



**Universidad de las Ciencias Informáticas
Facultad No 3 Turismo y Negocios**

“Diseño de la capa de datos del Sistema de Gestión de Inventarios y Almacenes”.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN
CIENCIAS INFORMÁTICAS

Autor: Yenier Figueroa Machado

Tutor: Dra. Ana María García Pérez

2007

“Año 49 de la Revolución”

"El genio comienza las grandes obras, pero sólo el trabajo las acaba"

Joseph Joubert (1754-1824)



DECLARACIÓN DE AUTORÍA

Por este medio declaro que soy el único autor de este trabajo y autorizo a la Facultad No. 3 de la Universidad de las Ciencias Informáticas de la Habana para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los 2 días del mes de Junio del 2007.

Firma del Autor

Firma del Tutor

AGRADECIMIENTO

Primeramente, a dos personas, las cuales han contribuido de forma *significativa* a este trabajo y a mi vida, Dra. Ana María García Pérez, y Cesar García.

A mi familia por que con su cariño, guía y apoyo he llegado a realizar uno de los anhelos más grande de la vida, fruto del inmenso apoyo, amor y confianza que en mí se depositó y con los cuales he logrado terminar mis estudios profesionales que constituyen el legado más grande que pudiera recibir y por lo cual les viviré eternamente agradecido.

A cada uno de mis profesores que ayudaron en mi formación.

Deseo agradecer a mis compañeros por su apoyo incondicional durante todo el proceso de formación académica y realización de este trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Muchas Gracias.....

DEDICATORIA

Al líder de la Revolución Cubana, Fidel Castro Ruz por darme la oportunidad de ser parte de este proyecto de la revolución.

A mis padres Mayda, Ana, Manuel y Lucas, por su inmenso amor, comprensión, por enseñarme a que debo tener la fortaleza de continuar hacia adelante sin importar las circunstancias que la vida nos presente, ustedes quienes siempre han sido ejemplo en mi vida.

A mis hermanos Osiel y Anabel por su cariño incondicional.

A Enma, Maro, Yeney, Michal y Leydi, por que junto a ustedes la alegría compartida aumenta su proporción y la tristeza disminuye significativamente.

A Yanelys por su amor, apoyo, cariño, respeto y colaboración, durante todo este proceso. Por compartir las alegrías, las penas y por siempre tener palabras de aliento.

A todos mis amigos que me han entregado su amor y apoyo siempre.

A todas las personas que han creído en mí.....

RESUMEN

En este Trabajo de Diploma se describe el diseño del Modelo Lógico de la Capa de Datos de un Sistema de Gestión de Inventarios (SIGIA) concebido para ser aplicado en una empresa.

Se ha utilizado un modelo genérico de empresa, al tratar las relaciones de la misma con sus proveedores y clientes, con las implicaciones que las operaciones tienen en el manejo de sus inventarios.

Se han estudiado almacenes de hoteles, de una empresa de reparación de vehículos, y de la Universidad de las Ciencias Informáticas (UCI).

Se obtuvo un modelo lógico que garantiza el desarrollo de la aplicación con sus requisitos funcionales esenciales para una versión 1.0.

Dicho modelo permite a los programadores garantizar la integridad, seguridad y trazabilidad de las operaciones. Se ha utilizado una metodología de diseño orientada a los objetos al extraer las entidades del entorno del dominio del problema. También se emplearon los conceptos de la modelación entidad relación extendida lo que permite que el modelo físico quede en Tercera Forma Normal evitando que exista redundancia en la información que se almacena.

Con este diseño se pretende sentar las bases estructurales de un Sistema de Gestión de Inventarios moderno y con las cualidades que necesita la empresa cubana actual.

Palabras claves: Base de Datos, Diseño, Almacén, Inventario

SUMMARY

In this Diploma Paper the author describes the design of the Logical Pattern of the data layer of an administration system of Inventories (SIGIA) conceived to be applied in a company.

A generic model of company has been used, when treating the relationships of the same one with its suppliers and clients, with the implications that the operations have in the handling of their inventories.

For the investigation some warehouses from different hotels, a company for the repairing of vehicles, and the informatics Science University (ICU) were studied.

A logical model that guarantees the development of the application was obtained with its essentials functional requirements for version 1.0.

This model allows the programmers to guarantee the integrity, security and trace of the operations. A design methodology has been used guided to the objects when extracting the entities of the environment of the problem domain. The concepts of the entity extended relationship model were also used which allows that the physical pattern remains in Third Normal Form avoiding existing redundancies in the stored information.

With this design it is intended to establish the structural bases of a modern System of Inventories Administration with the qualities the current Cuban company demands.

Keyword: Data Base, Design, Warehouses, Inventories.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO I: “Estudio del estado del arte en desarrollo de bases de datos para sistemas de gestión”.....	6
1.1 Introducción.....	6
1.2 Evolución de los Sistemas de Gestión de Bases de Datos.....	7
1.2.1 Primera generación de Sistemas Gestores de Bases de Datos.....	7
1.2.2 Segunda generación de Sistemas Gestores de Bases de Datos.....	10
1.2.3 Tercera y Cuarta Generación de Sistemas Gestores de Bases de Datos...	11
1.3 Metodologías de diseño de base de datos.....	15
1.3.1 Fases de diseño de Bases de Datos.....	16
1.4 Estrategias de diseño de bases de datos.....	17
1.5 El Proceso Unificado de Rational y los diseñadores de base de datos.....	17
1.6 ERStudio, herramienta CASE para el diseño de Base de Datos.....	19
1.7 Tendencias de las Bases de Datos.....	20
1.7 Conclusiones.....	21
CAPÍTULO II: Diseño Lógico de la base de datos del “Sistema de Gestión de Inventario Y Almacenes” (SIGIA).....	22
2.1 Introducción.....	22
2.2.1 Formato de las entidades.....	22
2.2.2 Formato de los atributos.....	23
2.3 Descripción del problema.....	23
2.3.1 Entrada de productos por compra a proveedores.....	25
2.3.2 Salida de productos.....	33
2.3.3 Solicitud de materiales desde un departamento solicitante.....	37
2.3.4 Transferencia de productos entre secciones.....	39
2.3.5 Devolución de entradas de productos.....	43
2.4 Descripción de la Aplicación.....	51
2.4.1 Administración.....	51
2.4.2 Nomencladores.....	55
2.5 Caracterización y análisis crítico de la propuesta de diseño lógico.....	59
2.6 Conclusiones.....	63

CONCLUSIONES	64
RECOMENDACIONES.....	65
REFERENCIAS BIBLIOGRÁFICAS.....	66

INTRODUCCIÓN

Desde tiempos inmemorables, los egipcios y demás pueblos de la antigüedad, acostumbraban almacenar grandes cantidades de alimentos para ser utilizados en los tiempos de sequía o de calamidades. Para hacer frente al manejo de los productos en los periodos de escasez, de manera que aseguraran la subsistencia de la vida y el desarrollo de sus actividades normales, se necesitaba llevar el control de las existencias. La necesidad de controlar el almacenamiento de todos los bienes y alimentos necesarios para sobrevivir motivó la aparición de los inventarios.

En estos tiempos modernos, la palabra inventario ha alcanzado una notoriedad muy alta, esto se debe a que la base de toda empresa comercial es la compra y venta de bienes y servicios. Con la introducción masiva de las nuevas tecnologías de la información y las comunicaciones, conocidas por las siglas TICs, y su impacto en el desempeño de las pequeñas y medianas empresas, es que surgen soluciones tecnológicas con un amplio nivel de aplicación en las operaciones relacionadas con la gestión de inventarios.

En el control del inventario las empresas manejan una gran cantidad de datos. Cualquier empresa que se precie debe tener almacenados todos estos datos en una base de datos para poder acceder a estos mediante una aplicación profesional; sin esta funcionalidad resultaría engorroso tratar y manejar en su totalidad los procesos que lleva a cabo la empresa y se perdería un tiempo y un dinero muy valiosos.

La revolución cubana, ha apostado fuerte por la informatización, debido a que ha reconocido la importancia estratégica de la informática para la sociedad. Más de una vez se han hecho planes estratégicos de desarrollo. Esta vez, sin embargo, ese reconocimiento y esa estrategia se ubican en un momento histórico que se caracteriza por:

- cientos de miles de computadoras de variado tipo instaladas en el país,
- un considerable capital humano relacionado con las diferentes especialidades informáticas como el software, el hardware y las comunicaciones,
- muchas entidades están basando su tratamiento de la información en soluciones informáticas,
- la cultura general informática ha penetrado en todos los rincones de la sociedad, pues hasta el ciudadano más alejado de los centros de gestión conoce qué es una computadora y para qué sirve.

