no sobre las tablas mismas.

3.2. Validación funcional.

3.2.1. Llenado automático de los datos y pruebas de carga a la base datos.

Para un llenado inteligente y voluminoso de la base de datos se utilizó la herramienta EMS Data Generator for MySQL, este llenado se realizó teniendo en cuenta el tipo de datos y el rango de valores de

Capítulo 3

cada atributo. Se llenaron 19 tablas con un total de 5000 registros y no excedió de los 2 minutos; de esas tablas eran 15 nomencladores, y 4 tablas (*dg_pacientes*, *dg_historia_perinatal*, *dis_vivienda* y *dis_discapitados*), las cuales son las tablas que mayor cantidad de columnas presentan en la base de datos. Una vez el llenado de la base de datos se procedió a realizar las pruebas de estrés, el objetivo de estas pruebas es poder simular conexiones de usuarios y observar como se comporta la base de datos bajo cargas extremas, y así solucionar los problemas potenciales de rendimiento que se puedan detectar.

Las pruebas de estrés llevan al sistema a un punto máximo para poder medir sus capacidades y las condiciones en las cuales trabaja realizando una cantidad definida de peticiones y procesos. Para realizar dichas pruebas a la base de datos se utilizó una aplicación implementada en C# que simula un número de conexiones y peticiones a la base de datos, indicado por el usuario. Luego de establecida esta cantidad n de conexiones entonces se procede a la ejecución de consultas para valorar el comportamiento del sistema frente a diferentes números de conexiones.

En las siguientes tablas se muestra el resultado de la prueba de estrés realizada a la base de datos, para ello se ejecutaron tres consultas con una conexión a la base de datos y después se ejecutaron cuando habían 250 conexiones establecidas, realizando peticiones a la base de datos.

Tabla 1. Resultados de las pruebas de estrés. Primera iteración.

Consulta	Conexiones	Tiempo
Consulta 1	1	31 ms
Consulta 2	1	31 ms
Consulta 3	1	47 ms

Tabla 2. Resultado de las pruebas de estrés. Segunda iteración.

Consulta	Conexiones	Tiempo
Consulta 1	250	47 ms
Consulta 2	250	47 ms
Consulta 3	250	62 ms

Capítulo 3

Tomando en consideración que la base de datos fue montada en un servidor con 512MB de memoria RAM y un procesador Intel Pentium IV a 3.00GHz, puede decirse que aunque el tiempo de respuesta a las consultas es menor para un menor número de conexiones y peticiones, en ambos casos se encuentra en correspondencia con las condiciones establecidas para ejecutar las pruebas. Para estas condiciones la base de datos soporta 255 conexiones, siendo esto satisfactorio ya que una vez que el sistema se encuentre desplegado no va a exceder de 200 conexiones y los servidores superan las condiciones descritas anteriormente. Teniendo en cuenta que el tiempo de respuesta aumenta con el número de conexiones se recomienda optimizar los parámetros de MySQL de modo que estos factores influyan lo menos posible en los resultados obtenidos.

Para ver las consultas utilizadas ver el Anexo 7.

Conclusiones

En este capítulo se han analizado una serie de aspectos que se deben tener en cuenta para la realización de un diseño eficiente de una base de datos. Donde se tuvo presente la integridad de los datos y como se garantizaba en la base de datos del SIGM, formas de normalización de la base de datos del SIGM, así como el análisis de la seguridad de la misma. También se describió y se analizó como se llevó a cabo la validación funcional de la base de datos mediante un llenado voluminoso de las tablas, así como las pruebas de funcionalidades de la misma.

CONCLUSIONES

Durante el desarrollo de esta tesis se expuso la necesidad de diseñar una única base de datos para el SIGM, que permita el manejo de la información de una forma eficiente en el CNGM. Mediante este trabajo se dio cumplimiento a los objetivos propuestos:

- Se llevó a cabo un proceso de integración con los siete módulos del SIGM, con el cual se diseñó una base de datos única que cuenta con 256 tablas. También se realizó la descripción de la arquitectura y de todas las tablas que componen la base de datos.
- Se utilizó el EMS Data Generator para el llenado de datos de las tablas de la base de datos, y se logró realizar pruebas.

RECOMENDACIONES

Por las características que presenta este proyecto, aún se le pueden incorporar nuevas prestaciones o incluso mejorar las que ya tiene. Considerando lo anterior se presentan las siguientes recomendaciones:

- 1) Realizar proceso de migración a PostgreSQL el cual es un SGBD libre, ya que MySQL en Cuba ha sido restringido y se debe adquirir licencias para su uso.

