

Universidad de las Ciencias Informáticas
Facultad # 7



**SISTEMA DE ALMACENAMIENTO DE INFORMACIÓN EN UN
SISTEMA DE GESTIÓN FARMACÉUTICA**

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

Autora: Suleydis Suárez Serpa.
Tutora: Ing. Arianne Méndez Mederos.

Ciudad de La Habana, Julio de 2007.

*No basta saber, se debe también aplicar.
No es suficiente querer, se debe también hacer.*

Johann Wolfgang Goethe

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 6 días del mes de julio del año 2007.

Suleydis Suárez Serpa

Ing. Arianne Méndez Mederos

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO.

Tutora:

Ing. Arianne Méndez Mederos, recién graduada de la Universidad de las Ciencias Informáticas y actual profesora de la facultad 7.

e-mail arianne@uci.cu

Asesor:

Ing. William Soñora Cruz, instructor recién graduado de la CUJAE y actual profesor de la facultad 7.

e-mail williamsc@uci.cu

Agradecimientos

A mis padres, por ser las mejores personas de este mundo, por dedicarme su vida y quererme tanto,
por tantos años de empeño y esfuerzo.

A mi familia, por darme su apoyo incondicional a lo largo de estos cinco años.

A Amarilis, por ser más que amiga, una familia y por toda la ayuda brindada.

A Yudi, por tener paciencia conmigo y ser la mejor amiga del mundo.

A mis compañeros de estudio, que han ocupado un lugar especial durante los últimos cinco años
compartiendo los momentos buenos y malos, en especial a Yorday y Daneysi.

A mis compañeros de proyecto Juan Alfredo y Yojanier.

A mi tutora, Ing. Arianne Méndez Mederos.

A todos los profesores y personas, que a lo largo de mi vida estudiantil han contribuido a mi formación
profesional.

A todos aquellos que no he mencionado, pero que siempre tendrán un pedacito de mi corazón.

A todos, muchas gracias.

Dedicatoria

A mis padres, por toda la ayuda brindada y la confianza depositada en mí. Son muy especiales.

A mi hermano.

A mi familia en general.

RESUMEN

Actualmente la gestión de la información en los procesos que en las farmacias hospitalarias se desarrollan se hace de forma manual, lo que provoca que: se introduzcan errores, haya pérdida de información, se alargue el período de actualización de la información, haya deterioro de documentos que contienen información, no se obtenga con rapidez la información que se necesite consultar, etc. Es por eso que se hace necesario, el diseño de un sistema que permita el almacenamiento, la gestión y el control de dicha información.

Con este trabajo se persigue el objetivo de diseñar una Base de Datos (BD) que permita almacenar información referente a los diversos procesos que se desarrollan en la gestión de los servicios farmacéuticos en los hospitales de nuestro país.

En el diseño de esta BD se utiliza la herramienta CaseStudio para el modelado de la misma, EMS SQL Manager for PostgreSQL para la administración y desarrollo del Sistema Gestor de Base de Datos utilizado, PostGreSQL 8.2. Se realiza la validación teórica de la Base de Datos diseñada mediante el análisis de propiedades como: integridad, normalización, trazabilidad y seguridad.

Con el desarrollo de propuesta, se espera que se logre el incremento de la capacidad organizativa de la información gestionada en las farmacias hospitalarias del país así como el aumento de la calidad de la asistencia farmacéutica a todos los servicios y personas que necesitan de la prescripción de productos farmacéuticos.

Palabras Claves

Base de Datos (BD), Sistema Gestor de Base de Datos (SGBD), gestión farmacéutica.

Índice

Resumen.....	VII
Índice de tablas.....	XI
Capítulo 1. Fundamentación teórica	18
1.1 Conceptos básicos relacionados con el dominio del problema.....	18
Farmacia hospitalaria	18
Gestión Farmacéutica Hospitalaria	18
Atención Farmacéutica.....	18
1.2 Análisis de las posibles herramientas de desarrollo a utilizar.....	19
1.2.1 Rational Rose Enterprise Edition.....	19
1.2.2 ER/Studio.....	20
1.2.3 Toad	20
1.2.4 CaseStudio	21
1.3 Sistemas Gestores de Base de Datos.....	21
1.3.1 Objetivos de un SGBD.....	22
1.3.2 Funciones de los SGBD.....	23
1.3.3 Funciones que se deben evaluar en un SGBD.	23
1.3.4 Arquitectura de Base de Datos.....	24
1.3.5 Características principales de algunos SGBD.....	26
SQL Server.....	26
MySQL.....	27
InterBase.....	28
Oracle.....	28
PostgreSQL	29
1.3.6 PostGreSQL vs. MySQL.	31

1.3.7 Herramienta para la administración y desarrollo de PostgreSQL.....	33
EMS SQL Manager for PostgreSQL.....	33
1.4 SGBD utilizados en sistemas de gestión farmacéutica hospitalaria existentes en nuestro país y en el mundo.	34
FARHOS.....	34
GESTIFARMA.....	35
CARE 2X.....	35
GALEN FARMACIA.....	35
BIOCOM.....	36
InFarmVisual.....	36
Conclusiones.....	36
Capítulo 2. Descripción y análisis de la solución propuesta.....	38
2.1 Estrategia de integración de la solución con otros módulos o sistemas.....	38
2.2 Descripción de la arquitectura y fundamentación.....	39
2.2.1 Modelo de datos relacional.	39
2.2.2 Arquitectura Centralizada.	40
2.3 Especificación de Requerimientos del sistema.	42
2.3.1 Requisitos funcionales.	42
2.3.2 Requisitos no funcionales.	45
2.4 Modelo de objetos o diagrama de clases persistentes.	48
2.4.1 Diagrama de clases persistentes obtenido a partir del diagrama de clases del diseño.....	49
Figura 2. Diagrama de clases persistentes.....	49
2.5 Descripción de las clases persistentes.....	50
2.6 Diseño de la Base de Datos.....	57
2.6.1 Diagrama Entidad Relación de la BD.	57
Figura 3. Diagrama Entidad Relación 1-1.....	57
Figura 4. Diagrama Entidad Relación 1-2.....	58

Figura 5. Diagrama Entidad Relación 2-1.....	59
Figura 6. Diagrama Entidad Relación 2-2.....	60
2.6.2 Descripción de las tablas.....	61
2.7 Análisis de optimización de consultas.	88
Consultas SELECT	89
Conclusiones.....	93
Capítulo 3. Validación del diseño realizado.	94
3.1 Validación teórica del diseño.....	94
3.1.1 Integridad	94
3.1.2 Normalización de la Base de Datos.	98
3.1.3 Análisis de redundancia de información.....	100
3.1.4 Análisis de la seguridad de la base de datos.	101
3.1.5 Trazabilidad de la acciones.....	104
Conclusiones.....	106
Recomendaciones.....	107
Referencias Bibliográficas	108
Bibliografía	112
Glosario de términos	114

ÍNDICE DE TABLAS

Tabla 1. Descripción de la entidad Producto.....	50
Tabla 2. Descripción de la entidad Movimiento_Producto.	50
Tabla 3. Descripción de la entidad Control_Producto.....	51
Tabla 4. Descripción de la entidad Control_Stock.....	51
Tabla 5. Descripción de la entidad Pedido.	52
Tabla 6. Descripción de la entidad Inventario.	52
Tabla 7. Descripción de la entidad Entrada.....	53
Tabla 8. Descripción de la entidad Solicitud.....	53
Tabla 9. Descripción de la entidad Devolucion.....	54
Tabla 10. Descripción de la entidad Receta_Medica.....	54
Tabla 11. Descripción de la entidad Stock_Servicio.	55
Tabla 12. Descripción de la entidad Producto_Baja.	55
Tabla 13. Descripción de la entidad Usuario_Sistema.	56
Tabla 14. Detallada descripción de la tabla fm_producto_farmacia.	61
Tabla 15. Detallada descripción de la tabla fm_control_productos.	62
Tabla 16. Detallada descripción de la tabla fm_proveedor.	63
Tabla 17. Detallada descripción de la tabla fm_pedidos.	64
Tabla 18. Detallada descripción de la tabla fm_movimiento_producto.....	64
Tabla 19. Detallada descripción de la tabla fm_libro_fecha_vencimiento.....	65
Tabla 20. Detallada descripción de la tabla fm_inventario.....	66
Tabla 21. Detallada descripción de la tabla fm_sector_farmacia.....	67
Tabla 22. Detallada descripción de la tabla fm_documento_entrada.....	67
Tabla 23. Detallada descripción de la tabla fm_documento_solicitud.....	68
Tabla 24. Detallada descripción de la tabla fm_nombre_comercial.	69
Tabla 25. Detallada descripción de la tabla fm_consumo_paciente.	70
Tabla 26. Detallada descripción de la tabla fm_consumo_paciente.....	70
Tabla 27. Detallada descripción de la tabla fm_prescripcion_paciente.	71

Tabla 28. Detallada descripción de la tabla fm_parametro_producto.	72
Tabla 29. Detallada descripción de la tabla fm_devolucion_producto.	73
Tabla 30. Detallada descripción de la tabla nomencladora cfg_codigo_producto.	74
Tabla 31. Detallada descripción de la tabla nomencladora cfg_grupo_producto.	75
Tabla 32. Detallada descripción de la tabla nomencladora cfg_subgrupo_producto.	75
Tabla 33. Detallada descripción de la tabla nomencladora cfg_concentracion.	76
Tabla 34. Detallada descripción de la tabla nomencladora cfg_forma_almacenamiento.	76
Tabla 35. Detallada descripción de la tabla nomencladora cfg_forma_almacenamiento.	77
Tabla 36. Detallada descripción de la tabla nomencladora cfg_forma_farmaceutica.	77
Tabla 37. Detallada descripción de la tabla nomencladora cfg_forma_farmaceutica.	78
Tabla 38. Detallada descripción de la tabla nomencladora cfg_tipo_movimiento_farmacia.	78
Tabla 39. Detallada descripción de la tabla nomencladora cfg_tipo_sector.	79
Tabla 40. Detallada descripción de la tabla nomencladora cfg_unidad_externa.	79
Tabla 41. Detallada descripción de la tabla nomencladora cfg_servicio.	80
Tabla 42. Detallada descripción de la tabla nomencladora cfg_motivos_baja_prod_fm.	80
Tabla 43. Detallada descripción de la tabla nomencladora cfg_fabricante_producto.	81
Tabla 44. Detallada descripción de la tabla nomencladora cfg_estado_producto.	81
Tabla 45. Detallada descripción de la tabla nomencladora cfg_estado_pedido.	82
Tabla 46. Detallada descripción de la tabla nomencladora cfg_estado_solicitud.	82
Tabla 47. Detallada descripción de la tabla nomencladora cfg_usuario_ent_sal.	83
Tabla 48. Detallada descripción de la tabla nomencladora cfg_usuario_sistema.	83
Tabla 49. Detallada descripción de la tabla de relación rel_control_prod_inventario.	84
Tabla 50. Detallada descripción de la tabla de relación rel_prod_devolucion.	85
Tabla 51. Detallada descripción de la tabla de relación rel_prod_devolucion.	86
Tabla 52. Detallada descripción de la tabla de relación rel_producto_doc_entrada.	86
Tabla 53. Detallada descripción de la tabla de relación rel_producto_doc_entrada.	87
Tabla 54. Detallada descripción de la tabla de relación rel_solicitud_movimiento.	87
Tabla 55. Detallada descripción de la tabla de relación rel_servicio_producto.	88

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

En Cuba, en estos momentos se propone una utilización justa y racional sobre principios éticos sostenibles, de la informática, que se hace cada día más importante en las diversas áreas del quehacer humano, debido a su rapidez para analizar, procesar y comunicar grandes cantidades de información.

Uno de los campos de la actividad humana que se ha visto más beneficiado por la informática es el área de la salud. La informatización en esta área está dada por el conjunto de métodos, técnicas, procederes y actividades gerenciales encaminadas al manejo de la información en este sector, la cual comprende la información del estado de salud de la población, los datos sobre el conocimiento de las ciencias de la salud y la información en general para la toma de decisiones, clínico-epidemiológicas, operativas y estratégicas. [1]

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

A partir de 1997 se concibe una primera estrategia de informatización como respuesta del sector de la salud a los lineamientos estratégicos trazados para la informatización de la sociedad cubana, con la finalidad de coordinar esfuerzos para el desarrollo de este proceso en el SNS, de esta forma surge la propuesta de crear Sistemas de Información Hospitalaria (SIH). Esta propuesta fue hecha realidad con el esfuerzo de muchas personas y desde entonces han sido desarrolladas diferentes versiones.

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

Actualmente se continua trabajando integradamente, estrechando alianzas, en el desarrollo de este grupo de aplicaciones básicas de desarrollo tecnológico, y es precisamente esta integración la que permite hablar de informatización en el sector de la salud pública cubana.

Se trabaja en proyectos basados en nuevas tecnologías de Internet, *software* libre y otras que garantizan una explotación integrada y compatible. En su desarrollo e implementación participan diferentes empresas del Ministerio de la Informática y Comunicaciones (MIC) como Desoft, Softel, PcMax, Sys y la Universidad de Ciencias Informáticas (UCI), entre otros.

La UCI, protagonista actual de todo el proceso de informatización que se desarrolla en nuestro país, tiene como una de sus tareas primordiales el perfeccionamiento de *software* ya existente y entre ellos los mencionados SIH.

Este sistema informático consta de varios módulos estrechamente relacionados, entre ellos uno destinado al desarrollo de un sistema integral de información para las farmacias hospitalarias pero este no cuenta con una actualización de muchas de las funciones que se desarrollan actualmente en las mismas.

El aumento de la expectativa de vida, de la prevalencia de enfermedades crónicas y de la gama de medicamentos disponibles, convierte a la farmacoterapia en la forma de intervención médica más frecuentemente utilizada.

Una farmacoterapia apropiada permite obtener una atención sanitaria segura y económica, en tanto que el uso inadecuado de los fármacos tiene importantes consecuencias para los pacientes. Es necesario asegurar un uso adecuado, racional y económico de los medicamentos.

Los farmacéuticos tienen una responsabilidad fundamental en la atención de las necesidades de los individuos y de la sociedad en este aspecto por lo que se hace necesario la búsqueda de nuevos métodos que proporcionen mayor seguridad en cada uno de los procesos llevados a cabo por estos.

Para que todas sus actividades puedan llegar a buen fin, tanto su intervención en el ámbito de la salud como en el plano comercial requieren de una enorme capacidad de planificación, ejecución y control.

Los servicios farmacéuticos en los hospitales consisten en el conjunto de actividades desarrolladas por un profesional farmacéutico, que están encaminadas a la promoción del uso adecuado y racional de los medicamentos. Dichas actividades, están dirigidas, tanto a los profesionales de la salud, como a los pacientes.

En las farmacias hospitalarias del país se maneja gran cantidad de información referente a los productos farmacéuticos que en estas se gestionan, los medios utilizados para el procesamiento de estos datos actualmente son de forma manual lo que conlleva a que haya demora en la recepción de estos y el nivel de error sea considerable.

Existen tarjetones que son llamados tarjetas de estiba, donde se registra toda la información necesaria pero se hace un poco difícil y trabajosa la utilización y manipulación en general de los datos, además esta información no tiene seguridad al ser almacenada de esta forma, estas pueden sufrir deterioro con el tiempo. La función de estas tarjetas de estiba es archivar toda la información referente a la tramitación de los medicamentos, ya sean entregados, vencidos, solicitados o cualquier otra gestión que se necesite realizar.

Para llevar a cabo las solicitudes de productos farmacéuticos que son realizadas por unidades internas se utilizan documentos de papel. Los pedidos que se realizan de la farmacia a otras unidades y las devoluciones de medicamentos y productos entregados se archivan a través de planillas impresas, lo que conlleva a que existan dificultades cuando se necesita buscar información referente a la entrega de los mismos.

Es de gran importancia tener presente las fechas de vencimientos de cada uno de los medicamentos, la vía que se utiliza para darle solución a esta problemática es la utilización de un libro donde se registra y almacena este dato pero este libro puede deteriorarse o bien extraviarse perdiéndose esta valiosa información. Además no existe actualmente la forma de alertar con antelación la aproximación de la fecha de vencimiento de los productos.

Para tener control sobre los consumos de cada producto en la farmacia por período de tiempo se realizan los cálculos de stock, estos cálculos se realizan con el objetivo de tener un estricto control de los valores máximos y mínimos de consumo para saber el comportamiento de estos y poder dar un tratamiento

diferenciado, todo este proceso se realiza de forma manual corriendo el riesgo de que exista error generándose de esta forma información incorrecta.

Las tarjetas, documentos y libros anteriormente mencionados han sido la solución a muchos problemas existentes en el almacenamiento de los diferentes datos que se manipulan en las farmacias hospitalarias.

Teniendo en cuenta que en estos momentos se trabaja integradamente en el desarrollo de un grupo de aplicaciones básicas para la informatización del sector de la salud y con el desarrollo tecnológico con el cual contamos, para nada es imposible llevar a cabo un proceso de informatización en las farmacias hospitalarias que permita buscar, obtener, actualizar, mostrar, recuperar, eliminar o modificar los datos siempre que sea necesario de manera clara, sencilla, ordenada y segura, de esta forma se expresará con mayor eficiencia y calidad el trabajo en la atención farmacéutica.

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

Basado en lo antes expuesto se tiene como **problema**: ¿Cómo automatizar la gestión de la información referente a los procesos vinculados con las farmacias en los hospitales cubanos?

En correspondencia con el problema el **objeto de estudio** lo constituyen los procesos de gestión farmacéutica hospitalaria.

El **campo de acción** esta centrado en la informatización de los procesos de gestión de las farmacias de los centros hospitalarios.

Como **objetivo general** se define: Diseñar una Base de Datos que mejore la gestión de la información referente a los procesos vinculados a las farmacias hospitalarias.

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

- Realizar un estudio detallado de las herramientas a utilizar para el diseño de la BD.

- Estructurar y diseñar un modelo conceptual de la BD que cumpla con lo establecido.
- Obtener el Script que se genera en el CaseStudio.
- Obtener los artefactos de la metodología RUP que describen la BD.
- Implementar las funciones, vistas y dominios de la BD del módulo Gestión Farmacéutica.
- Realizar validación teórica del diseño propuesto para la BD.

El documento está estructurado de la manera siguiente:

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

Capítulo 2: Descripción y análisis de la solución propuesta: Descripción y profundo análisis de la solución que se propone para darle solución a la problemática planteada.