INTRODUCCIÓN

Es un momento favorable para la elaboración y mejoras de sistemas y aplicaciones informáticas, en especial las relacionadas con la gestión de inventarios.

La Universidad de las Ciencias Informáticas, como parte de este proceso de informatización, desarrolla un proyecto para la gestión de inventarios y almacenes cuyo nombre es “*Sistema de Gestión de los Inventarios y Almacenes*” (SIGIA), aplicable a cualquier empresa o entidad, con el que se quiere tratar de solucionar aspectos débiles que existen en el control de los inventarios.

La forma en que el sistema almacenará la información que se utiliza o genera a partir de la gestión de inventarios y almacenes constituye una interrogante.

Lo antes expuesto permitió identificar el siguiente **problema**: ¿Cómo almacenar los datos que se utilizan o generan a partir de la Gestión de los Inventarios y Almacenes?

Para ello se planteó como **objetivo**: Diseñar el modelo lógico de la capa de datos, que permita a la aplicación almacenar los datos que se utilizan o generan en la Gestión de Inventarios y Almacenes.

Objeto de estudio: Proceso de almacenamiento y utilización de los datos del Sistema de Gestión de Inventario y Almacenes.

Campo de acción: Diseño lógico de la base de datos relacional para el Sistema de Gestión de Inventarios de Almacenes.

Hipótesis: Si se construye el diseño lógico de una base de datos que cumpla con los requerimientos y con un diseño moderno de la misma, se podrá disponer de un sistema que controle con eficiencia los inventarios en los almacenes de una Empresa.

Variables independientes: Diseño lógico de una base de datos que cumpla con los requisitos del sistema.

Variable dependiente: Eficiencia en el control de los inventarios de los almacenes.

La investigación se desarrolló a través de un conjunto de **tareas**:

1. Revisar la bibliografía existente sobre la gestión de inventarios, con el fin de obtener los conocimientos necesarios para el desarrollo de la capa de datos del sistema de gestión de inventario y almacenes.
2. Revisar la bibliografía científica sobre los softwares usados en la gestión de inventarios.
3. Estudiar las herramientas de diseño de Base de Datos.
4. Contactar con diseñadores de bases de datos relacionales que han estado vinculados a sistemas de gestión de almacenes.
5. Visitar almacenes a fin de identificar las necesidades actuales de los mismos.
6. Decidir las herramientas a utilizar para el diseño de la capa de datos.
7. Diseñar el modelo lógico de los datos para el sistema de Gestión de Inventarios y almacenes.

La realización de las tareas estuvo sustentada en un conjunto de **métodos de investigación** que ahorran esfuerzos y tiempo. Estos métodos se dividen en dos grandes grupos: los métodos teóricos y los empíricos [Hernández y Coello 2002]. Los primeros son todos aquellos que crean las condiciones para ir mas allá de las características fenomenológicas y superficiales de la realidad, posibilitan el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada, su relación con otros fenómenos, mientras que los métodos empíricos permiten extraer de los fenómenos analizados las informaciones que se necesitan sobre ellos a través de observaciones, del uso de técnicas opináticas y la propia experimentación.

En el grupo de los **métodos teóricos** utilizados se encuentra el analítico-sintético, el cual plantea que el análisis y la síntesis son dos procesos inherentes al pensamiento, operaciones lógicas importantes que permiten, como métodos teóricos, buscar la esencia de los fenómenos, los rasgos que los caracterizan y los distinguen. Este método permitió analizar las teorías, documentos, etc.; facilitando la extracción de los elementos más importantes que se relacionan con los almacenes de las pequeñas y medianas empresas, así como de la bibliografía científica teórica sobre los softwares usados en la gestión de inventarios. También se empleó el inductivo – deductivo que consiste en aplicar formas de razonamiento que permiten llegar a un grupo de conocimientos generalizadores, tanto desde el análisis de lo particular a lo general, como desde el análisis de elementos generalizadores a uno de menor nivel de generalización, siendo éste uno de los métodos esenciales para lograr la

INTRODUCCIÓN

generalización de un conjunto de entidades en el diseño de la base de datos para el Sistema de gestión de inventarios de almacenes.

Otro método utilizado en la investigación lo es sin duda el Análisis histórico lógico, el cual permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Lo que permitió constatar teóricamente como ha evolucionado la disciplina de los sistemas de bases de datos hasta la actualidad. La Modelación como método teórico se utilizó en la creación de modelos, constituyendo estos una reproducción simplificada de la realidad. También el método sistémico a través del cual se estudió el objeto mediante la determinación de sus componentes, así como la relación entre ellos que conforma una realidad como totalidad. Esa relación determina por un lado, la estructura y la jerarquía de cada componente en el objeto y por otra parte su dinámica, siendo también la expresión del comportamiento del sistema como totalidad en que un componente depende de otro u otros. Convirtiéndose dichas relaciones en las leyes del sistema.

Por su parte, en el grupo de los **métodos empíricos** se emplea la observación a través de un registro visual de lo que ocurre en los almacenes, en la gestión de inventarios, clasificando y consignando los hechos y acontecimientos pertinentes. La observación realizada fue externa y en la modalidad de abierta, pues el objeto de la investigación tenía conocimiento de que estaban siendo observados. Se utilizó además la entrevista a través de un encuentro con diseñadores de Bases de datos de sistemas de gestión de almacenes, con el fin de obtener información relacionada con la misma y facilitar de esa forma la creación del nuevo modelo de datos para el sistema de gestión de inventarios.

Aportes prácticos del trabajo:

El diseño del modelo Lógico de la capa de datos, que permite a la aplicación almacenar los datos que se utilizan o generan en la Gestión de Inventarios y Almacenes.

Este trabajo tiene la siguiente estructura:

El Capítulo 1 aborda un análisis crítico de investigaciones anteriores y de fuentes con enfoques, teorías y modelos relacionados con el estudio del estado del arte de Sistemas de Bases de Datos, así como el papel decisivo del diseñador de las mismas para la obtención de diseños eficientes con la ayuda de la herramienta Case

INTRODUCCIÓN

ERStudio. Este capítulo incluye además algunas de las características de RUP (Proceso Unificado de Rational) como metodología empleada en el desarrollo del proyecto la cual define dentro de sus trabajadores el diseñador de base de Datos.

En el capítulo 2 se exponen los resultados principales de este trabajo, que consisten en la definición del modelo conceptual de la base de datos, utilizando la notación IDEF1x empleada por el software de diseño de bases de datos ER/Studio V 6.0.1, así como la descripción de cada una de las entidades informativas y sus atributos. Finaliza el mismo con una caracterización y análisis crítico del diseño de modelo lógico propuesto en dicho capítulo.

CAPÍTULO I: “Estudio del estado del arte en desarrollo de bases de datos para sistemas de gestión”.

1.1 Introducción.

El escenario actual de la tecnología de bases de datos es resultado del perfeccionamiento que a lo largo de casi 6 décadas ha tenido lugar en el procesamiento de datos y en la gestión de la información, por lo que es preciso conocer cuál ha sido la evolución y estado actual de la tecnología de bases de datos, con el objetivo de llevar adelante el desarrollo de este sistema sustentado por los conocimientos sólidos recogidos en esta primera parte investigativa.

El término base de datos ha sido definido de varias maneras. Algunas de ellas:

- Colección de datos interrelacionados. [Elmars, R, Navathe, S.B.1989].
- Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una BD puede considerarse una colección de datos variables en el tiempo. [Garcia, 1999]
- Colección no redundante de Datos que son compartidos por diferentes programas de aplicación. [Howe, 1983].
- Conjunto de datos de la empresa memorizados en un ordenador, que es utilizado por numerosas personas y cuya organización esta regida por un modelo de Datos. [Flory, 1982].
- “Una base de datos es un conjunto de información almacenado en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.” [Decups, 2002]
- “Una base de Datos es un conjunto exhaustivo, no redundante de datos estructurados, organizados independientemente de su utilización y su implementación en la maquina, accesible en tiempo real y compatible por usuarios concurrentes que tienen necesidad de información diferente y no predecible en el tiempo.” [Decups, 2002]
- “Colección o depósitos de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes a estas, y su definición y descripción, únicas para cada tipo de datos, han de estar almacenados junto con los mismos. Los procedimientos de actualización y recuperación, comunes

y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de los datos.” [Decups,2002]

1.2 Evolución de los Sistemas de Gestión de Bases de Datos.

Un breve recorrido por la historia de la tecnología de Base de Datos hará llegar a la modernidad con bases y argumentos sólidos para defender las decisiones tomadas para el diseño de la capa de datos del sistema de gestión de inventarios.