Referencias Bibliográficas

REFERENCIAS BIBLIOGRÁFICAS

- 1- Rush University Medical Center, Chicago, Illinois. HEALTH INFORMATION IN SPANISH. Disponible en: <http://www.rush.edu/spanish/speds/genetics/index.html>
- 2- López, Josmar. Concepto de Archivos y Datos. Disponible en: <http://boards5.melodysoft.com/app?ID=GM005N&msg=4>
- 3- Disponible en: <http://www.angelfire.com/planet/tareas/JorgeGuera>
- 4- Baños, Edilberto; Chacón, Jennifer. MONOGRAFIAS.COM. Disponible en: <http://www.monografias.com/trabajos38/tipos-lenguajes-programacion/tipos-lenguajes-programacion2.shtml#refer> .
- 5- Castellanos, Yadira; Verdecia, Teresa; Santiesteban, Isabel. Guía metodológica para realizar una herramienta idónea para indizar las ponencias de la base de datos FORUM Disponible en: <http://dspace.idict.cu/handle/123456789/154>
- 6- Burbano, Diego. Bases de Datos Open Source Vs Bases de Datos Comerciales. Disponible en: <http://www.telefonica.net/web2/todobi/Pixfolder/Comparativa-Bases-de-Datos-Open-Source-Vs-Comerciales.doc>
- 7- Catalá, Andreu. Sistema de agentes portables incrustados para entornos naturales seguros (SAPIENS). Disponible en: <http://www.imsersomayores.csic.es/documentos/documentos/imserso-estudiosidi-43.pdf>
- 8- Glosario de informática. Mayo del 2008. Disponible en: <http://glosario.panamacom.com/letra-o.html>
- 9- Marqués, María. Sistemas de bases de datos. Disponible en: <http://www3.uji.es/~mmarques/f47/apun/node4.html>.
- 10- Buenmaster.com. PostgreSQL, motor de base de datos. Agosto del 2007. Disponible en: <http://www.buenmaster.com/?a=545>
- 11- Garrido, Francisco. SOFTWARE ESPECIALIZADO CON EL QUE CUENTA EL LCI. Disponible en: <http://www.lci.ulsa.mx/software/index.html>

Referencias Bibliográficas

- 12-Jiménez, Darwin. Ciclo de vida del software. Agosto del 2007. Disponible en: <http://darwinjimenezgarzon.blogspot.com/2007/08/ciclo-de-vida-de-software.html>
- 13- Softonic. Crea y gestiona bases de datos ODBC desde una práctica interfaz visual. Disponible en: <http://dbdesigner.softonic.com/>
- 14-Méndez, Rosa. LO NUEVO DE RATIONAL ROSE. Disponible en: <http://www.usmp.edu.pe/publicaciones/boletin/fia/info36/proyectos.html>
- 15- Ayuda extendida del Rational Rose Enterprise Edition. 2003.
- 16- Sepúlveda, Dolores. Guías técnicas aplicables a la contratación de bienes y servicios de tecnologías de la información y las comunicaciones. Disponible en: <http://www.csae.map.es/csi/silice/Sqbd6.html> .
- 17-MySQL Hispano. Normalización de bases de datos. Disponible en: <http://www.mysql-hispano.org/page.php?id=16&pag=2>
- 18- Campoy, Lourdes. Diseño de Bases de Datos relacionales. Disponible en: <http://sistemas.itlp.edu.mx/tutoriales/basedat1/temas4.htm>