Capítulo 3: Validación del diseño realizado: Validación teórica del diseño de la BD.

Cada uno de estos capítulos es iniciado por una breve introducción donde se dan a conocer los temas que se desarrollarán durante el mismo. Finaliza con las conclusiones, en las que se plantean los resultados obtenidos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

En este capítulo, se hará un estudio detallado de los principales conceptos y tecnologías utilizadas en el desarrollo de la BD del módulo de Gestión Farmacéutica. Se abordan temas relacionados con la gestión de las farmacias hospitalarias y Sistemas de Gestión de Bases de Datos (SGBD).

1.1 Conceptos básicos relacionados con el dominio del problema.

Farmacia hospitalaria

Es una especialidad que se ocupa de servir a la población en sus necesidades farmacéuticas, a través de la selección, preparación, adquisición, control, dispensación, información de medicamentos y otras actividades orientadas a conseguir una utilización apropiada, segura y costo-efectiva de los medicamentos y productos sanitarios, en beneficio de los pacientes atendidos en el establecimiento asistencial. [2]

Gestión Farmacéutica Hospitalaria

La gestión es una actividad que permite adquirir, ordenar y emplear adecuadamente los recursos en función de un objetivo dado.

Desde la antigüedad, el hombre se enfrenta a fenómenos naturales, económicos, políticos y sociales y debe desplegar un conjunto de actividades para alcanzar sus objetivos. Ese conjunto de actividades no es más que su gestión. En las farmacias hospitalarias, se desarrollan diferentes dimensiones de la gestión.

La gestión farmacéutica no es exclusiva de aquellos que ocupan responsabilidades administrativas o directivas, sino de todo aquel que aspire a desempeñar su mejor papel en pos de sus objetivos y metas.

Atención Farmacéutica

La Organización Mundial de la Salud (OMS) define la Atención Farmacéutica (AF) como el compendio de práctica profesional, en el que el paciente es el principal beneficiario de las acciones del farmacéutico; es el compendio de actitudes, comportamientos, compromisos, inquietudes, valores éticos, funciones, conocimientos, responsabilidades y destrezas del farmacéutico en la prestación de la farmacoterapia, con el objetivo de lograr resultados terapéuticos definidos en la salud y calidad de vida del paciente. [3]

Es la participación activa del farmacéutico para la asistencia al paciente en la dispensación y seguimiento de un tratamiento farmacoterapéutico, cooperando así con el médico y otros profesionales sanitarios a fin de conseguir resultados que mejoren la calidad de vida del paciente. También conlleva la implicación del farmacéutico en actividades que proporcionen buena salud y prevengan las enfermedades.

1.2 Análisis de las posibles herramientas de desarrollo a utilizar.

Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador).

“Se puede definir a una herramienta CASE como 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*.” [4]

CASE es también definido como el 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. [4]

1.2.1 Rational Rose Enterprise Edition

Rational Rose es una herramienta CASE basada en el Lenguaje Unificado de Modelado (UML) que permite crear los diagramas que se van generando durante el proceso de ingeniería en el desarrollo del *software*.

UML (Unified Modeling Language) es la herramienta usada en la descripción y construcción de software reconocida por la industria como estándar mientras que el Proceso Unificado Rational (Rational Unified Process) es la metodología de IBM Rational para el desarrollo y construcción de software basado íntegramente en UML como soporte a la metodología. [5]

Rational Rose es completamente compatible con esta metodología RUP, posee un gran número de estereotipos predefinidos que facilitan el proceso de modelación del *software*. Además es capaz de generar el código fuente de las clases definidas en el flujo de trabajo de diseño, pero tiene la limitación de que aún hay varios lenguajes de programación que no soporta o que sólo lo hace a medias.

Una vez que se tiene el diagrama de las clases persistentes a partir del cual se genera la BD del sistema, no existe la posibilidad de exportar ese modelo hacia algún SGBD.

1.2.2 ER/Studio

Es una herramienta de diseño UML, de modelado de datos fácil de usar y multinivel, se utiliza para realizar el modelado para el diseño y creación de BD lógicas y físicas. Permite crear y mantener aplicaciones de BD.

ER/Studio está equipado para crear y manejar diseños de BD funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de BD, documentación y fácil creación de reportes. [6]

Las capacidades de diseño que contiene, ayudan a crear un diseño lógico que puede transformarse en cualquier número de diseños físicos. Como resultado, se puede mantener un diseño lógico normalizado mientras se desnormalizan los diseños físicos para su desempeño.

ER/Studio mantiene ligas entre todos los niveles de su diseño por lo tanto puede mezclar cambios en cualquier dirección entre ellos. Además revisa la normalización y la compilación con la sintaxis de la plataforma de la BD.

ER/Studio permite tomar por omisión las opciones para todos los diagramas así como realizar cambios al momento de la ejecución. También ayuda a prolongar la inversión que se haya hecho. Soporta el proceso de diseño interactivo inherente en el ciclo de vida de la aplicación.

1.2.3 Toad

Toad Data Modeler es una aplicación que permite diseñar esquemas de BD y generar el código SQL necesario para producirlas.

Permite desarrollar aplicaciones con mayor rapidez y facilidad, simplificando las tareas de administración de BD y con posibilidad de trabajar en varios documentos simultáneamente. [7]

Con esta herramienta se pueden desarrollar diagramas para la mayor parte de SGBD existentes: Access, Firebird, InterBase, MySQL, Oracle, Paradox, Postgresql, entre otros. Tiene un navegador interno que permite visualizar la BD y gestionar sus objetos con facilidad. El programa es capaz además de conectarse a varias BD simultáneamente.

La aplicación resulta muy útil para crear diagramas de entidad-relación, definir reglas de integridad referencial, generar scripts SQL que construyan la BD.

Posee una herramienta denominada 'Model Explorer' que permite navegar por todos los atributos del modelo que se esté creando.

Esta herramienta tiene la desventaja que solamente permite crear hasta un límite determinado, 25 tablas, es decir, no puede ser utilizado para diseñar un modelo que contenga más de 25 tablas.

1.2.4 CaseStudio

CASE Studio es una herramienta profesional con la que se puede diseñar BD, facilitando herramientas para la creación de diagramas de relación, modelado de datos y gestión de estructuras.

Tiene soporte para trabajar con una amplia variedad de formatos de BD (Oracle, SQL, MySQL, MS SQL, MaxDB, PostgreSQL, Access, etc.), permite además generar scripts SQL, usar plantillas de diseño personalizables y crear detallados informes en HTML. [8]

Consta de un potente sistema de ingeniería inversa, que permite identificar y estructurar BD ya existentes para poder trabajar con ellas sin problemas.

A través de los diagramas de relación que se realizan en el CaseStudio se tiene una visión más clara del contenido y estructura de la BD, facilitando la gestión y mantenimiento de la misma.

1.3 Sistemas Gestores de Base de Datos.

Una BD no es más que un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos. Es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo. [9]

El *software* de BD ha venido experimentado un auge paralelamente con el progreso de la informatización en las diferentes ramas se de la totalidad de las empresas de hoy día.

La efectividad, rapidez, confiabilidad, seguridad en los flujos de información y en general en todos los procesos constituyen la necesidad apremiante a la hora de optimizar servicios y productos. Ante esta demanda de soluciones informáticas ha surgido multitud de SGBD.

Un SGBD, puede definirse como un paquete generalizado de *software*, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario.

Las principales funciones que debe cumplir un SGBD es relacionarse con la creación y mantenimiento de la BD, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

1.3.1 Objetivos de un SGBD

En un ambiente multiusuario el SGBD ofrece un control centralizado de la información. Los objetivos que plantean estos sistemas están relacionados con la intención de evitar los problemas que existían en los sistemas de información orientados a los procesos.

Los principales objetivos son:

- Evitar la redundancia de los datos, eliminando así la inconsistencia de los mismos.
- Mejorar los mecanismos de seguridad de los datos y la privacidad.

Se pueden distinguir cuatro tipos de contextos para usar mecanismos de seguridad:

- seguridad contra accesos indebidos a los datos
- seguridad contra accesos no autorizados a la BD.
- seguridad contra destrucción causada por el-entorno (fuego, inundación, robo, etc.)
- seguridad contra fallos del propio sistema (fallos del *hardware*, del *software*, etc.)

- Mantener la integridad de los datos realizando las validaciones necesarias cuando se realicen modificaciones en la BD.
- Asegurar la independencia de los programas y los datos, es decir, la posibilidad de modificar la estructura de la BD (esquema) sin necesidad de modificar los programas de las aplicaciones que manejan esos datos.
- Mejorar la eficacia de acceso a los datos, en especial en el caso de consultas imprevistas.

1.3.2 Funciones de los SGBD.

Las principales funciones que debe realizar un SGBD son:

- La definición de los datos.
- La manipulación de los datos.
- Garantizar la seguridad e integridad de los datos.
- La gestión de las transacciones y el acceso concurrente. [10]

1.3.3 Funciones que se deben evaluar en un SGBD.

Una de las funciones primarias que presentan las BD y a partir de la cual empezaron a evolucionar, es el almacenamiento y recuperación de datos. Entre las tareas de recuperación de datos se pueden encontrar:

- la velocidad de ejecución de consultas.
- la creación de índices.
- la importación de datos.

La evaluación se ha realizado en función del tiempo que tarda cada SGBD en llevarlas a cabo, ya que la velocidad representa la eficiencia en la realización de cada función.

También existen variables menos importantes que podrían condicionar el rendimiento entre ellas:

- La protección de datos (bloqueos para garantizar el acceso exclusivo).
- La asignación de memoria interna.
- El control de accesos de los usuarios.
- El consumo de recursos (esto último se refiere a la capacidad de funcionar en un entorno de recursos limitados y en armonía con otras aplicaciones).

1.3.4 Arquitectura de Base de Datos.

El SGBD incorpora como herramienta fundamental dos lenguajes, para la definición y la manipulación de los datos.

- Lenguaje de definición de datos (DDL, Data Definition Language) provee de los medios necesarios para definir los datos con precisión, especificando las distintas estructuras. Acorde con el modelo de arquitectura de tres niveles, habrá un lenguaje de definición de la estructura lógica global, otro para la definición de la estructura interna, y un tercero para la definición de las estructuras externas.
- Lenguaje de manipulación de datos (DML, Data Manipulation/ Management Language), es el encargado de facilitar a los usuarios el acceso y manipulación de los datos. Pueden diferenciarse en procedimentales (aquellos que requieren qué datos se necesitan y cómo obtenerlos) y no procedimentales (que datos se necesitan, sin especificar como obtenerlos), y se encargan de la recuperación de los datos almacenados, de la inserción y supresión de datos en la BD, y de la modificación de los existentes. [11]

Las BD respetan la arquitectura de tres niveles definida, para cualquier tipo de BD, por el grupo ANSI/SPARC. En esta arquitectura la BD se divide en los niveles externo, conceptual e interno:

- Nivel interno: es el nivel más bajo de abstracción, y define cómo se almacenan los datos en el soporte físico, así como los métodos de acceso.

- Nivel conceptual: es el nivel medio de abstracción. Se trata de la representación de los datos realizada por la organización, que recoge las vistas parciales de los requerimientos de los diferentes usuarios y las aplicaciones posibles. Se configura como visión organizativa total, e incluye la definición de datos y las relaciones entre ellos.
- Nivel externo: es el nivel de mayor abstracción. A este nivel corresponden las diferentes vistas parciales que tienen de la BD los diferentes usuarios. En cierto modo, es la parte del modelo conceptual a la que tienen acceso. [12]

En ocasiones puede encontrarse el nivel conceptual dividido en dos niveles, conceptual y lógico. El primero de ellos corresponde a la visión del sistema global desde un punto de vista organizativo independiente, no informático. El segundo correspondería a la visión de la BD expresada en términos del sistema que se va a implantar con medios informáticos.

Este modelo de arquitectura permite establecer el principio de independencia de los datos. Esta independencia puede ser lógica y física. Por independencia lógica se entiende que los cambios en el esquema lógico no deben afectar a los esquemas externos que no utilicen los datos modificados. Por independencia física se entiende que el esquema lógico no se vea afectado por cambios realizados en el esquema interno, correspondientes a modos de acceso, etc.

Hasta el presente las compañías de *software* han desarrollado numerosos SGBD, estos se pueden clasificar en SGBD Libres y SGBD Comerciales atendiendo las diferencias entre utilizar *software* libre, *software* propietario o diferentes combinaciones.

A continuación se muestra una lista de los SGBD más comúnmente usados, distinguiendo entre los SGBD libres y los comerciales.

SGBD libres:

PostgreSQL.
MySQL.
SQLite Sybase ASE.
Firebird.

SGBD comerciales:

Microsoft SQL Server.
Oracle.
Fox Pro.
Microsoft Access.

1.3.5 Características principales de algunos SGBD.

El estudio y análisis de los diferentes SGBD que más se utilizan actualmente en el mundo se hace imprescindible para de acuerdo a la necesidad decidir cual es la mejor alternativa.

Entre los SGBD comúnmente utilizados en el mundo se encuentran:

SQL Server

Microsoft SQL Server es un SGBD relacional basado en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Microsoft SQL Server, pertenece a la familia de los sistemas de administración de BD, que operan en una arquitectura cliente/servidor de gran rendimiento.

Su desarrollo fue orientado para hacer posible manejar grandes volúmenes de información, y un elevado número de transacciones. Permite la creación de procedimientos almacenados que se ejecutan más rápido que instrucciones SQL independientes.

Puede manejar perfectamente BD de TeraBytes con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos, sólo depende de la potencia del hardware del equipo en el que esté instalado. Una de sus desventajas es que no es multiplataforma, solamente corre sobre Windows NT- 2000 Server. [13]

Algunas de sus características principales son las siguientes:

- Arquitectura de almacenamiento en disco, permite la escalabilidad desde BD de equipos portátiles hasta BD empresariales de tamaño de terabyte.
- Las herramientas y los programas se ejecutan mucho más rápido y están diseñados para que produzcan menos impacto en las operaciones del servidor.
- Las copias de seguridad activas proporcionan copias de seguridad en línea de alto rendimiento con un impacto mínimo en los sistemas en funcionamiento.
- El optimizador de consultas con múltiples fases busca el plan óptimo de consultas para mejorar el rendimiento de consultas complejas.
- Amplía la búsqueda de texto completo con alto rendimiento a documentos formateados a través de un mecanismo ampliable de filtro.
- Proporciona optimización automática de consultas y soporta consultas en múltiples servidores en la configuración distribuida de BD de SQL Server.

Desventajas:

- Licencias con costos altos.
- Plataformas Windows.

MySQL

Es un SGBD basado en Open Source (código abierto) diseñado para los sistemas Unix, aunque existen versiones para Windows.

Algunas de sus características principales son las siguientes:

- Diseñado en vistas a la velocidad.
- Consume muy pocos recursos de CPU y memoria. Muy buen rendimiento.
- Tamaño del registro sin límite.
- Buena integración con PHP.
- Buen control de acceso usuarios-tablas-permisos.

Desventajas:

- No soporta subconsultas.
- No soporta transacciones.
- No soporta triggers ni procedimientos en el servidor.
- No soporta claves ajenas. Ignora la integridad referencial.
- No soporta vistas.
- Se hace inestable cuando contiene gran cantidad de datos.

InterBase

Es un SGBD que tiene las siguientes características:

- Un producto con gran madurez: es un producto con muchos años de existencia en el mercado, siendo utilizado por diversas firmas, es un producto fiable y robusto, probado exhaustivamente y ofrece unos buenos niveles de seguridad.
- Código Abierto (Open Source): a partir de la versión 6, InterBase a pasado a ser Open Source, significando que las fuentes de la misma pueden ser editados y modificados por cualquier usuario.
- Coste de Desarrollo: baja considerablemente con la potencia que otorga SQL en las consultas, evitando usar bucles, instrucciones de salto, instrucciones de búsqueda, lo que también permite una mayor legibilidad del código.
- Tiene gran coste de desarrollo.
- Velocidad de desarrollo: si las cosas se hacen bien, el tiempo de desarrollo puede disminuir, además de permitir un mantenimiento más sencillo de la misma. [14]

Oracle

Es un SGBD fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de BD más completos, destacando su:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Su mayor inconveniente es su enorme precio, que es de varios miles de euros, según versiones y licencias. Otro aspecto a criticar es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad.

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

Aunque ha tenido un amplio dominio en el mercado de servidores sufre la competencia del Microsoft SQL Server de Microsoft y de otros con licencia libre como PostgreSQL, MySql o Firebird. [15]

PostgreSQL

PostgreSQL es un SGBD objeto-relacional para múltiples plataformas. Está considerado como la BD de código abierto más avanzada del mundo. Tiene licencia BSD (esta licencia básicamente consiste en que ves el código, puedes redistribuirlo y puedes modificarlo), debido a esto puede usarse, modificarse, y distribuirse por todos gratis para cualquier propósito, sea privado, comercial, o académico. Posee gran estabilidad y confiabilidad, nunca ha presentado caídas en varios años de operación de alta actividad. Tiene un buen soporte para subselects, triggers, vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos. [16]

Características importantes a destacar en el SGBD Postgres SQL son las siguientes:

- Corre en los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, entre otros.
- Soporte nativo para los lenguajes más populares como PHP, C, C++, Perl, Python, etc.
- Soporte de todas las características de una BD profesional (triggers, store procedures – funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, etc.)

- Utilidades para análisis y optimización de Querys.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. [17]
- PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Características adicionales en la versión Postgres SQL 8.2:

Además de las características listadas anteriormente, hay nuevas prestaciones y mejoras contribuidas por docenas de desarrolladores para versión 8.2. Entre ellas se encuentran:

Mejoras de Rendimiento: la versión 8.2 mejora el rendimiento en alrededor de un 20% en pruebas de sistemas OLTP (procesamiento en línea de transacciones) de gran escala. Los cambios incluyen ordenamientos en memoria y en disco más rápido, mejor escalabilidad en sistemas multi-procesador, mejor optimización de consultas sobre datos particionados, cargas masivas más rápidas y outer joins considerablemente acelerados.

Construcción en línea de índices: la construcción de índices puede ocurrir mientras las aplicaciones escriben a las tablas de la BD, permitiendo el afinamiento de rendimiento sin afectar la disponibilidad.