En la actualidad, la incursión en esta tecnología permite delegar la gestión de una base de datos a unos tipos de aplicaciones de software específicos, denominados *sistema de gestión de bases de datos* (SGBD) o, simplemente, *sistemas de bases de datos* [Hernandez Orallo, 2002]. Por esta razón, hablar de la tecnología de bases de datos es prácticamente lo mismo que hablar de la tecnología de los sistemas de gestión de bases de datos. Sin embargo es necesario saber diferenciar los términos de Bases de Datos y Sistemas de Gestión de Bases de Datos. En estos últimos el objetivo fundamental consiste en suministrar a los usuarios las herramientas que le permitan manipular *en términos abstractos* los datos, es decir, de forma que puedan obtener información útil a partir de los datos almacenados, sin necesidad de conocer su modo de almacenamiento en la computadora, ni el método de acceso empleado.

1.2.1 Primera generación de Sistemas Gestores de Bases de Datos.

Los primeros sistemas de bases de datos aparecieron a finales de los cincuenta. El Presidente Kennedy anunció que serían los Estados Unidos de América los primeros en poner un hombre en la Luna y hacerlo regresar a la Tierra. En la década de los 60 nacía el “Proyecto Apolo” al frente de Werner von Braun, el cual absorbería toda la actividad Astronáutica durante los años siguientes y para su realización se precisaba de un sistema que permitiera gestionar la inmensa cantidad de información que requería el proyecto.

Los predecesores de los sistemas de bases de datos fueron los *sistemas de ficheros* [Marqués Andrés, 2001] los cuales almacenaban datos durante un largo periodo de tiempo y permiten el almacenamiento de grandes cantidades de datos. Sin embargo, los sistemas de ficheros no garantizaban generalmente que los datos no se perdieran ante fallos bastante triviales, y se basaban casi exclusivamente en recuperación por copia de seguridad. No hay un momento concreto en que los sistemas de ficheros hayan cesado y hayan dado comienzo los sistemas de bases de datos. De hecho, todavía existen sistemas de ficheros en uso.

Los sistemas *jerárquicos* o basados en árboles y el *de red* o basados en grafos fueron los más popularizados en los años sesenta y constituyen la primera generación de los SGBD. [Marqués Andrés, 2001].

Los sistemas jerárquicos deben su origen a la empresa de aviación norteamericana “NAA”, que desarrolló un software denominado GUAM (General Update Access Method) basado en la idea de que componentes mas pequeños se juntan como partes de componentes mas grandes, y así siguiendo hasta obtener el producto final [Marqués Andrés, 2001]. Esta estructura, que tiene la forma de un árbol, es lo que se denomina una estructura jerárquica.

A mediados de los sesenta, IBM se unió a NAA para desarrollar GUAM en lo que ahora se conoce como IMS (Information Management System: Sistema de administración de información). El motivo por el cual IBM restringió IMS al manejo de jerarquías de registros fue el de permitir el uso de dispositivos de almacenamiento en serie, más exactamente las cintas magnéticas, que era un requisito del mercado por aquella época.

Este modelo fue adoptado por muchos bancos y compañías de seguros que todavía los utilizan en algún caso hoy en día. Los sistemas de base de datos jerárquicos se pueden encontrar en algunas empresas para gestionar el inventario y la contabilidad, aunque la renovación provocada por el efecto 2000 ha eliminado prácticamente su uso, así como el reciclaje de los expertos en estos sistemas a otros más modernos.

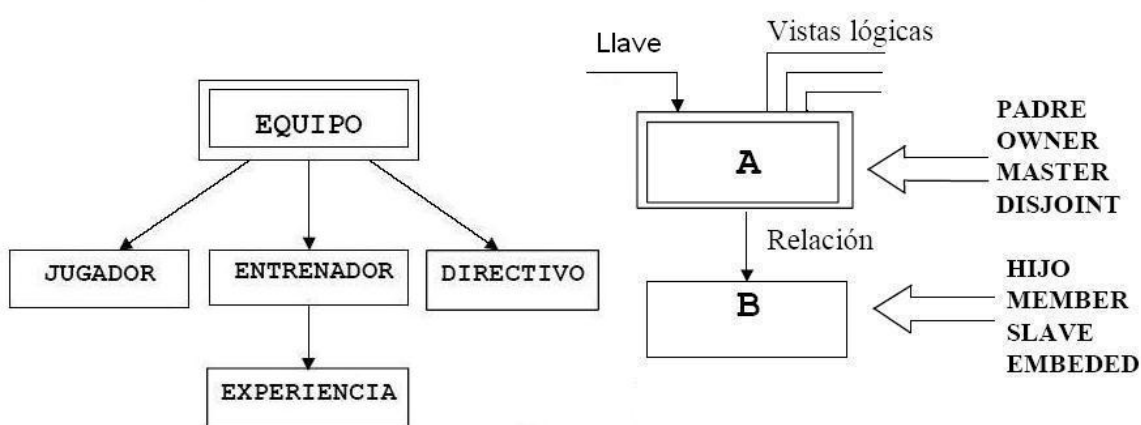


Figura 1.1: Modelo jerárquico (tomado de [Fernández 2006])

A mitad de los sesenta, se desarrolló IDS (Integrated Data Store: Almacén integrado de datos), de General Electric [Marqués Andrés, 2001] para satisfacer en parte la necesidad de representar relaciones entre datos más complejos que las que se podían modelar con los sistemas jerárquicos, y con visión de imponer un estándar de bases de datos. Este trabajo fue dirigido por uno de los pioneros en los sistemas de bases de datos: Charles Bachmann. IDS era un nuevo tipo de sistema de bases de datos conocido como sistema de red, que no consiste en que la base de datos está almacenada en una red de ordenadores, sino por la manera en la que los datos se enlazaban con otros datos. Se llama, por tanto, modelo en red porque representa los datos que contiene en la forma de una red de registros y conjuntos (en realidad listas circulares llamadas sets) que se relacionan entre sí, formando una red de enlaces. Para hacerlo utiliza registros, tipos de registro y tipos de conjunto, lo que produjo un gran efecto sobre los sistemas de información de aquella generación.

El modelo en red tampoco se utiliza casi hoy en día, y si subsiste es como consecuencia del mantenimiento de un sistema todavía no reconvertido o no portado a modelos y SGBD más modernos. Aunque este modelo permite más flexibilidad que el modelo jerárquico, y en algunos casos se adapta muy bien a algunos tipos de transacciones, se considera superado por otros modelos, como el relacional, o sumido en parte por modelos más modernos, como el objetual.

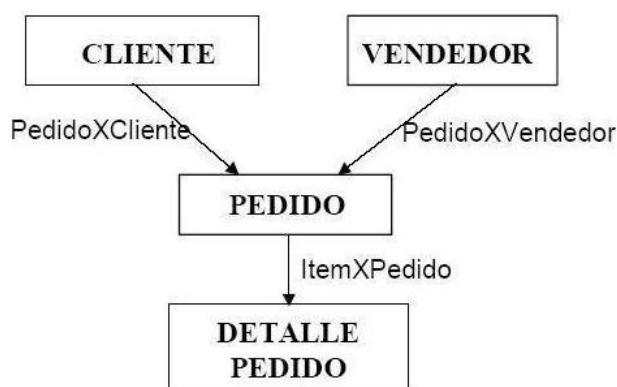


Figura 1.2: Modelo red (tomado de [Fernández 2006])

Los modelos jerárquicos y en red, con el paso de los años, se pueden considerar como modelos *puente* hacia el modelo relacional, ya que se incorporan en los primeros sistemas de gestión de bases de datos que introducen un mayor nivel de independencia, respecto a la estructura interna, pero estos sistemas presentan algunos inconvenientes: [Marqués Andrés, 2001]

- Es necesario escribir complejos programas de aplicación para responder a cualquier tipo de consulta de datos, por simple que ésta sea.
- la independencia de datos es mínima.
- no tienen un fuerte fundamento teórico.

lo que se resume en una estructura de cierto bajo nivel y de compleja manipulación.