BIBLIOGRAFÍA

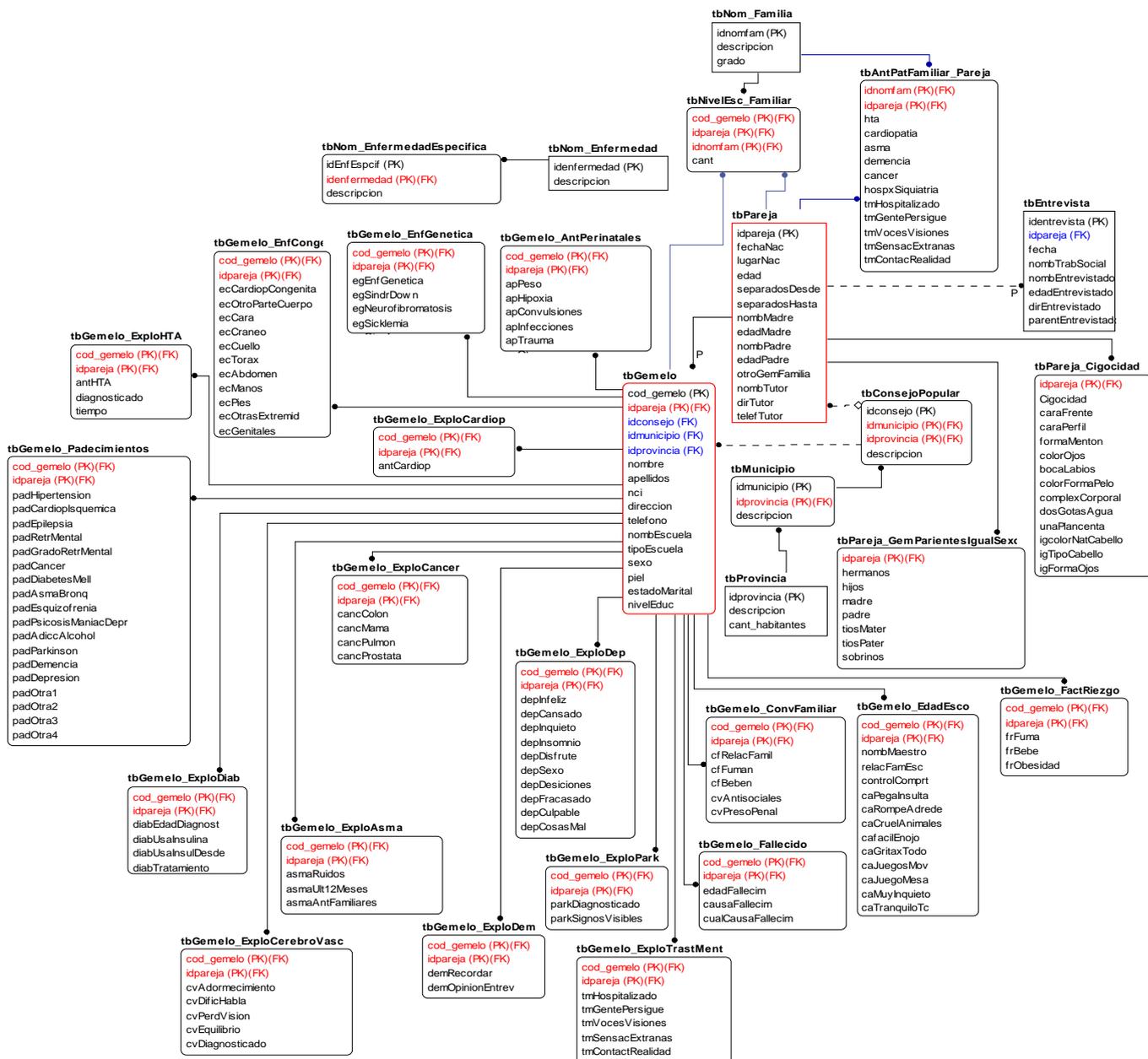
- ACIMED v.11 n.4 Ciudad de La Habana jul.-ago. 2003. Impacto de la Bioinformática en las ciencias biomédicas. Disponible en: <http://eprints.rclis.org/archive/00001767/01/impacto.pdf>
- Stoll, Dr. Mario. Genética Médica Disponible en: <http://www.diariosalud.net/content/blogcategory/109/506/>
- Asignatura: Sistemas de Bases de Datos. Curso: 2005-2006. Conferencia 1.Introducción a las Bases de Datos. Modelo Entidad-Relación (ER). Disponible en: <http://teleformacion.uci.cu/course/view.php?id=45>.
- Rush University Medical Center, Chicago, Illinois. El Proyecto Genoma Humano <http://www.rush.edu/spanish/speds/genetics/genome.html>
- Asignatura: Ingeniería de Software 1 Curso: 2007-2008. Conferencia No.1. Introducción a la Ingeniería de Software. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=6655> .
- TIENDALINUX.COM. Servicios de Soporte. Disponible en: <http://soporte.tiendalinux.com/portal> .
- CA.COM. Press Release. Disponible en: <http://ca.com/es/press/release.aspx?cid=145405>
- TARINGA. DB Designer - Diseño de Base de Datos. Disponible en: <http://www.taringa.net/posts/downloads/840850/DB-Designer---Dise%C3%B1o-de-Base-de-Datos.html>
- Formacion.com. Normalización de la base de datos. Disponible en: http://www.adrformacion.com/curso/aplicacionesaccesxp/leccion1/normalizacion_base_datos.htm
- Trucos técnicos. Normalización de Bases de Datos. Disponible en: http://www.trucostecnicos.com/trucos/ver.php?id_art=278
- MySQL Hispano. Normalización de bases de datos. Disponible en: <http://www.mysql-hispano.org/page.php?id=16&pag=2>
- TUTORIAL DEL CURSO DE BASES DE DATOS 1. Disponible en: <http://atenea.udistrital.edu.co/profesores/jdimate/basedatos1/temas4.htm>
- García, Carlos. Curso: Diseño de base de datos relacionales. Disponible en: <http://www.mailxmail.com/curso/informatica/disenobasesdatosrelacionales/capitulo9.htm>