PL/Python: soporta parámetros con nombre y funciones que devuelven conjuntos. [18]

Limites de Postgre 8.2

Máximo de base de datos: ILIMITADO

Máximo de tamaño de tabla: 32TB

Máximo de tamaño de registro: 1.6TB

Máximo de tamaño de campo: 1GB

Máximo de registros por Tabla: ILIMITADO

Máximo de campos por tabla: 250 a 1600 (depende de los tipos usados)

Máximo de índices por tabla: ILIMITADO

Número de lenguajes en los que se puede programar funciones:

(Pl/pgsql, pl/java, pl/perl, pl/python, tcl, pl/php, C, C++, Ruby, etc.)

1.3.6 PostGreSQL vs. MySQL.

Actualmente los SGBD más populares en el ambiente del software libre y a su vez los más utilizados en los diferentes sistemas de Gestión Farmacéutica que existen en el mundo son PostgreSQL y MySQL. A continuación una breve comparación de estos dos SGBD después de haber consultado información al respecto:

Lo mejor de PostGreSQL:

Las características positivas que posee este gestor según las opiniones más comunes son:

1. Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (según algunas opiniones se dice que en muchas ocasiones ha llegado a soportar el triple de carga de lo que soporta MySQL).
2. Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.

3. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia BD, equiparándolo con los gestores de BD de alto nivel, como puede ser Oracle.
4. Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL.
5. Tiene mejor soporte para triggers y procedimientos en el servidor.

Los mayores inconvenientes y desventajas que se pueden encontrar a este gestor son:

1. Consume más cantidad de recursos que MySQL.
2. Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
3. Es más lento que MySQL.
4. Tiene menos funciones en PHP que MySQL.

Lo mejor de MySQL:

La gran mayoría de las personas utiliza este gestor en Internet, por lo que encontrar opiniones favorables no resulta en absoluto imposible:

1. MySQL tiene gran velocidad al realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
2. Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin presentar problemas.
3. Presenta grandes utilidades de administración, tiene gran facilidad de configuración e instalación.
4. Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

Debido a esta mayor aceptación en Internet, gran parte de los inconvenientes se exponen a continuación, estos han sido extraídos de comparativas con otras BD:

1. Carece de soporte para transacciones, rollback's y subconsultas.

2. El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
3. No es viable para su uso con grandes BD, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.
4. No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación. [19]

Se considera que PostgreSQL es un sistema de BD lento, es el servidor de BD de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, triggers, vistas, etc.

MySQL no es adecuada para aplicaciones críticas. Al no utilizar transacciones, un problema que interrumpiese una serie de comandos podría dejar la BD en un estado inconsistente, lo cual nunca ocurriría con PostgreSQL. Tampoco tiene triggers por lo que no se pueden establecer reglas de integridad y consistencia a nivel de servidor.

1.3.7 Herramienta para la administración y desarrollo de PostgreSQL.

EMS SQL Manager for PostgreSQL

EMS SQL Manager for PostgreSQL es una aplicación de alto desempeño para la administración y desarrollo del SGBD PostgreSQL. El programa trabaja con cualquier versión de PostgreSQL y soporta todas las últimas características de PostgreSQL, incluyendo espacios de tablas y nombres de argumentos en funciones. Su interfaz gráfica es sumamente atractiva e incluye un modo guiado de trabajo. Incluye las herramientas básicas de mantenimiento y administración.

Características:

- Soporte completo para PostgreSQL.
- Administración y navegación rápida de BD.
- Administración fácil de todos los objetos PostgreSQL.
- Administración efectiva de seguridad.

- Capacidades de exportación e importación de datos.
- Modo guiado para labores de mantenimiento.
- Interfaz atractiva. [20]

1.4 SGBD utilizados en sistemas de gestión farmacéutica hospitalaria existentes en nuestro país y en el mundo.

Actualmente existen en el mundo sistemas informáticos para la gestión farmacéutica hospitalaria. La mayoría de estos productos han sido desarrollados sobre la base de software propietario por lo que el acceso a su código fuente no está disponible. Estos sistemas no pueden ser usados ni modificados por personas no autorizadas.

Otra característica que se debe tener presente es que estos sistemas son implementados en aplicaciones de escritorio. Estos productos utilizan SGBD que aunque poseen diversas ventajas, muchos de estos traen consigo ciertos inconvenientes para la manipulación de los datos que se gestionan en el mismo.

Se hace imprescindible el estudio y análisis de cuales son los diferentes SGBD que utilizan algunos de estos sistemas de gestión farmacéuticas existentes en nuestro país y en el mundo.

FARHOS

Sistema integral de información para la farmacia hospitalaria. Fue desarrollado por un equipo de farmacéuticos especialistas en farmacia hospitalaria e informáticos con más de doce años de experiencia en el entorno sanitario y hospitalario. [21]

Este sistema gestiona todo lo referente a compras, prescripción asistida desde planta, distribución de medicamentos, dispensación individualizada tanto a pacientes ingresados como ambulatorios. Este sistema utiliza Interbase 6 u Oracle como SGBD ya que es una BD que requiere muy poco soporte de mantenimiento pero estos gestores de BD tienen la desventaja de ser comerciales.

GESTIFARMA

Gestifarma es un programa de solución informática para la gestión integral de las oficinas de farmacia. El programa está dividido en varios módulos. Cada uno de ellos hace referencia a una actividad habitual de la Oficina de Farmacia. [22]

Los módulos se encuentran integrados en una misma BD de manera que el usuario no tiene por qué exportar o importar datos de un apartado del programa a otro para obtener un resultado coordinado. El SGBD utilizado es MySQL que es un SGBD libre pero este tiene ciertas desventajas en comparación con otros SGBD como PostgreSQL.

CARE 2X

Es una plataforma informática para la gestión de hospitales hecha bajo licencia GPL. Este sistema de gestión hospitalaria consta de varios módulos, entre estos un módulo encargado específicamente de la gestión farmacéutica. El sistema cuenta con las necesidades fundamentales para el manejo farmacéutico pero dista de algunos elementos necesarios que no se ajustan a toda la realidad de las farmacias cubanas.

Care2x CDS es un servidor central de BD para el completo entorno integrado de cuidados de la salud. Este es un depósito central de datos que puede ser usado en todas las organizaciones de cuidado de la salud participantes Este servidor central es accedido remotamente vía el protocolo de intercambio de datos de la salud (HXP). Este sistema utiliza los SGBD MySQL y PostgreSQL. [23]

GALEN FARMACIA

Es el Sistema de Control de Materiales Hospitalarios (SCMH) para el control de Medicamentos y Materiales Médicos, este es un sistema automatizado, para el control de los movimientos de entradas y salidas que se realizan entre los diferentes sectores que integran la Unidad de Manipulación y Almacenamiento de Medicamentos y Materiales Médicos, y entre éstos y las distintas Entidades Externas a esta Unidad, que pueden ser Centros de Costo, Unidades Externas y Proveedores.

Además, permite la elaboración de pedidos, realizar inventarios, controlar existencias de productos, obtener reportes para la consolidación de la información y otras operaciones.

Este sistema utiliza el SGBD SQLServer que tiene la característica de ser un SGBD comercial. Para la construcción de la BD se utilizó el SGBD relacional SQL Server 7.0, a partir del cual se realizaron consultas y procedimientos

BIOCOM

Es un sistema modular integrado para prestadores de servicios de salud destinado a la gestión de compras, almacén, distribución de medicación a planta y dispensación en forma de unidosis.

BIOCOM es un producto bastante completo en la gestión hospitalaria con un elevado nivel entre sus sistemas y en especial el farmacéutico es un software desarrollado para otros sistemas hospitalarios que difieren de la salud cubana. El SGBD que utiliza este sistema para almacenar y gestionar todos los datos es Access. [24]

InFarmVisual

Es un producto desarrollado sobre la base de software propietario que tiene entre sus características principales la Dispensación de medicamentos, Atención Farmacéutica, Gestión de Compras, Inventarios entre otros. Utiliza como SGBD MySQL. Este programa no se puede utilizar fuera de España porque este sistema solo se ajusta a las especificaciones del sistema sanitario español. [25]

Conclusiones

La herramienta que se utilizará para el modelado de la BD es CaseStudio por las características y ventajas anteriormente descritas y principalmente por su soporte para trabajar formato de BD PostgreSQL. Para realizar el modelo de objetos o diagrama de clases persistentes se utilizará la Herramienta Case, Rational Rose Enterprise Edition.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

El SGBD que se pretende utilizar en el sistema es PostgreSQL, en su versión Postgre8.2 teniendo en cuenta principalmente, su condición de software multiplataforma y con licencia de uso libre, además de las potencialidades desde el punto de vista de funcionamiento que garantiza con su uso.

Se escoge específicamente la versión PostgreSQL 8.2 ya que tiene características que le otorgan rendimiento mucho mejor que los predecesores (8.1 y hacia atrás) para bases de datos grandes, concurrencia multiusuario entre otros y principalmente porque en su carpeta de instalación incluye la herramienta Slony-I que se utilizará para replicar los datos dentro del hospital. Se utilizará la aplicación EMS SQL Manager for PostgreSQL para la administración y desarrollo del SGBD.

Capítulo 2. Descripción y análisis de la solución propuesta.

En el presente capítulo, se explica la estrategia de integración de la solución propuesta con otros módulos. Se describe y fundamenta la arquitectura utilizada para la BD.

Se analizan los requerimientos funcionales y no funcionales del sistema propuesto. Se realiza el Diagrama de Clases Persistentes y el Diagrama Entidad Relación de la BD así como la descripción de las clases y tablas correspondientes.

2.1 Estrategia de integración de la solución con otros módulos o sistemas.

Para el completo y correcto funcionamiento de la BD se debe integrar con otros módulos debido a que la BD del módulo Gestión Farmacéutica necesita establecer relaciones con tablas de otros módulos como es el caso de configuración. En este, por ejemplo, se encuentra la tabla `cfg_unidad_medida` donde se almacenan todas las unidades de medidas que se gestionan en un hospital, entre ellas las de los diferentes productos farmacéuticos y estos datos son necesarios en la gestión de los diferentes procesos farmacéuticos.

Para obtener mayor seguridad, proteger la información y mantener la integridad de la misma esta integración se realizará a través de llamadas a servicios web o a nivel de negocio de la siguiente forma:

El código dentro un mismo componente utiliza llamadas a métodos o eventos de forma directa. La comunicación entre diferentes componentes se realiza mediante llamadas a servicios web o de forma directa a nivel de negocio, en caso de utilizarse servicios web, la información que es transmitida debe cumplir con los estándares internacionales que hay establecidos para facilitar la integración entre nuevos componentes y otros sistemas hospitalarios.

La BD es accesada de forma directa mediante controladoras y los componentes reusados son integrados mediante interfaces sencillas.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Con el objetivo de proteger la información y mantener la integridad de la misma para obtener la información que se necesite de otros módulos se accede a la BD a través del negocio, para esto se realizan consultas.

2.2 Descripción de la arquitectura y fundamentación.

2.2.1 Modelo de datos relacional.

Para el diseño de la BD_Gehos_far se utilizó el modelo de BD relacional debido a que esta fue la propuesta dada por los arquitectos del proyecto y también por presentar las características y ventajas que a continuación se describen.

El modelo de datos relacional se fundamenta principalmente en la existencia de “relaciones”. Este modelo de datos representa todos los datos en la BD como sencillas tablas de dos dimensiones llamadas relaciones.

Es una relación como si fuese una tabla que está compuesta por registros (filas de una tabla) que representan las tuplas y campos que representan las columnas de una tabla. La información puede ser recuperada o almacenada mediante consultas que ofrecen flexibilidad para administrar la información.

Resumiendo lo anteriormente descrito una BD relacional es un conjunto de dos o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave. A esta manera de construir bases de datos se le denomina modelo relacional. [26]

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

2.2.2 Arquitectura Centralizada.

La arquitectura que se utiliza para la BD es una Arquitectura Centralizada ya que la BD reside en conjunto con las restantes BD del proyecto en un mismo y único servidor proporcionando de esta forma una ubicación central para el almacenamiento de datos, la mayoría de las funciones se llevan a cabo más fácilmente.

En el servidor se encuentran las BD modularizadas, cada módulo tiene su BD por separadas. Esto significa que cada uno de los módulos que constituyen el sistema completo, incluyendo Gestión Farmacéutica, tengan su propia BD, en la que pueden establecer un mejor control de la información. De la BD existirá una copia en el servidor central y un administrador se encargará de gestionar toda la información a nivel central.

Para los usuarios acceder a la BD deben conectarse desde cada uno de los puestos de trabajos al servidor central donde esta se encuentra ubicada. Si es necesario modificar, adicionar o eliminar alguna función, vista, dominio o realizar cualquier otro cambio se tiene que acceder desde cada puesto de trabajo al servidor y es ahí donde se realiza la operación que se desee llevar a cabo.

En este servidor central también se encuentran la BD del módulo Configuración que contiene las tablas nomencladores que deben ser utilizadas por los restantes módulos.

Ventajas que trae consigo una BD Centralizada.

- Requiere poco personal de administración.
- Siempre actualizada.

Desventajas que trae consigo una BD Centralizada

- Grandes volúmenes de información (afectan el tiempo de respuesta).
- Dependencia de los sistemas de comunicación.
- Requiere de servidores de alto rendimiento, más costosos, más difícil de reemplazar en caso de fallos.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

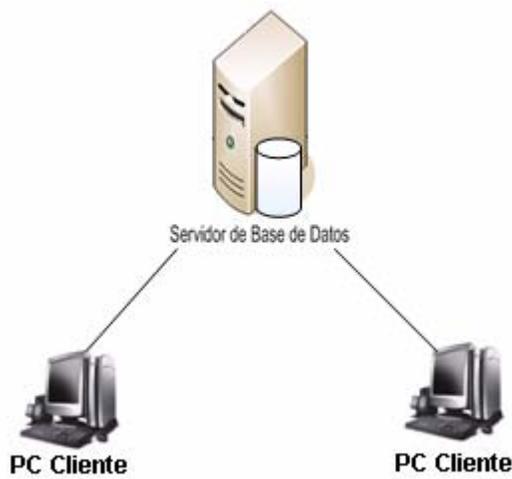


Figura 1. Arquitectura centralizada de la base de datos.

Para un futuro se propone obtener una BD distribuida con el objetivo de que cada módulo del proyecto tenga su propio sitio para la BD y se pueda realizar operaciones en esta sin ser necesario acceder a un servidor central.

Las BD distribuidas consisten precisamente en una especie de objeto virtual, cuyas partes componentes se almacenan físicamente en varias bases de datos "reales" distintas ubicadas en diferentes sitios. De hecho, es la unión lógica de esas bases de datos.

En otras palabras, cada sitio tiene sus propias BD "reales" locales, sus programas para la administración de transacciones y su propio administrador local de comunicación de datos. En particular un usuario dado puede realizar operaciones sobre los datos en su propio sitio local exactamente como si ese sitio no participara en absoluto en el sistema distribuido.

2.3 Especificación de Requerimientos del sistema.

La primera etapa de cualquier proceso de negocio comienza por la tarea de la identificación de los requisitos, para el logro de un mejor diseño, modelación y por ende un resultado exitoso ante el cliente que al final es el objetivo fundamental.

2.3.1 Requisitos funcionales.

Con la definición de los requisitos se pretende establecer un común entendimiento con el cliente sobre los objetivos del diseño de BD propuesto, reflejan todo lo que la BD debe hacer. A continuación se enumeran los requisitos funcionales, capacidades o condiciones que el sistema debe cumplir:

RF 1- Adicionar un producto.

El sistema de BD debe permitir insertar, almacenar, modificar así como mostrar los datos referentes a las características de un nuevo producto que no haya sido insertado anteriormente.

RF-2 Controlar producto.

El sistema de BD debe permitir almacenar, modificar, mostrar y eliminar los datos que se registran en las diferentes tarjetas de control de los productos farmacéuticos.

RF-3 Registrar Solicitudes.

El sistema de BD debe permitir almacenar los datos referentes a las solicitudes de productos.

RF-4 Registra Consumos.

El sistema de BD debe permitir insertar y almacenar los consumos por pacientes y servicios.

RF 5- Mostrar Datos de la Tarjeta de Control de cada medicamento.

El sistema de BD debe permitir mostrar los datos almacenados de la tarjeta de control de medicamento.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

RF 6- Eliminar Tarjeta de Control.

El sistema de BD debe permitir eliminar la tarjeta de control de medicamento que ya haya sido almacenada cuando este se agote o se venza.

RF 7-Actualizar tarjeta de control de medicamentos con déficit en la farmacia.

El sistema de BD debe permitir actualizar la tarjeta de control almacenadas de medicamentos con déficit en la farmacia a medida que se va registrando la carencia de los medicamentos.

RF 8- Actualizar el Libro de Fechas de Vencimiento.

El sistema de BD debe permitir actualizar las fechas de vencimientos almacenadas en el Libro de Fechas de Vencimiento.

RF 9- Buscar y seleccionar Medicamentos.

El sistema de BD debe permitir realizar búsqueda y selección de Medicamentos.

RF 10- Introducir datos de medicamentos elaborados.

El sistema de BD debe permitir insertar y almacenar nuevos medicamentos elaborados por la farmacia.

RF 11- Recepcionar la planilla de solicitud de medicamentos de cada sector del hospital de forma manual.

El sistema de BD debe permitir almacenar las tarjetas de solicitud de medicamentos de cada sector del hospital para tener constancia de los pedidos hechos a la farmacia.

RF 12- Insertar Plantilla de entrega de medicamentos de proveedores.

El sistema de BD debe permitir Insertar y almacenar la Plantilla de entrega de medicamentos de proveedores a la farmacia.

RF 13- Introducir nuevos medicamentos.

El sistema de BD debe permitir Introducir y almacenar los nuevos medicamentos entregados a la farmacia.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

RF 14- Introducir nuevas tarjeta de Control de medicamento.

El sistema de BD debe permitir introducir y almacenar nuevas tarjetas de control de medicamento cuando el medicamento entra a la farmacia.

RF 15- Completar la planilla de devolución de medicamentos.

El sistema de BD debe permitir almacenar los datos de una planilla de devolución de medicamentos que se reintegran por cada sector del hospital con el objetivo de darle entrada a esas devoluciones.

RF 16- Alertar ante la falta de medicamentos, cantidad mínima, vencida o próxima a vencer.

El sistema de BD debe ser capaz de Alertar ante la falta de medicamentos, cantidad mínima, vencida o próxima a vencer para generar los pedidos.

RF 17- Mostrar tarjeta de control de medicamentos con déficit.

El sistema de BD debe permitir mostrar los datos de la tarjeta de control de medicamentos con déficit.

RF 18- Crear plantilla de solicitud de pedidos de medicamentos para la farmacia.