1.2.2 Segunda generación de Sistemas Gestores de Bases de Datos.

Edgar F. Codd, de los laboratorios de investigación de IBM, no estaba satisfecho con los modelos jerárquicos y de red, por lo que empezó a trabajar en una serie de informes técnicos acerca de una manera ‘nueva’ de organizar y acceder a los datos. A partir de estos trabajos publicó el artículo “A Relational Model of Data for Large Shared Data Banks” en 1970 [Codd 1970] presentando el *modelo relacional*. Codd concibió un sistema donde el usuario sería capaz de acceder a la información con comandos parecidos al lenguaje natural y donde la información estuviera almacenada en ‘*Tablas*’

El modelo relacional, propuesto por Codd en 1970, en sus inicios no tuvo aceptación inmediata, debido en parte a la naturaleza técnica del artículo y a que fue considerado como una elegante teoría matemática sin posibilidad de implementación eficiente en productos comerciales, sin llegar a imaginarse que el mismo se convirtiese, en los años ochenta, en la segunda generación de los SGBD.

Surgieron, entonces, varios SGBD relacionales experimentales. Los primeros productos comerciales aparecieron a finales de los 70, principio de los 80. De real relevancia fue el “SYSTEM R”, de IBM San José Research Laboratory, en California. Este proyecto fue diseñado para probar la utilidad práctica del modelo relacional. Propusieron una implementación de sus estructuras de datos y operaciones, lo que condujo posteriormente a dos grandes desarrollos [Marqués Andrés, 2001]:

- El desarrollo de un lenguaje de consulta estructurado llamado SQL, el cual llegó a ser el lenguaje estándar para los SGBD relacionales.
- La producción de varios SGBD comerciales como DB2 y SQL/DS de IBM y ORACLE de Oracle Corporation.

El modelo relacional se distingue entre los modelos existentes en que representa al mundo real mediante tablas relacionadas entre sí por columnas comunes.

Sin embargo, también tiene sus fallos, siendo uno de ellos su limitada capacidad al modelar los datos y en tal sentido Chen, en 1975, presentó el modelo entidad-relación,

que es la técnica más utilizada en el diseño de bases de datos. Codd en 1979 intentó subsanar algunas de las deficiencias de su modelo relacional con una versión extendida denominada RM/T (1979) y más recientemente RM/V2 (1990).

A pesar de todo, este modelo es el más extendido en la actualidad. En los libros de hace poco más de una década, el modelo relacional aparecía a la par con los otros dos modelos tradicionales (también llamados navegacionales): el modelo en red o el modelo jerárquico [Ullman 1980] [Date 1981], [Everest 1986] [Gardarin 1988]. Sólo los libros explícitamente denominados “bases de datos relacionales” (p.ej. [Delobel & Adiba 1985] [Gardarin & Valduriez 1987]) se atrevían a centrarse en este modelo. En los libros genéricos sobre bases de datos, los modelos navegacionales han sido desplazados a apéndices o han desaparecido completamente de los mismos. Su lugar ha sido ocupado por el modelo objetual u objeto-relacional, aunque generalmente con menos profusión que el relacional [Date 1999] [Elsmasri & Navathe 2000]. También hay que destacar que estos dos últimos modelos tienen ciertas cosas en común con los navegacionales, lo que justifica en gran medida el concepto de “modelo de datos” como abstracción que recoge todos estos cambios tecnológicos.

1.2.3 Tercera y Cuarta Generación de Sistemas Gestores de Bases de Datos.

Como respuesta a la creciente complejidad de las aplicaciones que requieren bases de datos, han surgido dos nuevos modelos:

- El modelo de datos orientado a objetos
- El modelo relacional extendido.

Esta evolución representa la tercera generación de los SGBD.

Coincidiendo con la entrada de los lenguajes orientados a objetos como Smalltalk o C++ en el ámbito industrial, los investigadores se plantearon transportar estas ideas a las bases de datos, y fue [Copeland & Maier 1984] el primero en tratar el tema de las bases de datos orientadas a objetos con el sistema prototipo GemStone.

Una *base de datos orientada a objetos* es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

- Polimorfismo - Propiedad de una operación mediante la cual la misma puede ser aplicada a distintos tipos de objetos.

En este modelo la información sobre una entidad se almacena como un objeto persistente y no como una fila en una tabla, lo que en principio lo hace más eficiente en términos de requerimientos de espacio y asegura que los usuarios puedan manipular los datos sólo de las maneras en que el programador las haya especificado.

También es más eficiente en el uso de espacio de disco requerido para las consultas, ya que en vez de almacenar la consulta, simplemente se construye una serie de índices (punteros) a los objetos seleccionados. A esto hay que sumar las ventajas derivadas del modelo orientado a objetos, ya explotadas en sus lenguajes de programación, la mayor expresividad y su adecuación para almacenar muchos tipos de datos diferentes.

Sin embargo, las Bases de Datos Orientadas a Objetos no han superado en la práctica a las relacionales, lo que se debe en gran medida a tres aspectos fundamentales:

- Arrastran algunas de las propiedades no deseables de los modelos pre-relacionales. El programador tiene que tener mucha información sobre la estructura de los datos. Si se conocen las propiedades de los objetos, la consulta es rápida y simple. Pero la realidad es que en muchos casos se desconocen las identidades de los objetos. Lo que preocupa o interesa es almacenar los atributos de los objetos y relacionar los valores de ellos, aspecto en que el modelo relacional es más sencillo.
- El hecho de que las organizaciones sean capaces de alterar los métodos de bajo nivel utilizados en los SGBDO, hace que sea más difícil para terceros el hacer productos añadidos. Mientras que las bases de datos relacionales se pueden beneficiar del software realizado por otras firmas, los usuarios de SGBDO tienen que producir el software en casa para adaptarse a sus propias particularidades, parte de ellas incorporadas al comportamiento de los objetos.
- En tercer lugar y quizás más importante, es el hecho de que las organizaciones tiendan a ser conservadoras en relación con las bases de datos, uno de sus activos más valiosos. Muchas organizaciones, aunque utilizan lenguajes orientados a objetos como el C++ para las aplicaciones microinformáticas o aplicaciones específicas, desconfían de los lenguajes orientados a objetos en general por no considerarlos suficientemente estables para trabajar con

información crucial para la organización. Mito o realidad, el hecho es que no acaban de decidirse por utilizar un SGBDO y siguen aferrados al SQL para realizar sus informes y al SQL embebido para interrelacionar las aplicaciones con el SGBD, manteniendo una separación que consideran imprescindible.

El modelo E-R extendido pretende aportar soluciones a requerimientos un tanto más complejos no contemplados en el modelo E-R propuesto por Chen. Este modelo incluye todos los conceptos de Entidad-Relación, e incorpora los de Subclase y Superclase con los conceptos asociados de Especialización y Generalización. Otro nuevo concepto incluido por el “Entidad Relación Extendido” es el de Categoría. Asociado a estos conceptos está el importante mecanismo de Herencia de atributos.

El modelo Orientado a objetos (OO) ha guiado esta ampliación del modelo relacional, con el fin de lograr integrar lo mejor de los dos enfoques, pero cuenta con un tratamiento teórico pobre que inspira desconfianza y pasa a ser una incógnita el trabajo con el mismo, por lo que no muchos desarrolladores de bases de datos migran hacia el trabajo con este modelo.

Considerando los aspectos antes mencionados y coincidiendo con lo expresado en un artículo de Mario Piattini Velthuis [Piattini Velthuis, 2000] se concluye que esta última generación (la «tercera»), se caracteriza por proporcionar capacidades de gestión de datos, objetos y gestión de conocimiento y pretende responder a las necesidades de aplicaciones tales como: CASE (Ingeniería del software asistida por ordenador), CAD (Computer-Aided Design, diseño asistido por computador) /CAM (Computer-Aided Manufacturing, manufactura asistida por computador) /CIM (Computer-Integrated Manufacturing, manufactura integrada por computador), SIG (Sistemas de Información Geográfica), información textual, aplicaciones científicas, sistemas médicos, publicación digital, educación y formación, sistemas estadísticos, comercio electrónico, etc.

Durante la década de los 90, las necesidades de almacenamiento no se limitaron a la manipulación de datos, sino que era necesario tratar textos, gráficos, voces y videos, tampoco la información almacenada se limitaba a estar centralizada sino que se quería romper con las fronteras físicas para almacenar la información, así como otros elementos a los que SGBD tuvieron que enfrentar y que dan origen a la cuarta generación de las Bases de Datos.

Las Bases de datos distribuidas se incluyen como parte de esta generación, a pesar de que se vienen tratando desde la generación anterior pero es en esta última (la «cuarta»), donde el uso de las bases de datos distribuidas ha aumentado considerablemente, debido al aumento de la descentralización, debido al abaratamiento, ancho de banda y flexibilidad de las redes de computadores.