Bibliografía

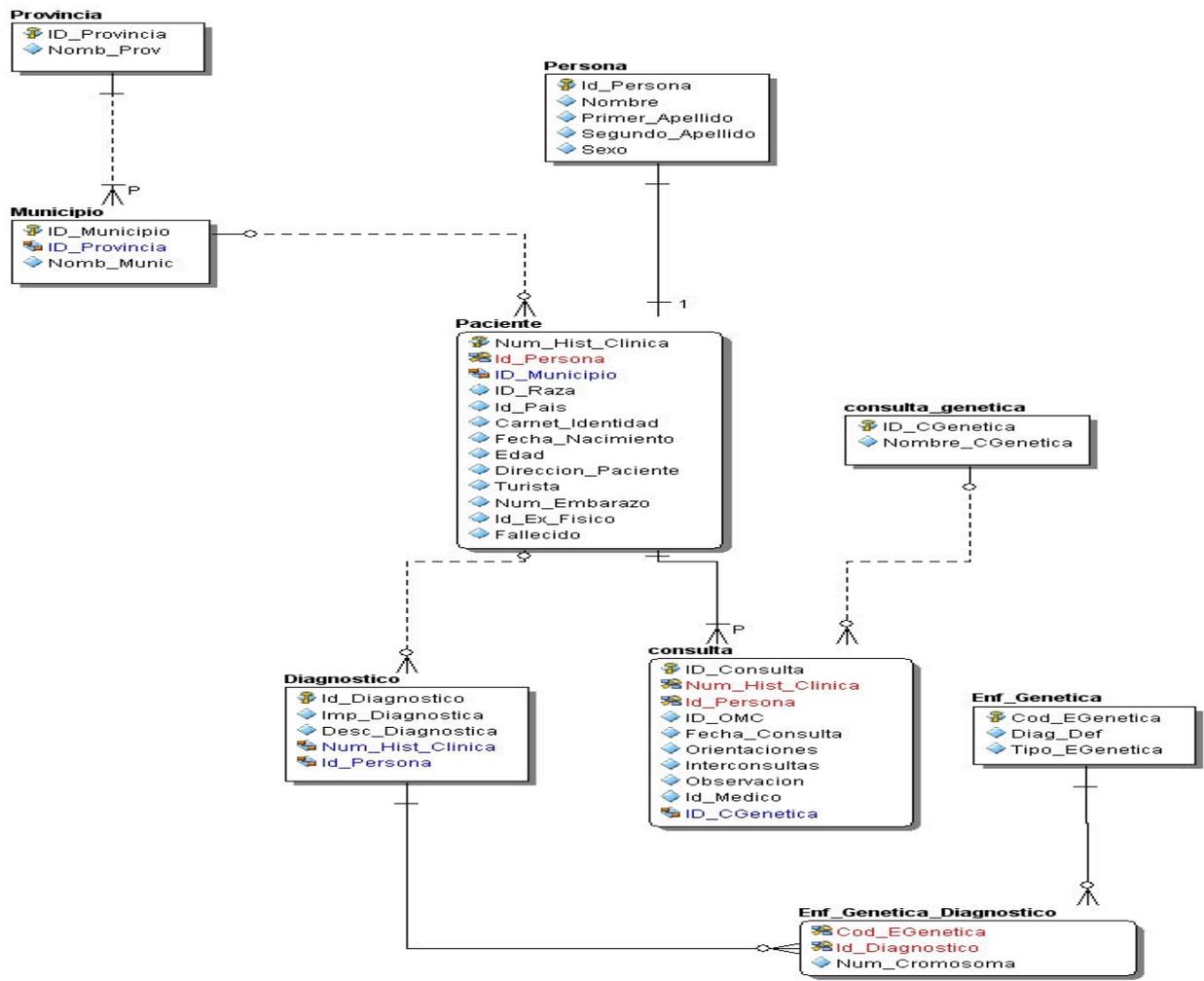
- Perezleo, Ligeya; Arencibia, Ricardo. Impacto de la Bioinformática en las ciencias biomédicas. Disponible en: http://bvs.sld.cu/revistas/aci/vol11_4_03/aci07403.htm
- Moreno, Antonio. Disponible en: <http://elies.rediris.es/elies9/4-2-3.htm>
- CA ER/Studio Data Modeler es la primera herramienta de diseño en integrarse con Microsoft Visual Studio 2005 Team Edition for Databa. Disponible en: <http://ca.com/es/press/release.aspx?cid=145405>
- Metodologías de Desarrollo de Software. Disponible en: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>