El sistema de BD debe permitir almacenar los datos que contiene una plantilla de solicitud de pedidos de medicamentos para la farmacia.

RF 19- Actualizar plantilla de solicitud de pedidos de medicamentos.

El sistema de BD debe permitir modificar los datos contenidos en las plantilla de solicitud de pedidos de medicamentos que ya fueron entregados.

RF 20- Archivar los Pedidos que no han sido atendidos por los proveedores.

El sistema de BD debe permitir almacenar los datos de los Pedidos que no han sido atendidos por los proveedores.

RF 21- Actualizar resumen de operaciones.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

El sistema de BD debe permitir modificar los datos almacenados en el resumen de operaciones que se genera diario.

RF-22 Eliminar Medicamentos.

EL sistema de BD debe permitir eliminar los datos almacenados de los medicamentos que sean dado de baja en la Farmacia.

RF-23 Actualizar Medicamentos.

El sistema de BD debe de permitir de modificar cualquier dato de los medicamentos existentes en la farmacia.

2.3.2 Requisitos no funcionales.

Los requisitos no funcionales se refieren a las propiedades o cualidades de la BD. Estos responden a cualidades que la BD debe tener y las características para que esta sea confiable, usable y segura.

RNF 1- Requerimientos de rendimiento

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

RNF 2- Requerimientos de soporte

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

RNF 3-Requerimientos políticos, culturales y legales.

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

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

RNF 4-Requerimientos de confiabilidad

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

RNF 5-Requerimientos de hardware

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

Periféricos: Mouse y Teclado.

Tarjeta de Red.

1GB MB de RAM

Capacidad de Almacenamiento que garantice soporte para la BD.

RNF 6-Requerimientos de software

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

PostgreSQL 8.2, Slony-I 1.2.9

RNF 7-Restricciones en el diseño y la implementación

Se utilizará Case Studio para el diseño y modelado de la BD.

La BD será implementada utilizando como SGBD PostgreSQL 8.2.

La comunicación de de las terminales clientes con el servidor será a través de conexiones de fibra óptica.

RNF 8-Requisitos de Independencia:

Hay que obtener un buen diseño de la BD para lograr evitar la aparición de información repetida o redundante por lo que seria muy factible lograr una redundancia nula.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

RNF 9-Requisitos de Consistencia:

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

RNF 10-Requisitos de Seguridad y Privacidad

La información almacenada en la BD, tendrá un gran valor por lo que se debe garantizar que esta información almacenada; se encuentre asegurada frente a usuarios malintencionados, que intenten leer información privilegiada, frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado. Se deben restringir las funcionalidades mediante roles de usuarios garantizando que la información sea accesible al usuario autorizado.

Los usuarios de la BD solamente tendrán acceso a las tablas y BD a las cuales se les designe según el rol que desempeña, no se utilizarán usuarios con privilegios administrativos para realizar las conexiones entre servidores.

Se registrarán y auditarán las trazas dejadas por los usuarios al realizar las diferentes operaciones en el sistema. La programación de las auditorías será programada según corresponda.

RNF 11-Requisitos de Integridad.

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

RNF 12-Requisitos de Respaldo y Recuperación.

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

RNF 13-Requisitos de Tiempo de respuesta.

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

2.4 Modelo de objetos o diagrama de clases persistentes.

El diseño de la BD, se basa en el diagrama de clases persistentes, este a su vez, en los diagramas de clases de diseño.

Los datos reflejados en las clases entidad son los que se capturan y almacenan durante el proceso que describe la aplicación, y estas a su vez son las que componen el diagrama de clases persistentes, del cual se deriva el modelo de datos.

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

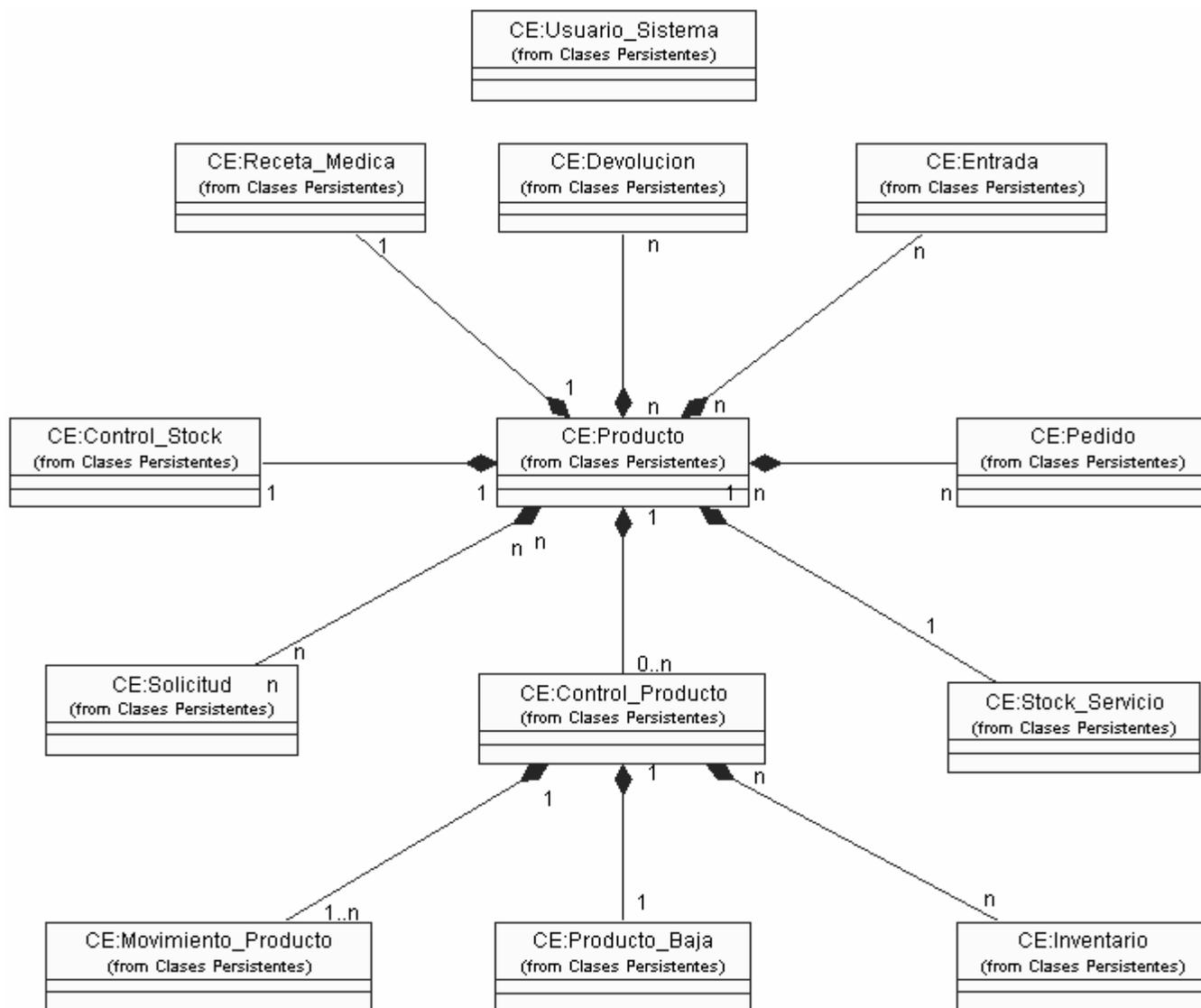


Figura 2. Diagrama de clases persistentes.

2.5 Descripción de las clases persistentes.

Tabla 1. Descripción de la entidad Producto.

Nombre: Producto	
Tipo de clase: Entidad	
Atributo	Tipo
grupo	string
subgrupo	string
nombre	string
codigo_producto	varchar
unidad_medida	integer
presentacion	integer
concentracion	integer
via_administracion	integer
forma_farmaceutica	integer
forma_almacenamiento	integer
estado	integer
observaciones	varchar
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción:	Este es el destructor de la clase.

Tabla 2. Descripción de la entidad Movimiento_Producto.

Nombre: Movimiento_Producto.	
Tipo de clase: Entidad	
Atributo	Tipo
No_movimiento	integer
Fecha_movimiento	date
Usuario_distribuye	string
Usuario_recibe	string
Cantidad	integer
Tipo_movimiento	integer
observaciones	string
Saldo	Integer
Codigo	string
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción:	Este es el destructor de la clase.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 3. Descripción de la entidad Control_Producto.

Nombre: Control_Producto	
Tipo de clase: Entidad	
Atributo	Tipo
Fecha_vencimiento	Date
Numero_Estiba	Integer
Seccion	String
Estante	String
Existencia	Integer
Fecha_recibo	Date
Id_control	Integer
Sector	Integer
Fuente_Recurso	Integer
Observaciones	String
Lote_Entrada	String
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

Tabla 4. Descripción de la entidad Control_Stock.

Nombre: Control_Stock	
Tipo de clase: Entidad	
Atributo	Tipo
Stock_maximo	Integer
Stock_minimo	Integer
Stock_Seguridad	Integer
Stock_Max_Tentativo	Integer
Dias_Stock	Integer
Lote_Reposicion	String
Existencia_Total	Integer
Reserva	Integer
Doble_Reserva	Integer
Fecha_Inicio_Consumo	Date
Mayor_consumo	Integer
Menor_consumo	Integer
Consumo_Farmacia	Integer
Mes_Mayor_Consumo	Integer
Mes_Menor_Consumo	Integer
Codigo_Producto	String
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 5. Descripción de la entidad Pedido.

Nombre: Pedido	
Tipo de clase: Entidad	
Atributo	Tipo
No_Pedido	Integer
Fecha	Date
Proveedor	String
Codigo	String
Farmacia	Integer
Usuario	String
Estado	Integer
Cantidad_pedida	Integer
Cantidad_servida	Integer
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

Tabla 6. Descripción de la entidad Inventario.

Nombre: Inventario	
Tipo de clase: Entidad	
Atributo	Tipo
Fecha	Date
No_Inventario	Integer
Nombre_Inventario	String
Codigo_Producto	String
Nombre_Producto	String
Existencia_Fisica	Integer
Existencia_Registrada	Integer
Auditoria	String
Observaciones	String
Id_Control	Integer
Farmacia	Integer
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 7. Descripción de la entidad Entrada.

Nombre: Entrada	
Tipo de clase: Entidad	
Atributo	Tipo
Fecha_Entrada	Date
No_Documento	Integer
Id_Documento	Integer
Documento	String
Origen	String
Origen_seleccionado	String
Tipo_Movimiento	Integer
Usuario	Integer
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

Tabla 8. Descripción de la entidad Solicitud.

Nombre: Solicitud	
Tipo de clase: Entidad	
Atributo	Tipo
Fecha	Date
No_solicitud	Integer
Servicio	String
Nombre_producto	String
Codigo	String
Cantidad_solicitada	Integer
Cantidad_dispensada	Integer
Estado	String
Solicitante	String
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 9. Descripción de la entidad Devolucion.

Nombre: Devolucion	
Tipo de clase: Entidad	
Atributo	Tipo
Fecha	Date
Motivo	Integer
Servicio	String
Documento	String
Cantidad	Integer
No_Devolucion	Integer
Producto	
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

Tabla 10. Descripción de la entidad Receta_Medica.

Nombre: Receta_Medica	
Tipo de clase: Entidad	
Atributo	Tipo
No.prescripcion	Integer
Nombre	String
Producto	String
Servicio	String
Fecha_pedido	Date
Fecha_entrega	Date
Medico	String
Cantidad	Integer
Dosis	String
Para cada responsabilidad:	
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 11. Descripción de la entidad Stock_Servicio.

Nombre: Stock_Servicio	
Tipo de clase: Entidad	
Atributo	Tipo
Servicio	String
Stock_maximo	Integer
Stock_min	Integer
Stock_Seguridad	Integer
Reserva	Integer
Mayor_consumo	Integer
Menor_consumo	String
Codigo	String
MesMayorConsumo	String
MesMenorConsumo	String
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

Tabla 12. Descripción de la entidad Producto_Baja.

Nombre: Producto_Baja.	
Tipo de clase: Entidad	
Atributo	Tipo
Codigo	String
No_control	String
Motivo	String
Tratamiento	String
Cantidad	Integer
Fecha	Date
MesMenorConsumo	String
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 13. Descripción de la entidad Usuario_Sistema.

Nombre: Usuario_Sistema	
Tipo de clase: Entidad	
Atributo	Tipo
Nombre	String
Password	String
Usuario	Integer
Niuvel_acceso	Integer
Nombre:	Crear()
Descripción:	Este es el constructor de la clase. Crea una nueva instancia de la clase
Nombre:	Destruir()
Descripción	Este es el destructor de la clase.

2.6 Diseño de la Base de Datos.

2.6.1 Diagrama Entidad Relación de la BD.

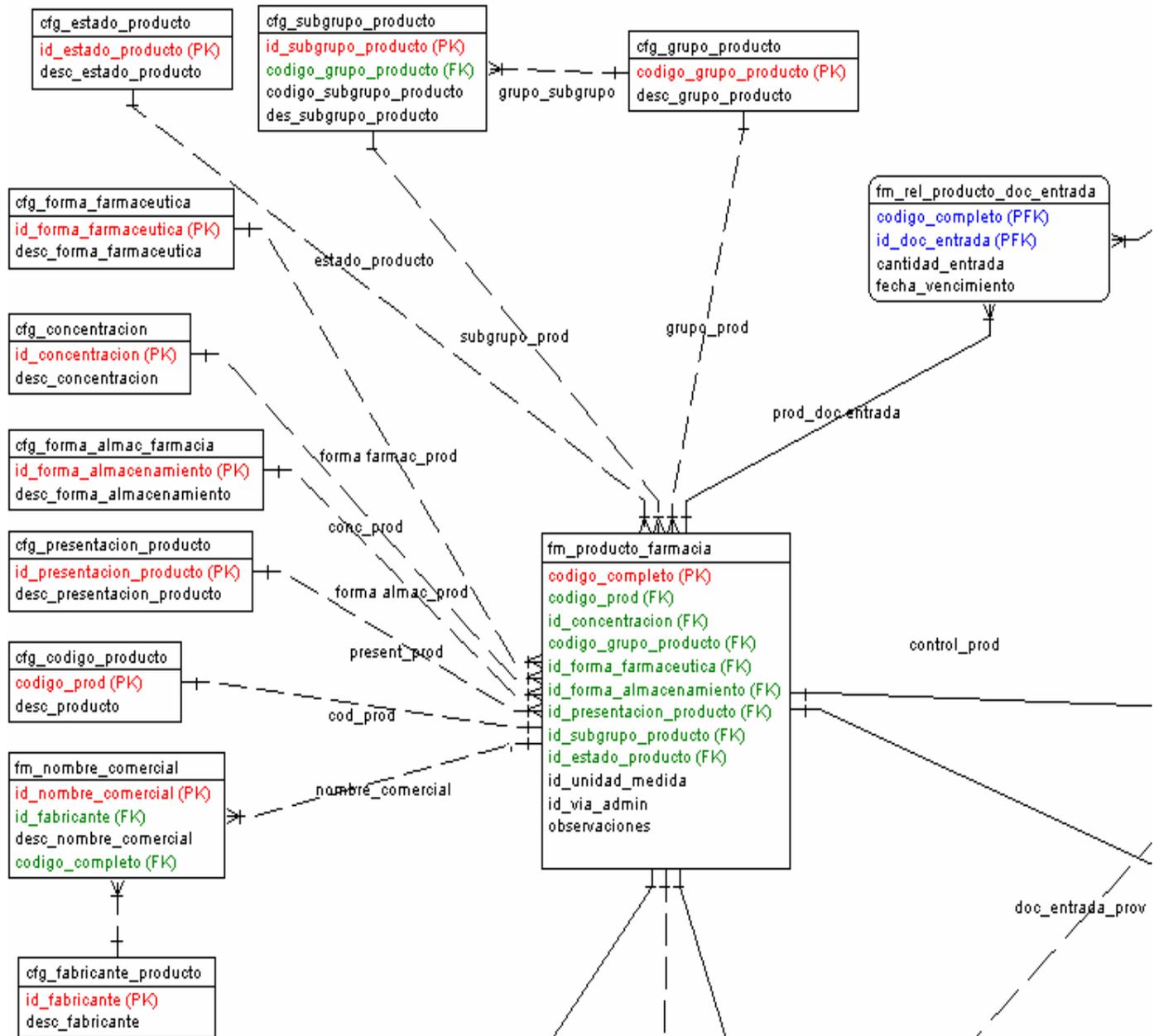


Figura 3. Diagrama Entidad Relación 1-1.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

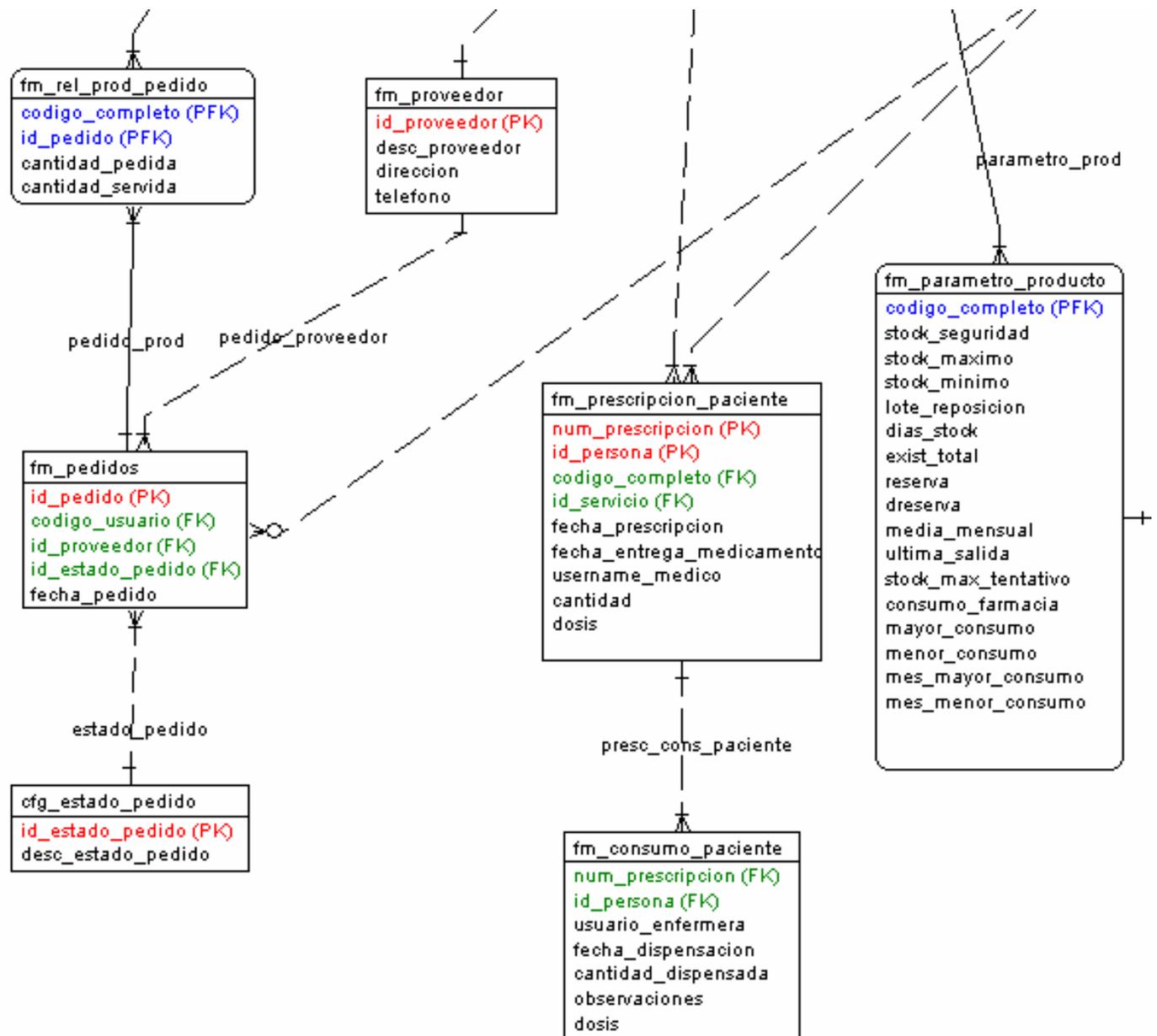


Figura 4. Diagrama Entidad Relación 1-2.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