Se estaría en presencia de una base de datos distribuida cuando se hable de un único sistema de gestión de bases de datos que actúa sobre distintos ordenadores distribuidos y gestionando la misma base de datos, en caso que cada sistema esté formado por varios SGBD, se suele hablar de Sistemas de Múltiples Bases de Datos [Hernández 2002], la esencia de estos sistemas esta en que los usuarios no deben percatarse de esta dispersión espacial de los datos, es decir, los usuarios deben percibir lo mismo que si trabajaran con un único sistema centralizado.

Esta generación contempla además las bases de Datos documentales, nacida de la evolución de los sistemas de almacenamiento, donde los documentos contenidos pueden abarcar datos, gráficos, imágenes estáticas (fotografía, grabados, dibujos), imágenes en movimiento, (vídeo, animación) voz, sonidos, música y cualquier tipo de información.

Se anexan a esta generación de bases de datos aquellas que incluyen sistemas de inferencia que permiten el "aprendizaje", como base de los sistemas expertos, por lo que también son conocidas como "bases de conocimientos". Se trata de sistemas que almacenan reglas, además de datos [Hernández 2002]. Estos sistemas son capaces además de aplicar estas reglas consecuentemente a las situaciones que se les plantean. La aplicación más directa puede parecer la ayuda al diagnóstico médico, pero se usan también en análisis de fallos en la industria. La creación de una base de conocimiento es un proceso lento y costoso, debido a que el conocimiento (el conjunto de reglas) se ha de incorporar manualmente siendo este el mayor inconveniente de las bases de conocimientos.

Resulta de gran interés en esta generación los almacenes de datos (data-warehouses) donde su principal objetivos esta dado en la integración de los datos de una empresa en un repositorio único sobre el que los usuarios pueden hacer fácilmente consultas, informes y análisis y suele funcionar en relación con los sistemas de toma de decisión (DSS, Decision Support System) y los Sistemas de Información Ejecutiva (EIS, Executive Information System). Es relevante constatar que el uso de los almacenes de

datos es el original de los sistemas de información: recopilar datos para analizarlos y tomar decisiones a partir de este análisis.

1.3 Metodologías de diseño de base de datos.

Solo con la tecnología no es suficiente para que los sistemas actuales de Bases de Datos funcionen mejor que los de hace unos años. Vinculado a las tecnologías suelen asociarse unas metodologías que intentan sacar provecho de las primeras. Utilizar un sistema de gestión de bases de datos relacional no es por sí solo una garantía de que la base de datos que se construya utilizándolo vaya a funcionar bien. De hecho, la simplicidad del modelo relacional y de algunos SGBDR, han llevado a profesionales y no profesionales del área informática a la creación de verdaderos desastres, causando casi más problemas de los que resuelven.

Los siguientes párrafos tomados del libro Batini, Ceri y Navathe (1994), al referirse a las metodologías de diseños de bases de datos exponen: [C.Batini, S. Ceri, S.B. Navathe 1994].

“Desafortunadamente, las metodologías de diseño de bases de datos no son muy populares; la mayoría de las organizaciones y de los diseñadores individuales confía muy poco en las metodologías para llevar a cabo el diseño y esto se considera, con frecuencia, una de las principales causas de fracaso en el desarrollo de los sistemas de información. Debido a la falta de enfoques estructurados para el diseño de bases de datos, a menudo se subestiman el tiempo o los recursos necesarios para un proyecto de bases de datos, las bases de datos son inadecuadas o ineficientes en relación a las demandas de la aplicación, la documentación es limitada y el mantenimiento es difícil”.

“Muchos de estos problemas se deben a la falta de una claridad que permita entender la naturaleza exacta de los datos, a un nivel conceptual y abstracto. En muchos casos, los datos se describen desde el comienzo del proyecto en términos de las estructuras finales de almacenamiento; no se da peso a un entendimiento de las propiedades estructurales de los datos que sea independiente de los detalles de la realización”.

Lo anterior insta a convertirse en una condición insoslayable involucrarse en el tema de las metodologías de diseño de bases de datos, antes de encarar la construcción de

una base de datos.

1.3.1 Fases de diseño de Bases de Datos.

El diseño de una base de datos consta de tres fases:

- Diseño *conceptual*, que no es más que la producción de un esquema conceptual independiente de todas las consideraciones físicas.
- Diseño *lógico*, constituye un refinamiento de la fase anterior eliminando las construcciones que no se pueden representar en el modelo de base de datos escogido (relacional, orientado a objetos, etc.).
- Diseño *físico*, es una traducción del esquema lógico al físico para el SGBD escogido. La fase de diseño físico considera las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar un acceso eficiente a la base de datos en memoria secundaria.

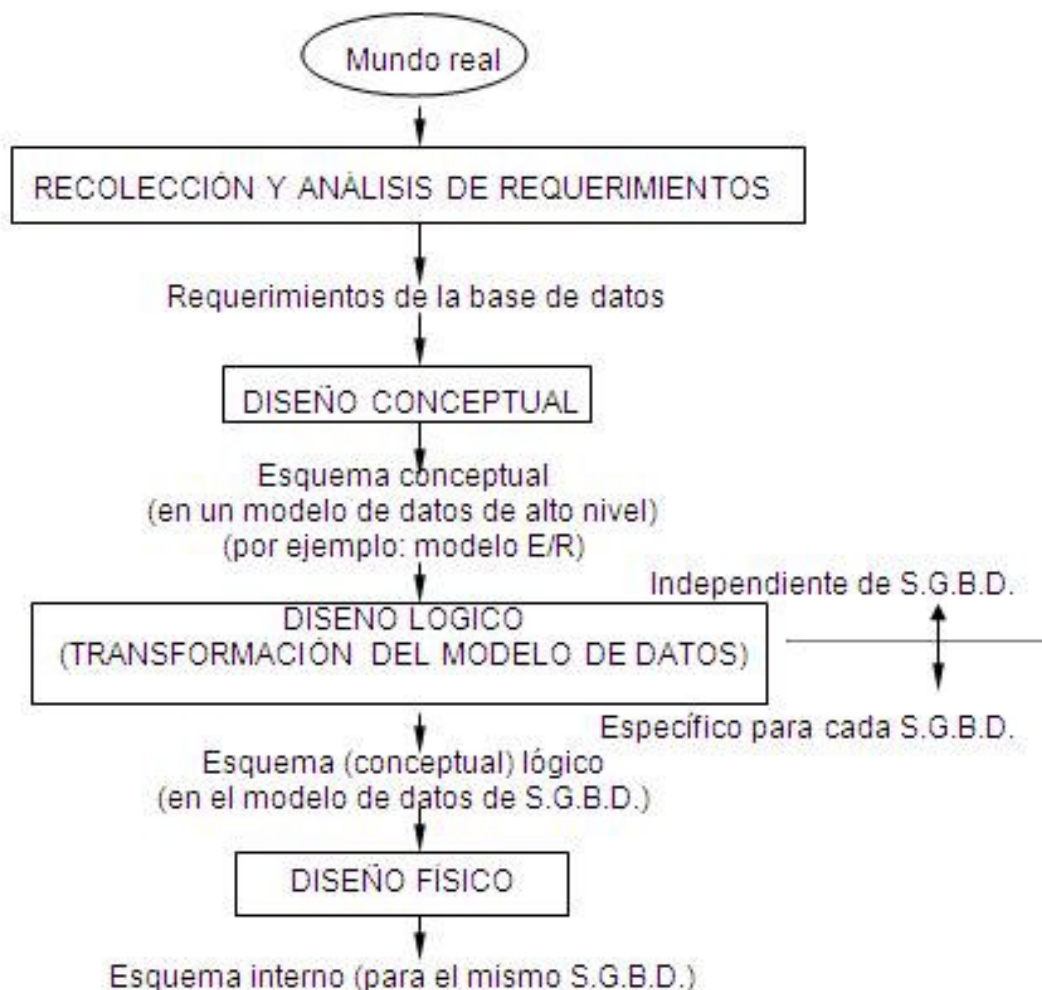


Figura 1.3: Los pasos en el diseño de una base de datos (tomado de [Ro y Norick 2003])

1.4 Estrategias de diseño de bases de datos.

Son múltiples las estrategias a seguir para realizar el diseño: de abajo a arriba, de arriba a abajo, de dentro a fuera y la estrategia mixta.