ANEXOS

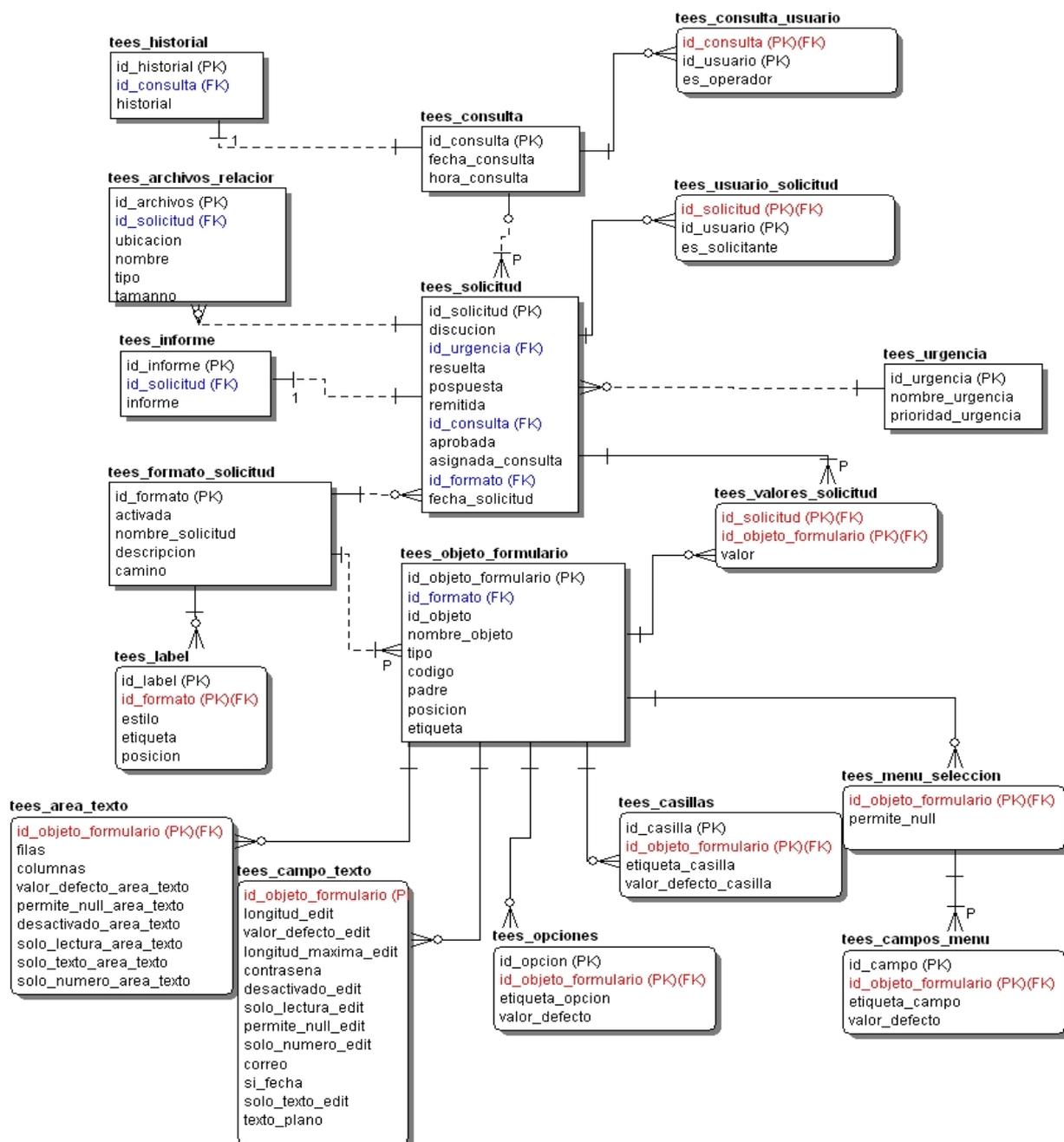
Anexo 1



Anexo 2



Anexo 4



Anexo 7**Consulta 1:**

```
SELECT * from `dg_pacientes`  
    inner join `dis_discapacitados` on `dis_discapacitados`.`id_paciente` = `dg_pacientes`.`id_paciente`  
    inner join `dis_vivienda` on `dis_vivienda`.`id_vivienda` = `dis_discapacitados`.`id_vivienda`  
where `dg_pacientes`.`edad` BETWEEN 20 and 60
```

Consulta 2:

```
SELECT * from `dg_pacientes`  
    inner join `dis_discapacitados` on `dis_discapacitados`.`id_paciente` = `dg_pacientes`.`id_paciente`  
    inner join `dis_vivienda` on `dis_vivienda`.`id_vivienda` = `dis_discapacitados`.`id_vivienda`  
where `dg_pacientes`.`edad` BETWEEN 20 and 60  
    and `dis_vivienda`.`cant_habitantes_nucleo` > 3  
    and `dis_discapacitados`.`amparo_filial_disc` = 1
```

Consulta 3:

```
SELECT * from `dg_pacientes`  
    inner join `dis_discapacitados` on `dis_discapacitados`.`id_paciente` = `dg_pacientes`.`id_paciente`  
    inner join `dis_vivienda` on `dis_vivienda`.`id_vivienda` = `dis_discapacitados`.`id_vivienda`  
    inner join `dg_ncolor_piel` on `dg_ncolor_piel`.`id_color_piel` = `dg_pacientes`.`id_color_piel`  
    inner join `dg_ncolor_ojos` on `dg_ncolor_ojos`.`id_color_ojos` = `dg_pacientes`.`id_color_ojos`  
where `dg_pacientes`.`edad` BETWEEN 20 and 60  
    and `dis_vivienda`.`cant_habitantes_nucleo` > 3  
    and `dis_discapacitados`.`amparo_filial_disc` = 1  
    and `dg_ncolor_ojos`.`descripcion` = "Azules"  
    and `dg_ncolor_piel`.`descripcion` = "Negro"
```

GLOSARIO DE TÉRMINOS

- **CASE:** Computer Aided Software Engineering (Herramientas de Ingeniería de Software Asistida por Computadora).