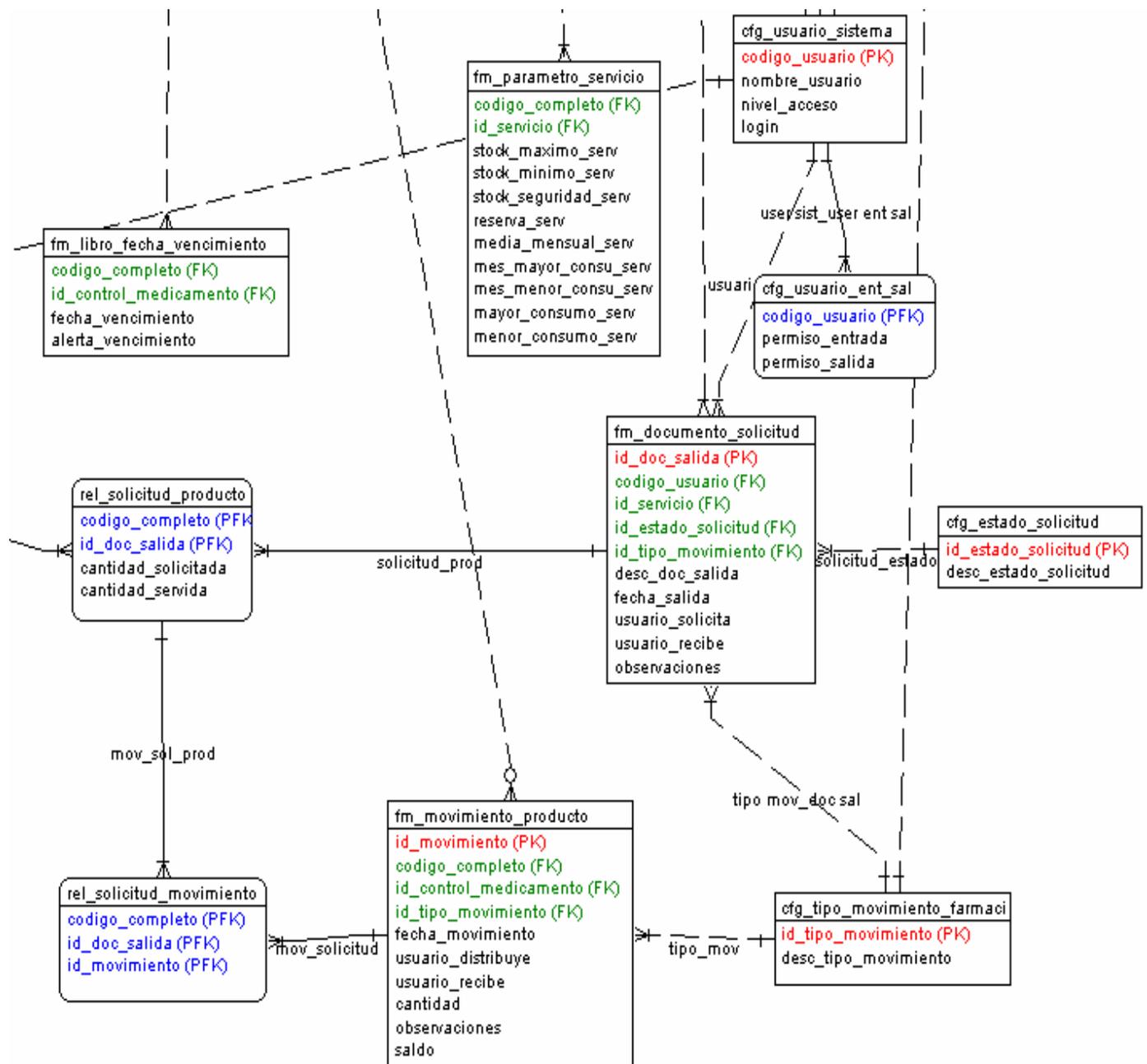


Figura 5. Diagrama Entidad Relación 2-1.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

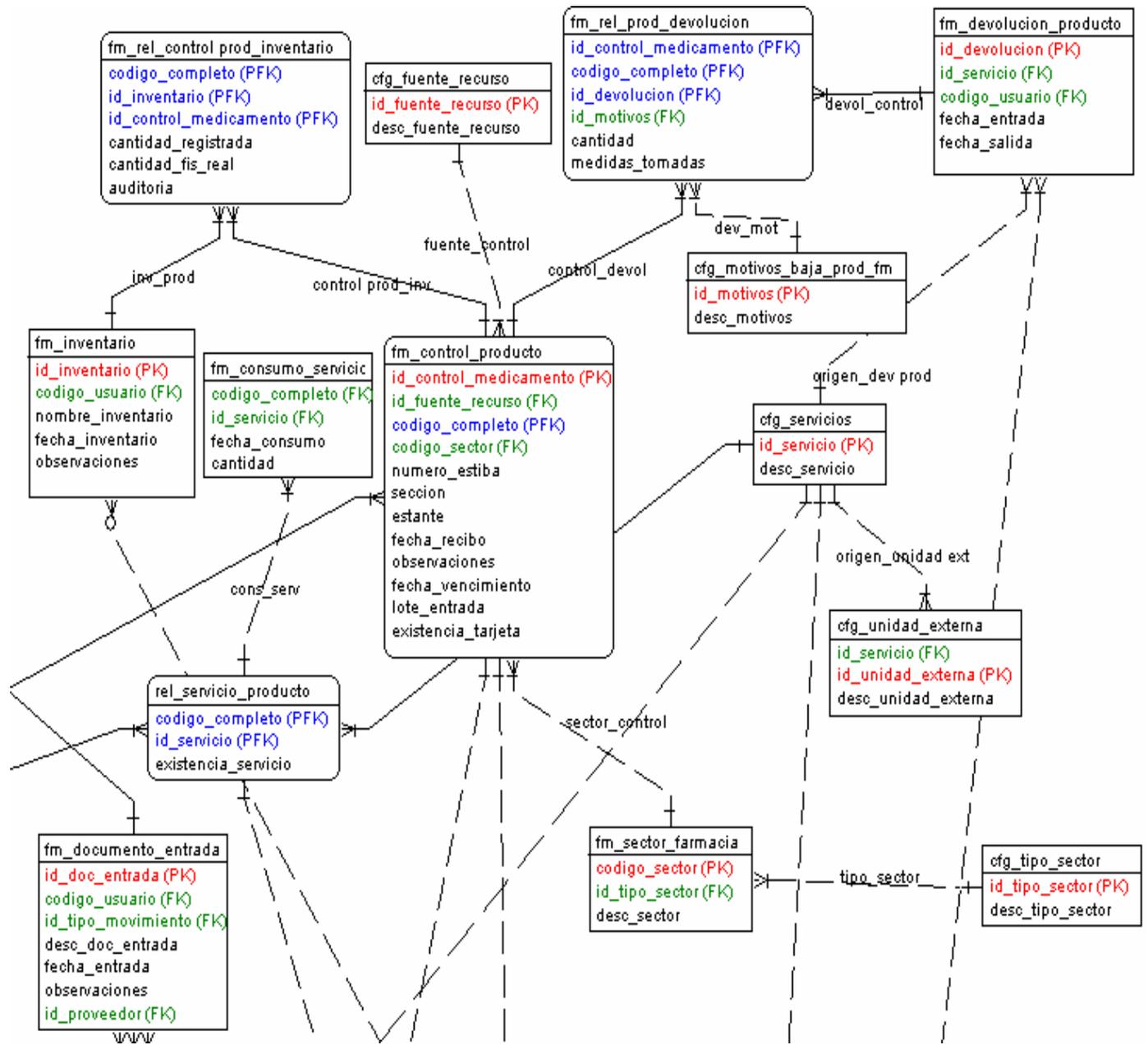


Figura 6. Diagrama Entidad Relación 2-2.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

2.6.2 Descripción de las tablas.

Tabla 14. Detallada descripción de la tabla fm_producto_farmacia.

Nombre: fm_producto_farmacia		
Descripción: contiene los datos referentes a un producto farmacéutico hospitalario incluyendo las características principales de cada uno.		
Atributo	Tipo	Descripción
codigo_completo	varchar	Identificador del producto farmacéutico. Este código esta compuesto por el id del grupo al que pertenece, seguidamente el id del subgrupo y por ultimo el código específico de cada producto. Este código es único para cada producto.
codigo_producto	varchar	Contiene el código específico de cada producto. Este es único para cada producto.
id_estado_producto	integer	Identificador del estado en que se encuentra el producto, puede ser activo o inactivo.
nombre_producto	varchar	Nombre genérico del producto.
observaciones	varchar	Datos opcionales que se deseen agregar.
id_via_admin	integer	Identificador de la vía por la que se debe administrar el medicamento al paciente, esta puede ser oral, intravenosa, etc.
id_unidad_medida	integer	Identificador de la unidad de medida del producto.
id_presentacion	integer	Identificador de la forma de presentación del producto, puede ser en tabletas, capsulas, ámpulas, etc.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

id_subgrupo_producto	integer	Identificador del subgrupo al que pertenece el producto.
id_grupo_producto	integer	Identificador del grupo al que pertenece el producto.
id_forma_almacenamiento	integer	Identificador de la forma en que se almacena el producto. Puede ser en el ambiente, en refrigeración, etc.
id_forma_farmaceutica	integer	Identificador de la forma farmacéutica que tiene el producto. Esta puede ser tabletas, capsulas, etc.

Tabla 15. Detallada descripción de la tabla fm_control_productos.

Nombre: fm_control_productos		
Descripción: contiene los datos referentes al control que se lleva de los productos en la farmacia.		
Atributo	Tipo	Descripción
id_control_medicamento	integer	Identificador del control que se va a tener sobre los productos farmacéuticos. Esta es la llave de la tabla.
numero_estiba	integer	Número de Estiba.
seccion	varchar	Sección en la que se encuentra ubicado el producto.
estante	integer	Estante en que se encuentra ubicado el producto.
observaciones	varchar	Breve descripción que se desee hacer sobre el producto.
existencia_tarjeta	integer	Cantidad total del producto que queda en existencia después de dar entrada o salida del mismo.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

fecha_recibo	timestamp with time zone	Fecha en que se le da entrada al producto.
codigo_completo	varchar	Identificador del producto farmacéutico.
fecha_vencimiento	timestamp with time zone	Fecha de vencimiento que tiene el producto.
lote_entrada	varchar	Lote de entrada que presenta el producto.
id_fuente_recurso	integer	Identificador de la fuente recurso del movimiento del producto.
codigo_sector	varchar	Código del sector para el cual se le da salida al producto o desde el cual proviene el producto.

Tabla 16. Detallada descripción de la tabla fm_proveedor.

Nombre: fm_proveedor		
Descripción: contiene los datos de la entidad que provee los productos al realizar un pedido.		
Atributo	Tipo	Descripción
codigo_proveedor	varchar	Código de la entidad que provee los productos que son entregados en los pedidos que se efectúen. Esta es la llave de la tabla.
desc_proveedor	varchar	Descripción de la persona que provee los productos en los distintos pedidos que se efectúen.
direccion	varchar	Dirección del proveedor. Este dato es opcional.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 17. Detallada descripción de la tabla fm_pedidos.

Nombre: fm_pedidos		
Descripción: contiene los datos que son necesarios registrar cuando se realiza un pedido.		
Atributo	Tipo	Descripción
id _ pedido	integer	Identificador de pedido. Esta es la llave de la tabla.
id _ proveedor	Integer	Identificador de la entidad a la que se le hace el pedido.
codigo_usuario	Varchar	Identificador el usuario que realizara el pedido.
fecha_pedido	timestamp with time zone	Fecha en que se realiza el pedido.
id_estado_pedido	integer	Identificador del estado que tiene el pedido, este puede ser: cancelado, solicitado, pendiente, entre otros.

Tabla 18. Detallada descripción de la tabla fm_movimiento_producto.

Nombre: fm_movimiento_producto		
Descripción: contiene los datos especificados para cuando se realiza un movimiento a algún producto.		
Atributo	Tipo	Descripción
id _ movimiento	integer	Identificador de movimiento que se realice al producto farmacéutico.
fecha _ movimiento	timestamp with time zone	Fecha en que se realiza el movimiento.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

codigo_completo	varchar	Identificador del producto que será movido.
cantidad	integer	Cantidad que ha sido movida del producto.
saldo	integer	Cantidad del producto que queda en existencia después que se realice el movimiento.
id_tipo_movimiento	varchar	Tipo de movimiento que se realiza con los productos, este puede ser entrada de producto, recibir préstamo, devolución de préstamo
id_control_medicamento	integer	Identificador para el control del medicamento.
usuario_distribuye	varchar	Usuario que distribuye el producto.
usuario_recibe	varchar	Usuario que recibe el producto.
observaciones	varchar	Datos adicionales del movimiento que se quieran registrar o señalar.

Tabla 19. Detallada descripción de la tabla fm_libro_fecha_vencimiento.

Nombre: fm_libro_fecha_vencimiento		
Descripción: contiene los datos referentes al vencimiento de cada uno de los productos.		
Atributo	Tipo	Descripción
fecha_vencimiento	timestamp with time zone	Fecha de vencimiento del producto.
alerta_vencimiento	integer	Alerta que mostrará el sistema ante la proximidad de la fecha de vencimiento del producto.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

id_control_medicamento	integer	Identificador para el control del medicamento.
codigo_completo	varchar	Identificador del producto.

Tabla 20. Detallada descripción de la tabla fm_inventario.

Nombre: fm_inventario		
Descripción: esta tabla contiene los datos que son necesarios archivar cuando se realiza inventario.		
Atributo	Tipo	Descripción
id_inventario	integer	Identificador del inventario que se realiza. Esta es la llave de la tabla.
codigo_usuario	varchar	Nombre de la persona que realiza el inventario.
nombre _ inventario	varchar	Nombre que se le da al inventario para identificarlo. Ej. Materiales quirúrgicos no desechables.
observaciones	varchar	Datos adicionales que se quieran señalar acerca del inventario.
fecha_inventario	timestamp with time zone	Fecha en que se realiza el inventario.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 21. Detallada descripción de la tabla fm_sector_farmacia.

Nombre: fm_sector_farmacia		
Descripción: esta tabla contiene los datos de cada sector farmacia.		
Atributo	Tipo	Descripción
codigo_sector	varchar	Código del sector. Esta es la llave de la tabla.
desc_sector	varchar	Descripción del sector.
id_tipo_sector	integer	Identificador del tipo de sector.

Tabla 22. Detallada descripción de la tabla fm_documento_entrada.

Nombre: fm_documento_entrada		
Descripción: esta tabla contiene los datos referentes a la entrada de productos a la farmacia.		
Atributo	Tipo	Descripción
id_doc_entrada	serial	Identificador de la entrada. Es la llave de la tabla.
desc_doc_entrada	varchar	Descripción de la entrada del producto.
codigo_usuario	varchar	Identificador del usuario que realizara la entrada del producto.
id_proveedor	integer	Origen de donde proviene el producto al cual se le dará entrada.
id_tipo_movimiento	integer	Identificador del tipo de movimiento que se le realizara al producto.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

fecha_entrada	timestamp with time zone	Fecha en que se le da entrada al producto.
observaciones	varchar	Datos adicionales que se deseen agregar.

Tabla 23. Detallada descripción de la tabla fm_documento_solicitud.

Nombre: fm_documento_solicitud		
Descripción: esta tabla contiene los datos referentes a las solicitudes realizadas a la farmacia.		
Atributo	Tipo	Descripción
id_doc_salida	serial	Identificador del documento de salida. Es la llave de la tabla.
desc_doc_salida	varchar	Descripción de la salida del producto.
codigo_usuario	varchar	Identificador del usuario que realizara la salida del producto.
id_servicio	integer	Lugar hacia donde se le dará salida al producto.
id_tipo_movimiento	integer	Tipo de movimiento que tendrá el producto.
fecha_salida	timestamp with time zone	Fecha en que se le da salida al producto.
usuario _ recibe	varchar	Usuario que distribuye el producto.
usuario _ distribuye	varchar	Usuario que recibe el producto.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

observaciones	varchar	Datos adicionales que se deseen agregar.
id_estado_solicitud	integer	Estado en que se encuentra la solicitud.

Tabla 24. Detallada descripción de la tabla fm_nombre_comercial.

Nombre: fm_nombre_comercial		
Descripción: esta tabla contiene los datos referentes al nombre comercial de cada producto farmacéutico.		
Atributo	Tipo	Descripción
id_nombre_comercial	integer	Identificador del nombre comercial del producto. Es la llave de la tabla.
nombre_comercial	varchar	Nombre comercial del producto.
id_fabricante	integer	Identificador del fabricante del producto.
codigo_completo	varchar	Este contiene el código del producto el cual es el identificador del mismo.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 25. Detallada descripción de la tabla fm_consumo_paciente.