La estrategia *de abajo a arriba* parte de todos los atributos de la información y los va agrupando en entidades y relaciones. Ideal para las bases de datos simples, con pocos atributos.

La estrategia *de arriba a abajo* es más apropiada cuando se trata de bases de datos complejas. Se comienza con un esquema con entidades de alto nivel, que se van refinando para obtener entidades de bajo nivel, atributos y relaciones.

La estrategia *de dentro a fuera* es similar a la de abajo a arriba, pero difiere en que se parte de los conceptos principales y se va extendiendo el esquema para considerar también otros conceptos, asociados con los que se han identificado en primer lugar.

La estrategia *mixta* utiliza ambas estrategias, de abajo a arriba y de arriba a abajo, con un esquema de divide y vencerás. Se obtiene un esquema inicial de alto nivel, se divide en partes, y de cada parte se obtiene un sub-esquema. Estos sub-esquemas se integran después para obtener el modelo final.

1.5 El Proceso Unificado de Rational y los diseñadores de base de datos.

La obtención de un software profesional hace necesario acudir a la ingeniería que acoge los aspectos relacionados con las teorías, métodos y herramientas para el desarrollo de software, siendo esta la ingeniería de software, que implementa mediante el Proceso Unificado de Rational (RUP, el original inglés *Rational Unified Process*) un conjunto de mejores prácticas para el desarrollo de aplicaciones informáticas.

RUP utiliza como notación en sus modelos el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*), un dúo que se ha establecido hace ya alrededor de unos 10 a 12 años desde el momento en que fueron presentados por sus autores Jacobson, Rumbaugh y Booch y se ha establecido en la comunidad internacional como uno de los estándares principales en el desarrollo de software.

RUP en su modelación define como sus principales elementos: *trabajadores, actividades, artefactos y flujos de trabajo*. Los trabajadores definen el comportamiento

y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Analistas de sistemas, arquitectos, diseñadores de base de datos, integradores de sistemas, son algunos de los trabajadores que define RUP, de ellos se centrará la atención sobre el tercero de los antes mencionados («diseñadores de base de datos»).

Los diseñadores de la base de datos realizan el diseño lógico de la base de datos, debiendo identificar los *datos*, las *relaciones* entre datos y las *restricciones* sobre los datos. Es tarea del diseñador de bases de datos tener un profundo conocimiento de los datos de la empresa y también debe conocer sus reglas de negocio. Las reglas de negocio describen las características principales de los datos, tal como las ve la empresa.

Si se desean obtener resultados satisfactorios, el diseñador de la base de datos debe implicar en el desarrollo del modelo de datos a todos los usuarios de la base de datos, tan pronto como sea posible. El diseño lógico de la base de datos es independiente del SGBD concreto que se vaya a utilizar, es independiente de los programas de aplicación, de los lenguajes de programación y de cualquier otra consideración física. El diseñador debe seguir un proceso que es imprescindible para desarrollar una base de datos relacional bien diseñada y, al principio, hay un camino largo a recorrer antes de crear las tablas de la aplicación. No se refiere a un largo camino en cuanto a tiempo, sino ciertamente en pensamiento. Un enfoque sistemático del diseño ayudará al diseñador, le ahorrará mucho tiempo y se acercará mucho más a lo que el cliente realmente necesita.

Que el diseñador realice su función de forma correcta será el cimiento para la construcción de una gran pirámide del cual él constituye la base. En un primer piso se sitúa el administrador de la base de datos, el que se encarga del diseño físico de la base de datos y de su implementación, realiza el control de la seguridad y de la concurrencia, mantiene el sistema para que siempre se encuentre operativo y se encarga de que los usuarios y las aplicaciones obtengan buenas prestaciones. Por su parte, en un segundo nivel, los programadores de aplicaciones son los que se encargarán de implementar los programas de aplicación que servirán a los usuarios finales. Estos programas de aplicación son los que permiten consultar datos, insertarlos, actualizarlos y eliminarlos. Y finalmente, en la cúspide de la pirámide, los usuarios finales que están constituidos por los clientes de la base de datos, ya que la

misma ha sido diseñada e implementada, y estará siendo mantenida, para satisfacer los requisitos en la gestión de su información.

1.6 ERStudio, herramienta CASE para el diseño de Base de Datos.

Desarrollar sistemas o herramientas que faciliten encontrar técnicas que permitan incrementar la productividad y el control de la calidad en cualquier proceso de elaboración de software, ha sido el principal área de desarrollo durante las última década. Las herramientas CASE (Computer Aided Software Engineering) reemplaza al papel y al lápiz por el ordenador para transformar la actividad de desarrollar software en un proceso automatizado.

Herramientas como Rational Rose Enterprise Edition, Erwin o el ERStudio pudieran citarse como resultado de lo antes mencionado, especial interés para el desarrollo de este trabajo lo constituye el ERStudio, ya que es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejas.

ERStudio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes. Soporta el proceso de diseño iterativo inherente en el ciclo de vida de la aplicación.

Las capacidades de diseño que contiene, ayudan a crear un diseño lógico que puede transformarse en cualquier número de diseños físicos. ER/Studio revisa la normalización y la compilación con la sintaxis de la plataforma de la base de datos. Se pueden desplegar los modelos de datos usando la notación IDEF1X o IE.

ER/Studio soporta el muy popular SQL y Bases de Datos de escritorio, incluyendo:

- Oracle 7.3® , 8.x & 9i
- Sybase® System 11.9.2, 12.x & 12.5
- Microsoft® SQL Server 6.5, 7 & 2000
- IBM® DB/2® Universal Database® 4.x, 5.x, 6.x & 7.x for Open Systems, OS/390® & AS/400 4.5
- Informix® OnLine and SE

- SQL Anywhere™ and Watcom™ SQL
- InterBase® 4
- Microsoft Access 2.0, 95, 97 & 2000
- Microsoft Visual FoxPro®

1.7 Tendencias de las Bases de Datos.

Peter Ferdinand Drucker, escritor y consultor estadounidense, al referirse al futuro expresó: *“La mejor manera de predecir el futuro es crearlo”*.

Sobre esta base, sería razonable que en los próximos diez años maduraran algunas de las tecnologías citadas y que todavía hoy se encuentran poco desarrolladas. Esta madurez se producirá en los tres planos siguientes [DeMiguel y Piattini, 1999]:

- Plano *científico*, es decir, la investigación dedicada a la tecnología en sí.
- Plano *industrial*, en cuanto al desarrollo de productos que empleen la tecnología por parte de suministradores.
- Plano *comercial*, es decir, la aceptación que tiene la tecnología y su utilización por parte de los usuarios.

Entonces, serían visibles avances espectaculares en tecnologías como las bases de datos orientadas a objetos, multimedia, distribuidas, deductivas, paralelas, en tiempo real, almacenes de datos, etc. Todo ello implicaría la extensión de los modelos de datos existentes (tanto a nivel conceptual como lógico) o la aparición de otros nuevos, sobre la base del paradigma objetual.

Sin lugar a dudas, la difusión de la tecnología de bases de datos será mucho mayor, lo que hará posible acceder de forma fácil a la información en cualquier momento, en cualquier lugar y desde cualquier equipo (bases de datos móviles) a todo tipo de datos, utilizando herramientas de consulta inteligentes que ayuden al usuario en la obtención de información.

Si la visión se hace un poco mas allá de la primera década, entonces las bases de datos relacionales ya se habrán convertido en sistemas heredados y se estaría conjeturando demasiado sobre la situación futura de las bases de datos.

1.7 Conclusiones.

En este capítulo se enuncian las principales definiciones de Base de Datos y se muestra la relación existente entre los Sistemas Gestores de Bases de Datos y las bases de datos en sí, así como el papel decisivo de los diseñadores de Base de Datos que con el cumplimiento de las metodologías de diseño, estrategias de diseño y demás aspectos tratados permitirá obtener una base de datos caracterizada por:

- Integridad de toda la información de la organización.
- Persistencia de los datos.
- Acceso simultáneo para distintos usuarios (o aplicaciones).
- Mecanismos para asegurar la integridad y seguridad de los datos.

características que son apreciadas por cualquier aplicación.

El marco de trabajo RUP, escogido por sus excelentes guías para desarrolladores noveles, ha sido nuestra propuesta para llevar a cabo el proyecto en general y en particular para la obtención de los artefactos relacionados con la base de datos.

Se describe el futuro de las Bases de Datos, el cual invita a reflexionar profundamente sobre las repercusiones que pueden tener estas tecnologías tanto para los administradores de bases de datos como para los diseñadores y desarrolladores, los cuales dispondrán de nuevos entornos de trabajo más potentes.