- **Framework Symfony:** es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web.
- **GPL:** GNU Public License acrónimo en inglés de General Public License (Licencia Pública General). Esta licencia regula los derechos de autor de los programas de software libre (free software) promovido por el Free Software Foundation (FSF) en el marco de la iniciativa GNU. Permite la distribución de copias de programas (e incluso cobrar por ello), así como modificar el código fuente de los mismos o utilizarlo en otros programas.
- **Infomed:** Infomed es la Red Telemática de Salud en Cuba. Dedicada a proporcionar información médica actualizada tanto nacional como internacional utilizando las nuevas tecnologías de la información y las comunicaciones (NTIC).
- **Licencia BSD:** La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution).
- **MINSAP:** Ministerio de Salud Pública.
- **MySQL AB:** MySQL AB es una compañía de software fundada en 1995, creadora del sistema administrador de bases de datos relacionales MySQL.
- **Release del proyecto:** Versión del proyecto.
- **SAAA:** Autenticación, Autorización y Auditoría (Authentication, Authorization, Accounting, AAA) con Autenticación de firma única (Single Sign On).
- **SISalud:** Servicios Integrales de salud.
- **Softel:** Empresa que ofrece soluciones informáticas para el Sistema de Salud.
- **Trigger:** Son tipos especiales de procedimientos almacenados que son ejecutados automáticamente cuando ciertos eventos se llevan a cabo.
- **UML (Unified Modeling Language):** es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.