Nombre: fm_consumo_paciente		
Descripción: esta tabla contiene los datos referentes al consumo que tiene cada paciente del producto farmacéutico.		
Atributo	Tipo	Descripción
num_prescripcion	integer	Número que tiene la prescripción realizada.
id_persona	Integer	Identificador de la persona que recibe la dispensación.
usuario_enfermera	varchar	Enfermera que realiza la dispensación.
fecha_dispensacion	timestamp with time zone	Fecha en que fue realizada la dispensación.
cantidad_dispensada	integer	Cantidad del producto que fue dispensada.
dosis	timestamp with time zone	Dosis que se le da del producto al paciente.
observaciones	varchar	Datos adicionales que se deseen agregar.

Tabla 26. Detallada descripción de la tabla fm_consumo_servicio.

Nombre: fm_consumo_servicio		
Descripción: esta tabla contiene los datos referentes al consumo de los productos farmacéuticos que tiene cada servicio.		
Atributo	Tipo	Descripción
id_servicio	integer	Identificador del servicio que realiza el consumo.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

codigo_completo	varchar	Código del producto consumido.
cantidad	integer	Datos adicionales que se quieran señalar acerca del consumo del servicio.
fecha_consumo	timestamp with time zone	Fecha en que el servicio realiza el consumo del producto.

Tabla 27. Detallada descripción de la tabla fm_prescripcion_paciente.

Nombre: fm_prescripcion_paciente		
Descripción: esta tabla contiene los datos referentes a la prescripción de productos farmacéuticos que tiene cada paciente.		
Atributo	Tipo	Descripción
num_prescripcion	integer	Número de prescripción.
id_persona	integer	Identificador del paciente al cual se le realiza la prescripción.
codigo_completo	varchar	Código del producto que es dispensado.
id_servicio	integer	Identificador del servicio.
fecha_prescripcion	timestamp with time zone	Fecha en que se realiza la prescripción.
fecha_entrega_medicamento	timestamp with time zone	Fecha en que se realiza la prescripción del medicamento.
username_medico	varchar	Médico que realiza la prescripción al paciente.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

cantidad	integer	Cantidad del producto que es dispensado.
dosis	varchar	Dosis que se le da al paciente.

Tabla 28. Detallada descripción de la tabla fm_parametro_producto.

Nombre: fm_parametro_producto		
Descripción: esta tabla contiene los datos referentes al control de stock que se tiene sobre cada producto farmacéutico.		
Atributo	Tipo	Descripción
codigo_completo	varchar	Código del producto farmacéutico.
stock_maximo	Double precision	Cantidad máxima del producto que queda en el stock.
stock_minimo	Double precision	Cantidad mínima del producto que queda en el stock.
stock_seguridad	Double precision	Cantidad del producto que hay en stock para seguridad.
lote_reposicion	varchar	Lote que tiene los productos que se utilizan para reposición.
dias_stock	Integer	Días de stock que tiene el producto.
exist_total	Double precision	Existencia total que existe del producto farmacéutico.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

reserva	Double precision	Cantidad del producto que queda como reserva.
dreserva	Double precision	Cantidad del producto que queda para doble reserva.
media_mensual	Double precision	Media mensual del producto, es decir, promedio de la media del consumo del producto.
consumo_farmacia	Double precision	Consumo del producto que tiene la farmacia.
mes_mayor_consumo	timestamp with time zone	Mes en el que se realiza mayor consumo del producto farmacéutico.
mes_menor_consumo	timestamp with time zone	Mes en el que se realiza menor consumo del producto farmacéutico.
mayor_consumo	Double precision	Mayor consumo realizado durante la fecha seleccionada.
menor_consumo	Double precision	Menor consumo realizado durante la fecha seleccionada.

Tabla 29. Detallada descripción de la tabla fm_devolucion_producto.

Nombre: fm_devolucion_producto		
Descripción: esta tabla contiene los datos de los productos al realizarse una devolución de los mismos.		
Atributo	Tipo	Descripción
id_devolucion	integer	Identificador de la devolución de un producto. Esta es la llave de la tabla.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

id_servicio	integer	Lugar desde donde viene la devolución del producto.
codigo_usuario	varchar	Identificador del usuario que realizara la devolución del producto.
fecha_entrada	timestamp with time zone	Fecha en que se le dio entrada al producto.
fecha_salida	timestamp with time zone	Motivos por los que se realiza la devolución del producto.

Tabla 30. Detallada descripción de la tabla nomencladora cfg_codigo_producto.

Nombre: cfg_codigo_producto		
Descripción: esta tabla contiene los datos referentes al código de cada producto farmacéutico.		
Atributo	Tipo	Descripción
codigo_producto	varchar	Identificador del producto farmacéutico. Esta es la llave de la tabla.
desc_codigo_producto	varchar	Descripción del producto farmacéutico.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 31. Detallada descripción de la tabla nomencladora cfg_grupo_producto.

Nombre: cfg_grupo_producto		
Descripción: esta tabla contiene los datos referentes a los grupos a los cuales pertenece cada producto farmacéutico.		
Atributo	Tipo	Descripción
id_grupo_producto	serial	Identificador del grupo al cual pertenece el producto farmacéutico. Esta es la llave de la tabla.
desc_grupo_producto	varchar	Descripción del grupo al cual pertenece el producto farmacéutico.

Tabla 32. Detallada descripción de la tabla nomencladora cfg_subgrupo_producto.

Nombre: cfg_subgrupo_producto		
Descripción: esta tabla contiene los datos referentes a los subgrupos a los cuales pertenece cada producto farmacéutico.		
Atributo	Tipo	Descripción
codigo_subgrupo_producto	varchar	Código del subgrupo al cual pertenece el producto farmacéutico.
id_subgrupo_producto	serial	Identificador del subgrupo al cual pertenece el producto farmacéutico.
desc_subgrupo_producto	varchar	Descripción del subgrupo al cual pertenece el producto farmacéutico.
codigo_grupo_producto	varchar	Grupo al cual pertenece el subgrupo.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 33. Detallada descripción de la tabla nomencladora cfg_concentracion.

Nombre: cfg_concentracion		
Descripción: esta tabla contiene los datos referentes a la concentración que presenta el producto farmacéutico.		
Atributo	Tipo	Descripción
id_concentracion	serial	Identificador de la concentración que presenta el producto. Esta es la llave de la tabla.
desc_concentracion	varchar	Descripción de la concentración que presenta el producto farmacéutico.

Tabla 34. Detallada descripción de la tabla nomencladora cfg_forma_almacenamiento.

Nombre: cfg_forma_almacenamiento		
Descripción: esta tabla contiene los datos referentes a la forma de almacenamiento de cada producto.		
Atributo	Tipo	Descripción
id_forma_almacenamiento	serial	Identificador de la forma en que se almacena el producto.
desc_forma_almacenamiento	varchar	Descripción de la forma de almacenamiento que tiene el producto.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 35. Detallada descripción de la tabla nomencladora cfg_forma_almacenamiento.

Nombre: cfg_presentacion_producto		
Descripción: esta tabla contiene los datos referentes a la forma de presentación de cada producto.		
Atributo	Tipo	Descripción
id_presentacion_producto	serial	Identificador que tiene la forma de presentación del producto. Este puede ser en frasco, bolsa, tableta entre otros. Es la llave de la tabla.
desc_presentacion_producto	varchar	Descripción de la forma de presentación que tiene el producto.

Tabla 36. Detallada descripción de la tabla nomencladora cfg_forma_farmaceutica.

Nombre: cfg_forma_farmaceutica		
Descripción: esta tabla contiene los datos referentes a la forma farmacéutica que tiene cada producto.		
Atributo	Tipo	Descripción
id_forma_farmaceutica	serial	Identificador de la forma farmacéutica que tiene el producto, esta puede ser suspensión, crema, jarabe, entre otras. Es la llave de la tabla.
desc_forma_farmaceutica	varchar	Descripción de la forma farmacéutica que tendrá el producto.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 37. Detallada descripción de la tabla nomencladora cfg_forma_farmaceutica.

Nombre: cfg_fuente_recurso		
Descripción: esta tabla contiene los datos referentes a la fuente que utilizara un recurso determinado de la farmacia.		
Atributo	Tipo	Descripción
id_fuente_recurso	integer	Identificador de la fuente que utilizara un recurso determinado. Es la llave de la tabla.
desc_fuente_recurso	varchar	Descripción de la fuente en que se utilizara el recurso. Ej. Donaciones.

Tabla 38. Detallada descripción de la tabla nomencladora cfg_tipo_movimiento_farmacia.

Nombre: cfg_tipo_movimiento_farmacia		
Descripción: esta tabla contiene los datos referentes al movimiento que tendrá el producto.		
Atributo	Tipo	Descripción
id_tipo_movimiento	integer	Identificador del tipo de movimiento que se le realice al producto. Es la llave de la tabla.
desc_tipo_movimiento	varchar	Descripción del tipo de movimiento que se realice al producto. Este puede ser de entrada o salida.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 39. Detallada descripción de la tabla nomencladora cfg_tipo_sector.

Nombre: cfg_tipo_sector		
Descripción: esta tabla contiene los datos referentes a los tipos de sector.		
Atributo	Tipo	Descripción
id_tipo_sector	serial	Identificador del tipo de sector. Es la llave de la tabla.
desc_tipo_sector	varchar	Descripción del tipo de sector.

Tabla 40. Detallada descripción de la tabla nomencladora cfg_unidad_externa.

Nombre: cfg_unidad_externa		
Descripción: esta tabla contiene los datos referentes a las unidades externas de donde proviene alguna acción.		
Atributo	Tipo	Descripción
id_unidad_externa	serial	Identificador de la unidad externa. Esta es la llave de la tabla.
desc_unidad_externa	varchar	Descripción de la unidad externa. Esta puede ser un centro de costo, un almacén farmacéutico, entre otros.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 41. Detallada descripción de la tabla nomencladora cfg_servicio.

Nombre: cfg_servicio		
Descripción: esta tabla contiene los referentes al origen de una acción.		
Atributo	Tipo	Descripción
id_servicio	serial	Identificador del servicio. Esta es la llave de la tabla.
desc_servicio	varchar	Descripción del servicio.

Tabla 42. Detallada descripción de la tabla nomencladora cfg_motivos_baja_prod_fm.

Nombre: cfg_motivos_baja_prod_fm		
Descripción: esta tabla contiene los datos referentes a los motivos por los cuales se les da baja a un producto farmacéutico.		
Atributo	Tipo	Descripción
id_motivos	serial	Identificador de los motivos por los que se realiza la baja del producto. Esta es la llave de la tabla.
desc_motivos	varchar	Descripción de los motivos por los que se realiza la baja.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 43. Detallada descripción de la tabla nomencladora cfg_fabricante_producto.

Nombre: cfg_fabricante_producto		
Descripción: esta tabla contiene los datos referentes al fabricante del producto.		
Atributo	Tipo	Descripción
id_fabricante	serial	Identificador del fabricante del producto.
desc_fabricante	varchar	Descripción del fabricante del producto.

Tabla 44. Detallada descripción de la tabla nomencladora cfg_estado_producto.

Nombre: cfg_estado_producto		
Descripción: esta tabla contiene los datos referentes a los estados que puede tener un producto.		
Atributo	Tipo	Descripción
id_estado_producto	serial	Identificador del estado del producto. Esta es la llave de la tabla.
desc_estado_producto	varchar	Descripción del estado del producto, este puede ser Activo o Inactivo.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 45. Detallada descripción de la tabla nomencladora `cfg_estado_pedido`.

Nombre: <code>cfg_estado_pedido</code>		
Descripción: esta tabla contiene los datos referentes a los estados que puede tener un pedido.		
Atributo	Tipo	Descripción
<code>id_estado_pedido</code>	serial	Identificador del estado del pedido. Esta es la llave de la tabla.
<code>desc_estado _ pedido</code>	varchar	Descripción del estado del pedido.

Tabla 46. Detallada descripción de la tabla nomencladora `cfg_estado_solicitud`.

Nombre: <code>cfg_estado_solicitud</code>		
Descripción: esta tabla contiene los datos referentes a los estados que puede tener una solicitud.		
Atributo	Tipo	Descripción
<code>id_estado_solicitud</code>	serial	Identificador del estado de la solicitud. Esta es la llave de la tabla.
<code>desc_estado _ solicitud</code>	varchar	Descripción del estado de la solicitud.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 47. Detallada descripción de la tabla nomencladora cfg_usuario_ent_sal.

Nombre: cfg_usuario_ent_sal		
Descripción: esta tabla contiene los datos referentes a los permisos que tendrá cada usuario al interactuar con el sistema.		
Atributo	Tipo	Descripción
codigo_usuario	integer	Contiene el identificador del usuario del sistema.
permiso_entrada	boolean	Permiso que tiene el usuario del sistema para dar entrada de productos.
permido_salida	boolean	Permiso que tiene el usuario del sistema para dar salida de productos.

Tabla 48. Detallada descripción de la tabla nomencladora cfg_usuario_sistema.

Nombre: cfg_usuario_sistema		
Descripción: esta tabla contiene los datos referentes a los usuarios que accederán al sistema.		
Atributo	Tipo	Descripción
codigo_usuario	varchar	Identificador de los usuarios del sistema. Esta es la llave de la tabla.
nombre_usuario	varchar	Nombre de los usuarios del sistema.
nivel_acceso	integer	Nivel de acceso en el sistema que tendrá cada usuario.
login	varchar	Contraseña que tendrá cada usuario del sistema para acceder a este.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 49. Detallada descripción de la tabla de relación rel_control_prod_inventario.

Nombre: rel_control_prod_inventario		
Descripción: esta tabla contiene datos necesarios para la realización de inventarios en la farmacia.		
Atributo	Tipo	Descripción
id_inventario	integer	Identificador del inventario.
id_control_medicamento	integer	Identificador del control del producto a realizar inventario.
codigo_completo	varchar	Identificador del producto a realizar inventario.
cantidad_registrada	integer	Cantidad de producto que ha sido registrada en la farmacia a través de la computadora durante las operaciones de Entrada y Salida.
cantidad_fisica_real	integer	Cantidad del producto que realmente existe físicamente. Es el resultado que se obtiene cuando se realiza el Conteo físico del producto en la farmacia.
auditoria	Boolean	Toma valor de ERROR si existen diferencias entre la Cantidad Física Real y la Cantidad Registrada, cuando se audita el inventario.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 50. Detallada descripción de la tabla de relación rel_prod_devolucion.

Nombre: rel_prod_devolucion		
Descripción: esta tabla contiene datos necesarios para realizar devoluciones de productos farmacéuticos.		
Atributo	Tipo	Descripción
id_devolucion	integer	Identificador de la devolución a realizar.
id_control_medicamento	integer	Identificador del control del medicamento.
codigo_completo	varchar	Identificador del producto a realizar la devolución.
id _ motivos	integer	Identificador de los motivos para realizar la devolución.
cantidad	integer	Cantidad del producto que se realizara la devolución.
medidas_tomadas	varchar	Medidas que son tomadas debido a la devolución del producto farmacéutico.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 51. Detallada descripción de la tabla de relación rel_prod_devolucion.

Nombre: rel_prod_pedido		
Descripción: esta tabla contiene datos necesarios para realizar pedido de productos.		
Atributo	Tipo	Descripción
id_pedido	integer	Identificador del pedido.
codigo_completo	varchar	Identificador del producto a realizar pedido.
cantidad_pedida	integer	Cantidad del producto que es pedido.
cantidad_servida	integer	Cantidad del producto que es servida.

Tabla 52. Detallada descripción de la tabla de relación rel_producto_doc_entrada.

Nombre: rel_producto_doc_entrada		
Descripción: esta tabla contiene datos que son necesarios para realizar la entrada de un producto.		
Atributo	Tipo	Descripción
id_doc_entrada	integer	Identificador del documento de entrada del producto farmacéutico.
codigo_completo	varchar	Identificador del producto a darle salida.
cantidad_entrada	integer	Cantidad que se dará entrada del producto.
fecha_vencimiento	Timestamp with time zone	Fecha de vencimiento que tiene el producto al cual se le dará entrada.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Tabla 53. Detallada descripción de la tabla de relación rel_producto_doc_entrada.

Nombre: rel_solicitud_producto		
Descripción: esta tabla contiene datos necesarios para realizar la salida de un producto.		
Atributo	Tipo	Descripción
id_doc_salida	integer	Identificador del documento de salida del producto farmacéutico.
codigo_completo	varchar	Identificador del control del medicamento al que se le realizara la salida.
cantidad_servida	integer	Cantidad del producto que es servida.
cantidad _ solicitada	integer	Cantidad del producto que es solicitada.

Tabla 54. Detallada descripción de la tabla de relación rel_solicitud_movimiento.

Nombre: rel_solicitud_movimiento		
Descripción: esta tabla contiene datos necesarios para realizar la salida de un producto.		
Atributo	Tipo	Descripción
id_doc_salida	integer	Identificador del documento de salida del producto farmacéutico.
codigo_completo	varchar	Identificador del control del medicamento al que se le realizara la salida.
id_movimiento	integer	Identificador del movimiento que tiene el producto

Tabla 55. Detallada descripción de la tabla de relación rel_servicio_producto.

Nombre: rel_servicio_producto		
Descripción: esta tabla contiene datos necesarios para realizar la salida de un producto.		
Atributo	Atributo	Atributo
id_servicio	id_servicio	id_servicio
codigo_completo	codigo_completo	codigo_completo
existencia_producto	existencia_producto	existencia_producto

2.7 Análisis de optimización de consultas.

Para lograr la optimización de una BD es necesario un buen diseño inicial, tanto lógico como físico. El objetivo de la optimización consiste en minimizar el tiempo de respuesta para cada petición y maximizar el rendimiento de todo el sistema disminuyendo el tráfico de red, el acceso a disco y el tiempo de CPU.

Los planteamientos sobre la optimización deben ser propuestos durante todo el ciclo de desarrollo, y no sólo una vez, que el sistema ya esté implementado.

Cuando se desarrolla una aplicación de BD, se debe tener en cuenta no sólo que funcionen las consultas sino que estas tengan la suficiente eficiencia como para que preste un servicio muy rápido. [27]

Para conseguir este objetivo se puede utilizar los métodos siguientes:

Consultas SELECT

En general, cuando se desea hacer que una consulta SELECT... WHERE se ejecute más rápido, lo primero que se debe chequear es si se puede agregar un índice. Todas las referencias entre tablas diferentes deben usualmente ser hechas con índices. Se debe usar una sentencia EXPLAIN para determinar cuáles índices están siendo usados para resolver la consulta.