CAPÍTULO II: Diseño Lógico de la base de datos del “Sistema de Gestión de Inventario Y Almacenes” (SIGIA).

2.1 Introducción.

En este capítulo se presenta una solución a la problemática planteada en la introducción, que toma como base el estado del arte plasmado en el marco teórico referencial y que consiste en el diseño lógico de la base de datos para el Sistema de Gestión de Inventarios y Almacenes (SIGIA). Primero se realiza una descripción del problema que se va a resolver, explicando el conjunto de procesos del negocio relacionados con la gestión de los inventarios. Posteriormente, de las operaciones contenidas en estos procesos se identifican las entidades del diseño lógico de los datos.

Se muestra el esquema del diseño lógico obtenido utilizando ER/Studio v 6.01.

2.2 Definición de estándar de diseño

2.2.1 Formato de las entidades

El nombre para cada entidad tiene el formato que se describe a continuación:

Encabezamiento, estará compuesto por las letras N, D, o R y luego guion bajo o underescolar y posterior el nombre de la entidad; el cual expresa claramente su función en el modelo. En caso que el nombre de la entidad este compuesto por una sola palabra esta debe comenzar con mayúscula, en el caso que contenga mas de una palabra el nombre, las mismas deben unirse y escribir con letra mayúscula el empezar de cada una de ellas, ejemplos:

Entidad: N_Seccion, donde N: significa nomenclador, _: realiza la función de separador y Seccion: nombre que identifica la entidad dentro del negocio.

Entidad: D_AjusteInventario, donde D: significa datos, _: realiza la función de separador y AjusteInventario: nombre que identifica la entidad dentro del negocio.

Entidad: R_ProductoAjusteEntrada, donde R: significa relación, _: realiza la función de separador y ProductoAjusteEntrada : nombre que identifica la entidad dentro del negocio.

Las entidades que se corresponden con Documentos primarios que se utilizan en la gestión de inventarios el color de fondo de la entidad debe ser azul claro para que sean identificados del resto de las entidades del modelo que su color de fondo es blanco.

2.2.2 Formato de los atributos

El nombre de los atributos comienza con mayúsculas y estos debe ser claros y expresar claramente su significado.

En caso de que el encabezado del atributo sea ID, significa que es un identificador ese atributo con respecto al resto de los atributos que componen esa entidad, El color Rojo de la fuente de cada atributo indica que constituyen llave primaria y el color azul de la fuente de cada atributo indica que constituye llave foránea, el color de fuente negro indica que es una característica de la entidad.

2.3 Descripción del problema.

El siguiente esquema representa la dinámica del manejo de inventarios de almacenes de cualquier empresa.

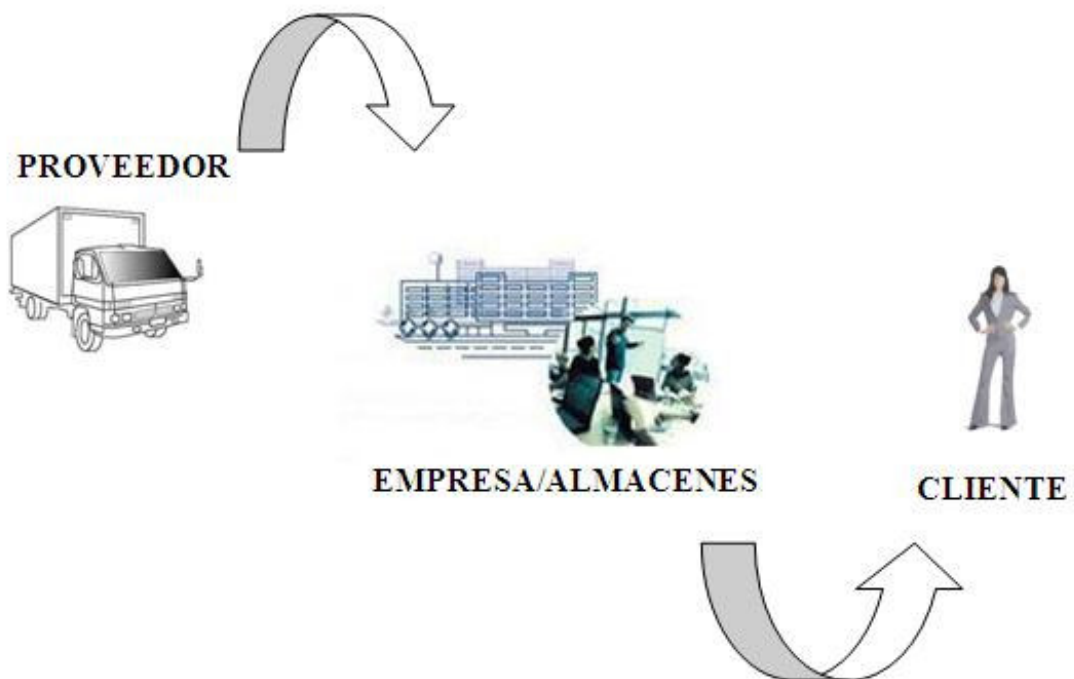


Figura 2.1 Diagrama de funcionamiento de una empresa y sus almacenes.

CAPÍTULO II

La empresa tiene asociado un conjunto de almacenes, cada almacén está compuesto por un conjunto de secciones de almacén (por ejemplo: Sección Central, Sección de Gastronomía, Sección de insumos, etc.).

A cada una de estas secciones se les da el tratamiento de almacén, lo que hace que para el desarrollo de este trabajo sea tratada la sección como un almacén.

Para identificar la ubicación física de un producto en una sección, se ha incluido en el código de cada sección una identificación de la ubicación geográfica de la misma como almacén, y además una identificación del estante y de la casilla.

La Empresa utiliza *Proveedores* a los que se compran productos que pueden ser materias primas o materiales, etc., pudiendo contar o no con *Áreas Productoras* que fabrican *productos terminados*, los cuales son entrados a los almacenes para su posterior venta o insumo. Estas áreas productoras *solicitan materiales* a los almacenes para fabricar los productos terminados o para consumo. La Empresa puede comercializar sus productos terminados o los que ha comprado a proveedores.

Se debe cumplir con el requisito de poder valorar los inventarios con un método de valoración adecuado, tanto en moneda nacional como en moneda libremente convertible.

Los métodos de valoración de inventario son:

- Método “FIFO”: El precio de costo del *primer* producto entrado define el precio que se toma como base para costear las primeras salidas.
- Método “LIFO”: El precio de costo del *último* producto entrado define el precio que se toma como base para costear las primeras salidas.
- Método “PONDERADO”: Para valorar las existencias se toma el promedio de los precios de costo con que han entrado y salido los productos

Entre las operaciones básicas sobre los inventarios se pueden citar:

1. Entradas de productos.
2. Salidas de productos.
3. Solicitud de materiales desde un departamento solicitante.
4. Transferencia de productos entre secciones.
5. Devolución de productos.
6. Ajustes de Inventarios.
7. Entradas de productos elaborados.

Cada una de estas operaciones tiene asociado un proceso de negocio, que se describe a continuación. De la comprensión del proceso del negocio se identificó las entidades, sus relaciones, atributos fundamentales, se determinaron los dominios de los atributos y los identificadores de cada entidad. Por último se describe todo el modelo lógico en forma de diccionario de datos y se muestra el diagrama en ER/Studio.

2.3.1 Entrada de productos por compra a proveedores.

El proceso de entrada de productos al almacén debido a una compra a proveedor, transcurre de la siguiente manera:

1. Se registra la Factura de Compra (Ver anexo 13).
2. Se produce el documento primario “Informe de Recepción”, sin especificar las cantidades de los productos (Ver anexo 1, 6, 10, 14).
3. El almacenero completa el Informe de Recepción con las cantidades realmente recibidas.
4. Se define la ubicación donde se van a colocar los productos.
5. Se registran qué productos y sus cantidades se les va a dar entrada en la sección.