Los índices permiten un rápido acceso a registros que contienen valores específicos. Son estructuras, se definen para un atributo o conjunto de atributos asociados, que nos permiten simular una secuencia lógica para las entidades. La principal cualidad de un índice reside en la capacidad para acelerar el acceso a un dato específico.

Sentencias INSERT

- Si se está insertando muchas filas desde el mismo cliente al mismo tiempo, se debe utilizar una sentencia INSERT con múltiples listas de valores para insertar varias filas a la vez. Esto es mucho más rápido que usar varias sentencias INSERT de manera separada.

Por ejemplo, la siguiente consulta:

```
INSERT INTO nombreTabla VALUES (registro1),(registro2),... (registroN);
```

es mucho más rápida que esta alternativa:

```
INSERT INTO nombreTabla VALUES (registro1);  
INSERT INTO nombreTabla VALUES (registro2);  
...  
INSERT INTO nombreTabla VALUES (registroN);
```

- Si se está insertando una gran cantidad de filas desde diferentes clientes, se puede obtener una mayor velocidad al usar la sentencia INSERT DELAYED.
- Cuando se esté cargando datos en una tabla a partir de un archivo de texto, lo mejor es usar la sentencia LOAD DATA INFILE. Esto es usualmente 20 veces más rápido que usar una gran cantidad de sentencias INSERT.

Sentencias UPDATE

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Las sentencias UPDATE pueden ser optimizadas de manera similar a las sentencias SELECT con la sobrecarga adicional de la escritura.

Por ejemplo, para efectos de optimización, el siguiente código:

```
UPDATE nombreCampo FROM nombreTabla WHERE Condicion
```

Es el mismo que este:

```
SELECT nombreCampo FROM nombreTabla WHERE Condicion
```

La velocidad de escritura depende de la cantidad de datos que están siendo actualizados y el número de índices que son actualizados, por lo tanto se debe tener cuidado de crear índices que no sean verdaderamente útiles, o bien, hacer que los campos de la tabla sean más grandes de lo que realmente se necesita.

Sentencias DELETE

El tiempo de eliminar registros individuales es exactamente proporcional al número de índices. Cuando se elimina, cada registro necesita ser eliminado desde cualquier índice asociado, así como también del archivo principal de datos.

Si se desean eliminar todos los registros de una fila, es más factible usar el comando TRUNCATE TABLE en lugar de ejecutar la tradicional sentencia DELETE, ya que se borra toda la tabla en una sólo operación, sin la necesidad de eliminar cada índice y cada registro de manera individual.

El rendimiento de una BD también se mide por los tiempos de respuesta de sus transacciones y por el throughput, que es el número de transacciones por unidad de tiempo. Si estos tiempos están fuera del rango esperado es necesario llevar a cabo un proceso de entonación de la BD.

La entonación de una BD relacional se puede dividir en dos componentes fundamentales: la entonación del diseño físico y la entonación de las consultas.

Con respecto al diseño físico, se define la organización primaria de los archivos que van a contener las tablas de una BD tomando en cuenta el uso que se le va a dar, esto se refiere a cuáles transacciones

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

están definidas sobre la BD y cuál es su frecuencia de ejecución. También se toman en consideración las restricciones de tiempo de respuesta de cada transacción.

En esta sección sólo se tratará el análisis de optimización de consultas.

Las actualizaciones de la BD por lo general no tienen en sí mismas, muchos aspectos que se puedan entonar, basta con conocer cuáles actualizaciones tienen lugar en una BD y con cuál frecuencia. Sin embargo, las consultas tienen más variedad de formas y espacio para mejorar el rendimiento.

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

1. La cláusula DISTINCT es costosa de ejecutar porque generalmente involucra un ordenamiento de las tuplas resultantes para eliminar duplicados. Es necesario analizar si realmente hace falta usar esa cláusula.
2. Las subconsultas en muchos manejadores se ejecutan ineficientemente. Una razón para ello es que al estar anidada no se utiliza algún índice relevante, al eliminar la subconsulta es posible lograr que el manejador utilice el índice apropiado.
3. En algunas consultas el usuario almacena resultados intermedios explícitamente en tablas temporales. Estas tablas pueden bajar el rendimiento de la consulta.
4. A pesar de lo anteriormente señalado, en ocasiones las tablas temporales pueden tener efecto positivo al reescribir consultas que contienen subconsultas correlacionadas complejas. Las subconsultas correlacionadas pueden ejecutarse ineficientemente, pero si se logra calcular primero las tuplas de la subconsulta y almacenarlas en una tabla temporal, se puede reescribir la consulta original, sin la subconsulta, accediendo a la tabla temporal.
5. Otro uso beneficioso de las tablas temporales es para evitar el uso de la cláusula ORDER BY que puede ser costosa.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

6. Las condiciones de join se pueden evaluar más eficientemente contra un índice primario. Y en general, en términos de rendimiento es preferible evaluar una condición de igualdad numérica que una condición de igualdad sobre cadenas ("strings") de caracteres.

7. Es preferible no usar la cláusula HAVING si la condición deseada se puede expresar en la cláusula WHERE.

8. Otra particularidad importante del manejador que es bueno conocer es si el orden de las tablas en la cláusula FROM puede afectar la implementación del join utilizada, posiblemente esto es relevante para joins que involucran cinco tablas o más.

9. El uso de vistas pueden causar una ejecución ineficiente de consultas. Muchas veces la ejecución de consultas sobre las tablas base es más eficiente. [28]

Para la realización de las consultas y funciones en la BD_Gehos_far se han tenido en cuenta cada uno de los señalamientos planteados, evitando hacer uso de la cláusula DISTINCT debido a lo costosa que resulta su ejecución, no ha sido realmente necesaria su utilización.

Se debe señalar que ha sido necesaria la utilización de vistas, debido a la necesidad de realizar consultas a varias tablas a vez. Lo más factible en estos casos es hacer la unión de estas tablas en una vista, para luego acceder a la vista desde la función o consulta que se vaya a realizar.

A continuación se describen algunas de las vistas realizadas en la BD_Gehos_far:

- `fm_producto_farmacia_view`: esta vista permite mostrar la descripción de cada una de las características del producto farmacéutico ya que en la tabla `fm_producto_farmacia` solo muestra los identificadores de estas características debido a que las toma como llave foráneas de las tablas de configuración correspondientes.
- `grupo_subgrupo_view`: esta vista permite mostrar a que grupo pertenece cada subgrupo de producto, es decir, muestra el código y la descripción de los grupos y subgrupos existentes en la BD.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

- `fm_movimiento_salida_view`: esta vista permite mostrar el código y nombre de cada producto con la cantidad que se le ha dado salida.
- `fm_relacion_solicitud_producto_view`: esta vista permite mostrar todos los datos que son necesarios registrar de cada producto al darle respuesta a una solicitud determinada.
- `fm_relacion_producto_control_view`: esta viste permite mostrar los datos almacenado en la tabla de control y las características que se registraron del mismo.

Para lograr una eficaz optimización, también se debe tener presente el uso de funciones para el acceso a los datos. Esto optimiza considerablemente el tiempo de respuesta debido a que el procedimiento almacenado es un código precompilado de manera que al ejecutarlo se consume menos tiempo de respuesta. En la `BD_Gehos_far` se implementaron las funciones necesarias para el completo funcionamiento de la misma.

Conclusiones

En este capítulo, se obtuvieron los requisitos funcionales y no funcionales que debe cumplir el sistema de BD. Se realizó el diseño quedando bien detallada la descripción de cada una de las tablas que lo conforman. Además, se explicó la arquitectura que presenta la BD, así como la alternativa que se desarrolla para integrarla con otros módulos del Sistema de Información Hospitalaria.

Capítulo 3. Validación del diseño realizado.

En el presente capítulo, se presenta la validación teórica del diseño realizado. Se realizan el análisis y fundamentación de: la integridad, normalización, seguridad, y redundancia de información en la BD.

3.1 Validación teórica del diseño

La validación teórica del diseño de la BD permite conocer si esta fue diseñada correctamente. Para ello se tienen en cuenta: como se implementa la integridad en la BD, si existe redundancia en la información almacenada, como se implementa la seguridad de la BD, como se realiza la normalización y como se almacena la trazabilidad de las acciones que se llevan a cabo en ella.

3.1.1 Integridad

La integridad de una BD significa que, esta ó los programas que generaron su contenido, incorporen métodos que aseguren que no se rompan ni los datos del sistema, ni las reglas del negocio.

En otras palabras la integridad no es más, que reglas que deben cumplir los datos almacenados para garantizar que sean correctos. [29]

Objetivo de la Integridad:

El objetivo en cuanto a integridad, es proteger la BD contra operaciones que introduzcan inconsistencias, por eso se habla de integridad en el sentido de corrección, validez o precisión de los datos de la BD.

El subsistema de integridad de un SGBD debe, por tanto, detectar y corregir, en la medida de lo posible las operaciones incorrectas.

Cuando los contenidos de una BD se modifican con sentencias INSERT, DELETE O UPDATE, la integridad de los datos almacenados puede perderse de muchas maneras diferentes.

Por ejemplo:

- Pueden añadirse datos no válidos a la BD, tales como un pedido que especifica un producto no existente.
- Pueden modificarse, datos existentes tomando un valor incorrecto, como por ejemplo si se reasigna un movimiento a un producto no existente.
- Los cambios en la BD, pueden perderse debido a un error del sistema o a un fallo en el suministro de energía.
- Los cambios pueden ser aplicados parcialmente, como por ejemplo si se añade un pedido de un producto sin ajustar, la cantidad disponible para darle salida. [30]

Para evitar precisamente que la integridad de los datos almacenados en la BD no se pierda se debe tener en cuenta las reglas de integridad que existen.

Reglas de Integridad en BD.

Los conceptos básicos de integridad, en el modelo relacional son: llave primaria, llave foránea, valores nulos y dos reglas de integridad.

Una llave primaria, es uno o un conjunto de atributos que permiten identificar a las distintas columnas de la tabla de manera única en cualquier momento.

Una llave foránea de una relación, es un atributo que hace *referencia* a una llave primaria de otra relación; esto da pie a que una relación pueda tener varias llaves foráneas.

Un valor nulo, es un valor que está fuera de la definición de cualquier dominio, el cual permite dejar el valor del atributo *``latente''*.

- *Integridad de entidad*: Establece que ningún componente de la llave primaria de una relación puede aceptar valores nulos y que esta debe tener un valor único para cada fila de la tabla, sino la BD perderá su integridad.

CAPÍTULO 3. VALIDACIÓN DEL DISEÑO REALIZADO

- *Integridad referencial:* la BD no debe contener valores de llaves ajenas sin concordancia. Si en una relación hay alguna llave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos.

Es de gran importancia señalar que para lograr una integridad óptima en la BD se tuvieron en cuenta cuatro actualizaciones que pueden corromper la integridad referencial:

- La inserción de una fila hija es cuando no coincide la llave ajena con la llave primaria del padre.
- La actualización en la llave ajena de la fila hijo, donde se produce una actualización en la llave ajena de la fila hijo con una sentencia UPDATE y la misma no coincide con ninguna llave primaria.
- La supresión de una fila padre, donde si una fila padre tiene uno o más hijos se suprime, las filas hijos quedaran huérfanas.
- La actualización de la llave primaria de una fila padre, donde si una fila padre tiene uno o mas hijos se actualiza su llave primaria, las filas hijos quedaran huérfanas.

Para lograr la integridad de los datos en la BD también se tuvo presente la integridad del dominio realizando restricciones para algunos atributos.

La integridad de dominio no es más que aquella que están asociadas a un dominio de datos específicos y son aplicadas a cada columna definida sobre el dominio. En otras palabras, en estas restricciones permiten restringir los valores que puede tomar un atributo, estos verifican que no haya valores fuera del dominio establecido.

La BD, del módulo Farmacia tiene varias reglas de integridad de dominio que validan los datos proporcionados a un atributo o columna de la tabla.

A continuación, se describen algunos de los dominios establecidos:

- `dom_identificadores_integer`: este dominio fue realizado para restringir los valores tomados por los identificadores de cada tabla que se les fueron asignados el tipo de dato Integer (entero) ya que este tipo de dato admite valores negativos y un id no debe tomar estos valores.

CAPÍTULO 3. VALIDACIÓN DEL DISEÑO REALIZADO

- Con la implementación de este dominio se asegura que los identificadores de cada tabla no tomen valores negativos ni nulos.
- dom_cantidad_servida: este verifica que la cantidad servida del producto farmacéutico no tome valor negativo o nulo.

Este dominio se asocia al atributo cantidad_servida que se encuentra en las tablas fm_rel_solicitud_producto, fm_rel_prod_pedido y fm_rel_producto_doc_entrada.

- fm_cantidad_solicitada_producto: este verifica que la cantidad solicitada de un producto farmacéutico no tome valores negativos ni cero.

Este dominio se asigna al atributo cantidad_solicitada que se encuentra en la tabla fm_rel_producto_solicitud.

- fm_estado_producto: este verifica que el estado del producto solo tome valor de Activo o Inactivo.

Este dominio se asigna al atributo estado _ producto que se encuentra en la tabla cfg_estado_producto.

- dom_cantidad_cons_serv: este verifica que la cantidad consumida por un servicio de un producto farmacéutico no tome valores negativos.

Para mantener esta integridad referencial en la BD_ Gehos_far se trazaron las siguientes políticas:

- Restricción, logrando impedir modificar o borrar una tupla con clave primaria relacionada con una secundaria.
- Actualización en cascada, permite modificar la tupla, actualizando las relacionadas.
- Anulación, con esto se logra anular en cascada las relacionadas.

Específicamente, se aplicó la restricción para el borrado, y la actualización en cascada para la modificación.

3.1.2 Normalización de la Base de Datos.

La normalización, es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de BD a desarrollar un esquema que minimice los problemas de lógica.

La normalización se adoptó, porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la BD era ineficiente y conducía a errores de lógica, cuando se trataban de manipular los datos.

Las guías que la normalización provee, crean el marco de referencia para simplificar una estructura de datos compleja, evitar la redundancia, evitar problemas de actualización de los datos en las tablas y proteger la integridad de estos.

Otra ventaja de la normalización es el consumo de espacio. Una BD 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.

El proceso de normalización, es aplicable a los modelos entidad relación y a los modelos relacionales. Para normalizar una BD es necesario conocer y aplicar reglas que van a permitir que el diseño realizado sea correcto.

Existen 6 Formas Normales que garantizan un buen diseño de la BD. Básicamente se considera una BD bien diseñada si se encuentra en la Tercera Forma Normal (3ra FN).

La BD_Gehos_far se llevará hasta 3ra FN ya que de esta forma cumple con las funcionalidades requeridas.

- Primera Forma Normal (1ra FN): Una relación está en Primera Forma Normal si y sólo si todos los dominios son atómicos. Un dominio es atómico si los elementos del dominio son indivisibles. Esta forma normal se definió para prohibir los atributos multivaluados, compuestos y sus combinaciones.

CAPÍTULO 3. VALIDACIÓN DEL DISEÑO REALIZADO

La BD del módulo de Farmacia, cumple con esta regla ya que los dominios atómicos existentes fueron eliminados dividiendo esta relación en otras repartiendo sus atributos entre las resultantes. Se eliminaron los atributos que violaban la 1ra FN de la relación original y se colocaron en una relación aparte junto con la clave primaria de la relación de partida. Hasta este momento la BD se encuentra en 1ra FN.

- Segunda Forma Normal (2da FN): "Una relación está en segunda forma normal si está en la 1ra FN y todos los atributos no clave dependen de la clave completa y no sólo de una parte de esta". Varía la clave y varían los atributos.

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

Todas las entidades de la BD Farmacia tienen una clave primaria simple y como se encuentran en 1ra FN pues se garantiza que ya está en 2da FN.

- Tercera Forma Normal (3ra FN): Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave. Esta FN plantea que una relación está en tercera forma normal si todos los atributos de la relación dependen funcionalmente sólo de la clave, y no de ningún otro atributo". Esto significa que en una relación en 3ra FN, para toda **Dependencia funcional F: $X \rightarrow Y$, X es una clave.** [31] En la BD, no existen dependencias transitivas y por lo tanto no se incumple con ella forma normal.

El Modelo Entidad Relación de la BD del módulo Farmacia se modeló aplicando las reglas de la normalización anteriormente mencionadas hasta lograr que se encontrara en 3ra FN, encontrándose en esta forma óptima para cumplir las funcionalidades requeridas. No se modeló hasta otra forma normal más avanzada porque puede conducir a tener una BD ineficiente y construir un esquema demasiado complejo para trabajar.

3.1.3 Análisis de redundancia de información.

En la creación tradicional de programas con procesamiento de archivos, cada grupo de usuarios, mantiene sus propios archivos para manejar las aplicaciones de procesamiento de datos. Estos grupos de usuarios tendrán muy probablemente datos repetidos, por ejemplo, el mismo producto farmacéutico en la BD Farmacia.

La redundancia en el almacenamiento de los datos provoca varios problemas:

- En primer lugar, es necesario realizar una misma actualización lógica varias veces: una vez en cada archivo en el que se almacenen datos. Esto implica una duplicación del trabajo.
- En segundo lugar se desperdicia espacio de almacenamiento al guardar los mismos datos en varios lugares.
- En tercer lugar, es posible que los archivos que representen los mismos datos sean inconsistentes, quizás porque la actualización se haya aplicado a varios archivos y no a otros.

Con el enfoque de BD, las vistas de los diferentes grupos de usuarios se integran durante el diseño de la BD. Para conservar la consistencia, debe crearse un diseño que almacene cada dato lógico –como el nombre de un producto farmacéutico- en un solo lugar de la BD. Esto evita la inconsistencia y ahorra espacio de almacenamiento

En la BD_Gehos_far se ha tenido en cuenta que si una unidad ha realizado más de una solicitud todos los datos de esta unidad no estarán repetidos tantas veces como solicitudes haya, lo mismo sucede para los productos farmacéuticos con el objetivo precisamente de evitar la redundancia de información evitando la repetición de los mismos.

Si se va a dar entrada a un producto farmacéutico ya existente en la BD se realizará una búsqueda en el control que se tiene de ese producto para saber si aún queda en existencia, de ocurrir esto a la existencia actual se le adicionará la nueva cantidad del producto a dar entrada. No se guardará nuevamente este producto en otro lugar con la nueva cantidad a la que se le da entrada.

En la BD_Gehos_far no existe redundancia de la información porque no ha sido necesario duplicar información y sacrificar la propiedad deseable de lograr que no exista información duplicada. Gracias a ello se garantiza, que la información no sea inconsistente.

3.1.4 Análisis de la seguridad de la base de datos.

Los sistemas de gestión hospitalaria deben manejar un gran flujo de datos sensibles con relación a la salud de diversas personas, estos datos son estrictamente confidenciales y es responsabilidad de los creadores de dicho sistema garantizar la confidencialidad y seguridad de estos datos, sobre todo cuando se tienen de manera centralizada en un servidor de bases de datos.

Seguridad significa la capacidad de los usuarios para acceder y cambiar los datos de acuerdo a las políticas del negocio, así como, las decisiones de los encargados.

El SGBD debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el administrador de la BD.

Restricción de los accesos no autorizados:

Cuando muchos usuarios comparten la misma BD, es probable que no todos tengan la autorización para tener acceso a toda la información que contiene.

También es posible que algunos usuarios sólo tengan permiso para recuperar datos, mientras que otros pueden obtenerlos y actualizarlos. El SGBD cuenta con un subsistema de seguridad y autorización que permite crear cuentas y especificar restricciones para ellas.

La seguridad de la base de datos en PostgreSQL está implementada en varios niveles:

- Protección de los ficheros de la BD. Todos los ficheros almacenados en la BD están protegidos contra escritura por cualquier cuenta que no sea la del usuario Postgres u otro usuario que se cree con los mismos permisos que este.

- Las conexiones de los clientes son restringidos por dirección IP y/o por nombre de usuario mediante el fichero `pg_hba.conf` situado en `PG_DATA`.
- A cada usuario de Postgres se le asigna un nombre de usuario y (opcionalmente) una contraseña.
- Los usuarios pueden ser incluidos en *grupos*, y el acceso a las tablas puede restringirse en base a esos grupos. [32]

El acceso a la BD también es controlado por un mecanismo de contraseñas; un usuario que quiera acceder al sistema debe dar una contraseña y esta es validada por el sistema.

Autenticación de Usuarios en Postgres:

Autenticación es el proceso mediante el cual el servidor de la BD se asegura de que el usuario que está solicitando acceso es en realidad quien dice ser. Todos los usuarios que quieren utilizar Postgres se comprueban en la tabla `pg_user` para asegurarse que están autorizados a hacerlo.

Además de estas propiedades con que cuenta el SGBD PostgreSQL para implementar la seguridad en el proyecto se desarrolla un módulo de seguridad que provee otros métodos para la seguridad en el servidor donde se encuentran todas las BD del proyecto.

Por lo general no se crean usuarios a los que se les otorga los permisos pertinentes. Es más simple crear roles a los que se les otorgan permisos determinados. A estos roles se les asignan usuarios en dependencia de la funcionalidad que tendrá. Una vez asignados los usuarios que se determinen a los roles creados, cada usuario tendrá los permisos que tenga ese rol al que fue asignado.

La BD consta del usuario `postgres`. Este por los privilegios otorgados, es capaz de ejecutar cualquier funcionalidad como: insertar, eliminar, actualizar, modificar las tablas y las funciones implementadas. Este usuario es quien tiene los permisos para realizar cualquier cambio en la BD.

También se cuenta con el rol `usuario`. En este rol estarán todos los usuarios que tendrán solo los permisos de actualizar, insertar, seleccionar y eliminar información de la BD, además de permitir la ejecución de los procedimientos almacenados para manipular la información.

Se creó dentro de este rol el usuario `gehosfar`, que tiene los mismos permisos que el rol usuario que es a quien pertenece. Este usuario debe ser correctamente manipulado por la capa de negocio, pues debe ser asignado sólo a las personas autorizadas a realizar cambios en la información.

Para aquellos usuarios que sólo tendrán permisos de lectura, se cuenta con otro rol, denominado usuario limitado que sólo tendrá permisos a consultar información de la BD, y no por el contrario de escribir, modificarla y realizar algún cambio drástico en ella, como eliminar alguna tabla.

Copia de seguridad y restauración

Hacer copias de seguridad significa realizar una copia de los ficheros de datos a cualquier otro medio (disco duro, unidades de cinta, cd-rom, etc.) que permita recuperarlos si por alguna incidencia se pierden. Restaurar copias de seguridad es hacer el proceso contrario, es decir, volcar los datos existentes en alguna copia de seguridad sobre los datos del programa para recuperar alguna información perdida o para otros fines como cambiar el programa de un ordenador a otro, etc. Es muy importante no confundir ambos procesos porque al restaurar una copia de seguridad se sustituyen los datos que actualmente tenga el programa por los que haya en esa copia de seguridad y que pueden estar desfasados o incompletos.

PostgreSQL proporciona dos utilidades para realizar las copias de seguridad de su sistema: `pg_dump` para copias de seguridad de bases de datos individuales y `pg_dumpall` para realizar copias de seguridad de toda la instalación de una sola vez. [33]

Las copias de seguridad de la `BD_Gehos_far` se realizarán a nivel de servidor en conjunto con las restantes BD del proyecto y además se realizarán copias de seguridad individual, para ello se utiliza la siguiente orden:

```
% pg_dump nombredb > nombredb.pgdump
```

y será restaurada utilizando:

```
cat nombredb.pgdump | psql nombredb
```

3.1.5 Trazabilidad de la acciones.

Cuando se crea una BD se hace necesario e imprescindible tomar ciertas medidas de seguridad para garantizar el control de los datos y el uso que se hace de ellos. Para tener un estricto control del uso que se hace de los datos almacenados se toman un conjunto de medidas, acciones y procedimientos que permiten registrar e identificar todas las operaciones realizadas final por un determinado usuario que acceda a la BD.

Trazabilidad es la capacidad que tiene un sistema de BD para rastrear, reconstruir o establecer relaciones entre objetos monitoreados, para identificar y analizar situaciones específicas o generales en los mismos. Está estrechamente relacionada, con el requisito de los sistemas por mantener evidencia de las acciones realizadas por usuarios o procesos en el uso de los sistemas de información y las posibles relaciones entre ellos. Puede realizarse sobre tablas o sobre procesos. Incluso pueden crearse trazabilidad sobre errores o conflictos de la aplicación.

Es necesario llevar control de cada una de las acciones que se realizan en la BD, de esta forma se tiene una detallada descripción de lo que ocurre con cada uno de los elementos que componen la BD para poder saber en un determinado momento donde estuvo el error que se cometió, quién y cuándo lo cometió. De esta forma es mucho más fácil saber por donde empezar a restablecer el sistema en caso de algún fallo.

El SGBD PostgreSQL brinda la posibilidad de registrar y guardar todo lo que ocurre en la BD. En los ficheros logs (traducido al español como registros) se guardan todas las acciones que hizo un usuario determinado, estas acciones pueden ser: inserción, eliminar, actualización, que alteren la información almacenada en la BD. Esta opción es habilitada en el fichero de configuración de este gestor, denominado "postgres.conf".

El usuario que se registra en estos logs es el que se utiliza para conectarse a la BD, este usuario es el que se crea a nivel de BD.

De esta forma se obtendrá la información sobre la trazabilidad de las acciones realizadas por los usuarios en la BD.

Conclusiones.

En este capítulo, se ha realizado una validación teórica del diseño propuesto, exponiéndose los métodos realizados para asegurar el buen funcionamiento de la BD.

Se analizó la integridad referencial que guarda cada tabla, con las que se relaciona.

Se llevó a cabo la normalización de la BD llevándola hasta la tercera forma normal, garantizando que exista poca redundancia en la BD.

Además, se crearon roles en la BD para lograr mayor seguridad y proteger la información que se ha almacenado. Con estos, se controlan los permisos de cada usuario que se conecte a la BD. También, se guardan las trazas de todas las operaciones que se realizan en la BD, garantizando de esta forma que se registren cada una de las modificaciones que se realicen en ella.

Conclusiones

Durante el desarrollo de este trabajo, se realizaron estudios que demuestran la necesidad de crear una BD que permita almacenar, consultar, modificar, insertar la información referente a los datos que se gestionan y capaz de adaptarse a las crecientes necesidades de las farmacias.

Para dar respuesta a las necesidades que existe actualmente en las farmacias hospitalarias se diseñó una BD capaz de almacenar toda la información que se gestionan en estas. Para ello:

- Para la elaboración de la BD se hizo utilizó el gestor de BD PostgreSQL 8.2. Esta es una potente herramienta que permite contar con una BD que satisface las necesidades de almacenamiento de las farmacias hospitalarias.
- Se estructuró y diseñó un modelo conceptual de la BD cumpliendo con los requerimientos obtenidos.
- Se implementaron las funciones (procedimientos almacenados), vistas y dominios de la BD del Módulo Farmacia.
- Se hizo una validación teórica, del diseño propuesto para la BD.

De esta forma se cumple con las tareas trazadas para darle cumplimiento al objetivo planteado, obteniendo el diseño de una BD que mejorará la gestión de la información referente a los procesos que en las farmacias hospitalarias se gestionan. Esto favorecerá el incremento en la eficiencia de los servicios ofrecidos en estas instituciones.

Recomendaciones

Con el objetivo de mejorar y concluir una versión más completa de la BD_Gehos_far propuesta en este documento se recomienda lo siguiente:

- Realizar validación funcional del diseño propuesto, con la realización de pruebas de carga intensiva que permitan validar el rendimiento de la BD.
- Continuar con la investigación, para garantizar nuevas y buenas mejoras en futuras versiones de la BD del sistema propuesto.
- Para una próxima iteración, considerar los cambios que puedan surgir en el diseño del sistema y que influyan en el comportamiento de la BD.

Referencias Bibliográficas

[1] Delgado Ramos, Ariel y Vidal Ledo, María. Informática en la salud pública cubana. *Rev Cubana Salud Pública*. [online]. jul.-sep. 2006, vol.32, no.3 [citado 13 Junio 2007], p.0-0. Disponible en la World Wide Web:

<http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-34662006000300015&lng=es&nrm=iso&tlng=es>. ISSN 0864-3466.

[2] Programas de formación MIR. Farmacia Hospitalaria. 2007. Disponible en:

http://www.trainmed.com/trainmed/cda/CdaDispatcher?page=html/pf_farmahos.html&menuitemid=11

[3] Programa de desarrollo de la atención farmacéutica en oficinas de farmacias. Julio 1999. Disponible en:

<http://www.colfacor.org.ar/atencionfarmaceutica.htm>

[4] Valle, José. "Herramientas CASE para BD". 2006. Disponible en:

<http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml>

[5] Vallejo, Cesar. ¿Qué es UML y qué es RUP? octubre 08, 2005. disponible en:

<http://www.solotecperu.com/blog/2005/10/qu-es-uml-y-qu-es-rup.html>

[6] Rommell, Laitano. Modelo de base de datos con ER/studio. 2003. Disponible en:

<http://www.ilustrados.com/publicaciones/EpyuVuuuEuEqmfYGtf.php>

[7] Toad. 2003. Disponible en:

<http://descargas.terra.es/ie/26344-Toad>

[8] Case Studio. 2005. Disponible en:

<http://case-studio.softonic.com/>

[9] Dennys. Dennys Cafe Net. Bases de Datos. 2006. Disponible en:

http://www.foroswebgratis.com/tema-bases_de_datos-36291-209428.htm

[10] Sistemas de Bases de Datos. Disponible en:

<http://usuarios.lycos.es/cursosgbd/UD2.htm>

[11] Tramillas, Jesús y Kronos. Introducción a la documática. Los sistemas de Base de Datos y los SGBD. Disponible en:

<http://www.tramullas.com/documatica/2-4.html>

[12] Tramillas, Jesús y Kronos. Introducción a la documática. Los sistemas de Base de Datos y los SGBD. Disponible en:

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

[13] Díaz, Walter. Manual de SQL. Disponible en:

<http://walter.freesevers.com>

[14] ¿Por qué decidirse por InterBase? Disponible en:

<http://www.intitec.com/articulos/DecisionInterBase.htm>

[15] Cañete, Patricio. Oracle. Disponible en:

<http://www.pcanete.com.ar/leer.asp?idx=429>

[16] PostGreSQL. Disponible en:

http://www.netpecos.org/docs/mysql_postgres/x15.html

[17] Quiñones, Ernesto. AINTRODUCCION A POSTGRESQL. Disponible en:

http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf

[18] Diseño y Programación. PostgreSQL8.2. Disponible en:

<http://www.buayacorp.com/archivos/postgresql-82/>

[19] Aguilar, Vicente y Suau, Pablo. MySQL vs. PostgreSQL. Disponible en:

<http://www.fedora-es.com/node/189>

[20] *EMS SQL Manager 2005 Lite for PostgreSQL*. Disponible en:

<http://www.archivospc.com/programas/archivos/EMS%20SQL%20Manager%202005%20Lite%20for%20PostgreSQL.php>

[21] FARHOS. Gestión de farmacias hospitalarias. Disponible en:
http://www.visual-limes.com/pdf/farhos_presentacion.pdf

[22] Gestifarma. Disponible en:
<http://www.cenfarte.es/gestifarma/derecha.html>

[23] Care2x. Disponible en:
<http://www.care2x.org/>

[24] BIOCOM. Disponible en:
<http://www.biocom.com/home.html>

[25] InfarmVisual. Software para Oficinas de Farmacia
<http://www.farmages.es/page9.html>

[26] Modelo Relacional. Disponible en:
http://es.wikipedia.org/wiki/Modelo_relacional

[27] Pulido Bernal, Andrés Javier. Optimización de consultas. Disponible en:
<http://www.dimensis.com/article140.html> 03/06/2007

[28] Abad Mota, Soraya. Entonación de Bases de Datos. (Apuntes de la electiva libre)

[29] ANDRÉS, M. M. M. *Reglas de integridad*, 6 de junio de 2007]. Disponible en:
<http://www3.uji.es/~mmarques/f47/apun/node52.html>

[30] http://es.wikipedia.org/wiki/Integridad_de_datos

[31] Diseño de Base de datos relacionales. Disponible en:
<http://usuarios.lycos.es/cursosgbd/UD4.htm>

[32] Informática para ingenieros. Software Básico. Base Datos y sus usuarios. Disponible en:
<http://webpages.ull.es/users/mbmelian/BD.pdf>

[33] Guía del Administrador de PostgreSQL. Disponible en:
<http://es.tldp.org/Postgresql-es/web/navegable/admin/security.html>

[34] Guía del Administrador de PostgreSQL. Gestión de una base de datos. Disponible en:
<http://es.tldp.org/Postgresql-es/web/navegable/admin/x1401.html>

Bibliografía

Abad Mota, Soraya. Entonación de Bases de Datos. (Apuntes de la electiva libre).

ANDRÉS, M. M. M. *Reglas de integridad*, 6 de junio de 2007]. Disponible en:
<http://www3.uji.es/~mmarques/f47/apun/node52.html>

Aguilar, Vicente y Suau, Pablo. MySQL vs. PostgreSQL. Disponible en:
<http://www.fedora-es.com/node/189>

Date C. J. *Introducción a los sistemas de Bases de Datos*, primera parte. La Habana, Editorial Félix Varela, 2003. 326 págs.

Date C. J. *Introducción a los sistemas de Bases de Datos*, segunda parte. La Habana, Editorial Félix Varela, 2003. 173 págs.

Date C. J. *Introducción a los sistemas de Bases de Datos*, tercera parte. La Habana, Editorial Félix Varela, 2003. 381 pág.

Dennys. Dennys Cafe Net. Bases de Datos. 2006. Disponible en :
http://www.foroswebgratis.com/tema-bases_de_datos-36291-209428.htm

Delgado Ramos, Ariel y Vidal Ledo, María. Informática en la salud pública cubana. *Rev Cubana Salud Pública*. [online]. jul.-sep. 2006, vol.32, no.3 [citado 13 Junio 2007], p.0-0. Disponible en:
<http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-34662006000300015&lng=es&nrm=iso&tlng=es>. ISSN 0864-3466.

Díaz, Walter. *Manual de SQL*. Disponible en:
<http://walter.freesevers.com>

Guía del Administrador de PostgreSQL. Disponible en:
<http://es.tldp.org/Postgresql-es/web/navegable/admin/security.html>

Informática para ingenieros. Software Básico. Base Datos y sus usuarios. Disponible en:
<http://webpages.ull.es/users/mbmelian/BD.pdf>

Quiñones, Ernesto. AINTRODUCCION A POSTGRESQL. Disponible en:

http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf

Rommell, Laitano. Modelo de base de datos con ER/studio. 2003. Disponible en:

<http://www.ilustrados.com/publicaciones/EpyuVuuuEuEqmfYGtf.php>

Tramillas, Jesús y Kronos. Introducción a la documática. Los sistemas de Base de Datos y los SGBD. Disponible en:

<http://www.tramullas.com/documatica/2-4.html>

Tramillas, Jesús y Kronos. Introducción a la documática. Los sistemas de Base de Datos y los SGBD. Disponible en:

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

Valle, José. "Herramientas CASE para BD". 2006. Disponible en:

<http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml>

Glosario de términos

A

Administración de una base de datos: Es el rol que juega una o varias personas de controlar un una base de datos en cuanto a seguridad y configuración, además de chequear el buen funcionamiento del mismo.

H

Herramienta Case: Ingeniería de sistemas asistida por ordenador (Computer-Aided Systems Engineering - CASE) es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo de sistemas

I

Información:

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

IBM

International Business Machines o **IBM** es una empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática.

INFOMED

Es el Portal de Salud Cubano y la red de personas e instituciones que comparten el propósito de facilitar el acceso a la información de salud en Cuba.

M

Medicamento:

Es toda sustancia o mezcla de sustancias que, preparada bajo una forma farmacéutica determinada con dosificación apropiada, se administra a un organismo para curar una afección o trastorno, o para prevenir enfermedades.

MINSAP

Ministerio de Salud Pública.

R

Reporte: Informe detallado sobre alguna información, o sobre el estado de la información.

S

Sistema:

Conjunto de instrucciones organizadas, sistematizadas y lógicas relacionadas entre sí, con el objetivo de obtener, analizar, relacionar y generar la información que satisfaga las necesidades operativas y administrativas de una organización.

Scripts

Son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución, se distinguen de los programas, pues deben ser convertidos a un archivo binario ejecutable para correrlos.

Software (componentes lógicos, programas, software).

Programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red.

U

UML

Lenguaje Unificado de Modelado.