Las entidades fuertes que se requieren para que la aplicación informaticé este proceso son:

1. D_FacturaCompra

Entidad:	D_FacturaCompra:	
Descripción	Entidad donde se almacenan los datos del encabezado de la factura de compra.	
Atributos:	Tipo	Descripción de los Atributos:
NumeroFC	Varchar	Numero del modelo de la factura de compra
IDProveedor	Integer	Identificador del proveedor
IDComprador	Integer	Identificador del comprador
Fecha	Date	Fecha del modelo de factura de compra
ArancelGeneral	Float	Recargo que tiene la factura por impuestos que ha transferido el importador.
DescuentoComercial	Float	Descuento comercial que ofrece el proveedor por la actitud mantenida para la empresa

IDMonedaArancel	Integer	Identificador de la moneda en la que está expresado el Arancel General
IDMonedaDescuento	Integer	Identificador de la moneda en la que está expresado el Descuento.
Observaciones		

2. D_Documento

Entidad:	D_Documento	
Descripción	Es una entidad base para todo tipo de documento. Almacena el nombre del documento primario, el tipo y la fecha en que se emite.	
Atributos:	Tipo	Descripción de los Atributos:
IDDocumento	Integer	Identificador de documento
Nombre	Char	Nombre del documento
Fecha	Date	Fecha de emisión del documento
Observaciones		

3. D_InformeRecepcion

Entidad:	D_InformeRecepcion:	
Descripción	Almacena los datos del encabezado del Informe de Recepción.	
Atributos:	Tipo	Descripción de los Atributos:
IDDocumento	Integer	Identificador del documento de Informe de Recepción.
NumIR	Integer	Número del modelo de Informe de Recepción.
NombreRecepciona	Varchar	Nombre y apellidos de la persona que elabora el Informe de Recepción.
NumeroFC	Varchar	Número del modelo de la factura de compra
Observaciones		

4. D_Entradas

Entidad:	D_Entradas	
Descripción	Registra las entradas de los productos a una sección. Es una entidad base que se relaciona con todos los tipos de entradas posibles	
Atributos:	Tipo	Descripción de los Atributos:
IDEntrada	Integer	Identificador para la entrada de un producto en una sección.
FechaEntrada	Date	Fecha de entrada de ese producto.
CodigoSeccion	Varchar	Código de la sección que está recibiendo el producto.
IDDocumento	Integer	Identificador del documento que generó la entrada.
Observaciones		

5. D_Comprador

Entidad:	D_Comprador	
Descripción	Recoge los datos de la persona autorizada a comprar.	
Atributos:	Tipo	Descripción de los Atributos:
IDComprador	Integer	Identificador del comprador
Nombre	Varchar	Nombre y apellidos del comprador
Direccion	Varchar	Dirección del comprador
Telefono	Varchar	Teléfono del comprador
Observaciones		

6. D_Proveedor

Entidad:	D_Proveedor	
Descripción	Recoge los datos del suministrador que provee o abastece a la entidad o empresa a la que pertenezca la sección o almacén.	

Atributos:	Tipo	Descripción de los Atributos:
IDProveedor	Integer	Identificador del proveedor
Nombre	Varchar	Nombre del proveedor (persona natural o jurídica)
Direccion	Varchar	Dirección del proveedor
Telefono	Varchar	Teléfono del proveedor
NumeroCuenta	Numeric	Número de cuenta bancaria del proveedor
SucursalCuenta	Numeric	Sucursal bancaria en que opera la cuenta del proveedor
NIT	Numeric	Número de Identificación Tributaria (NIT).
Observaciones		

7. D_MedioTransporte

Entidad:	D_MedioTransporte	
Descripción	Almacena la información del medio de transporte que se utiliza para llevar la mercancía al almacén.	
Atributos:	Tipo	Descripción de los Atributos:
IDVehiculo	Integer	Identificador del vehículo
MedioTransporte	Varchar	Especificar la chapa si es un medio de transporte sobre carretera; en caso de trenes, especificar el vagón donde se transporta la mercancía, si es avión especificar el vuelo y en caso de ser en barco especificar el nombre del mismo.
Descripcion	Varchar	Elementos que se deseen resaltar sobre el medio en que se transportan los productos
Observaciones		

8. D_Transportador

Entidad:	D_Transportador	
Descripción	Datos del transportador que conduce la mercancía al almacén.	
Atributos:	Tipo	Descripción de los Atributos:
CarnetIdentidad	Numeric	Carnet de identidad perteneciente al transportador
Nombre	Varchar	Nombre y apellidos del transportador

Direccion	Varchar	Dirección del transportador
Observaciones		

Las entidades que representan interrelaciones son:

9. R_ProductoFacturaCompra

Entidad:	R_ProductoFacturaCompra:	
Descripción	Relaciona los productos contenidos en la factura.	
Atributos:	Tipo	Descripción de los Atributos:
NumFC	Varchar	Numero de la factura de compra
IDProducto	Integer	Identificador del producto.
Cantidad	Float	Cantidad del producto
CodigoUM	Varchar	Código de la unidad de medida
PrecioMN	Float	Precio del producto en Moneda Nacional
PrecioMLC	Float	Precio del producto en moneda libremente convertible
IDMoneda	Integer	Identificador de la moneda libremente convertible
DescuentoMN	Float	Por ciento de descuento en Moneda nacional para ese producto
DescuentoMLC	Float	Por ciento de descuento en Moneda libremente convertible para ese producto
Arancel	Float	Arancel que se aplica a cada producto
Observaciones		

10. R_VehiculoTransportadorFactura

Entidad:	R_VehiculoTransportadorFactura	
Descripción	Relaciona el chofer del vehículo que transportó la mercancía amparada por la factura de compra.	
Atributos:	Tipo	Descripción de los Atributos:
IDVehiculo	Integer	Identificador del vehículo
CarnetIdentidad	Numeric	Carnet de identidad del transportador
NumFC	Varchar	Número del modelo de la factura de compra
Observaciones		

11. R_ProductoInformeRecepcion

Entidad:	R_ProductoInformeRecepcion:	
Descripción	Relaciona los productos contenidos en el Informe de Recepción.	
Atributos:	Tipo	Descripción de los Atributos:
IDEntrada	Integer	Identificador para la entrada de un producto
IDProducto	Integer	Identificador del producto
IDDocumento	Integer	Identificador del Informe de recepción.
Cantidad	Float	Cantidad de cada producto asociado al informe de recepción
CodigoUM	Varchar	Código de la unidad de medida
FechaVencimiento	Date	Fecha de vencimiento de ese producto
PrecioUnitarioMN	Float	Precio por unidad del producto en moneda nacional
PrecioUnitarioMLC	Float	Precio por unidad del producto en moneda libremente convertible.
IDMonedaMLC	Integer	Identificador de la moneda libremente convertible
Observaciones		

12. R_ProductoFechaVencimiento

Entidad:	ProductoFechaVencimiento	
Descripción	Existencia de productos por fecha de vencimiento.	
Atributos:	Tipo	Descripción de los Atributos:
IDProducto	Integer	Identificador del producto
CodigoSeccion	Varchar	Código de la sección
FechaVencimiento	Date	Fecha de vencimiento del producto
Cantidad	Float	Cantidad del producto
CodigoUM	Varchar	Código de la Unidad de Medida
ImpteExistMN	Float	Importe de la existencia del producto en moneda nacional, según el método de valoración
ImpteExistMLC	Float	Importe de la existencia del producto en moneda

		libremente convertible, según método de valoración
IDMoneda	Integer	Identificador de la moneda en la que esta expresado el valor del producto.
Observaciones		

13. R_ProductoSeccion

Entidad:	ProductoSeccion	
Descripción	Refleja la existencia total de productos por cada sección de almacén.	
Atributos:	Tipo	Descripción de los Atributos:
IDProducto	Integer	Identificador del producto
CodigoSeccion	Varchar	Código de la sección
ExistenciaGeneral	Float	Existencia General del producto
CodigoUM	Varchar	Código de la Unidad de Medida
ImpteExistMN	Float	Importe de la existencia del producto en moneda nacional, según método de valoración
ImpteExistMLC	Float	Importe de la existencia del producto en moneda libremente convertible, según método de valoración
IDMoneda	Integer	Identificador de moneda en que está expresado el valor del producto.
StockMin	Float	Stock mínimo que se requiere de ese producto en esa sección
StockMax	Float	Stock máximo que se requiere de ese producto en esa sección
FechaHora	DateTime	Fecha/Hora en que se realizó la última actualización.
Observaciones		

En este proceso intervienen las entidades N_Seccion (tabla 43) y N_TipoProducto (tabla 41), que constituyen los nomencladores para las Secciones y para los Productos, respectivamente. Nota: Todos los nomencladores son descritos en el epígrafe 2.4 de este capítulo.

CAPÍTULO II

El diseño lógico que representa las necesidades de almacenamiento de información del proceso de entrada de productos al almacén debido a una compra a proveedor, se presenta en las siguientes figuras.

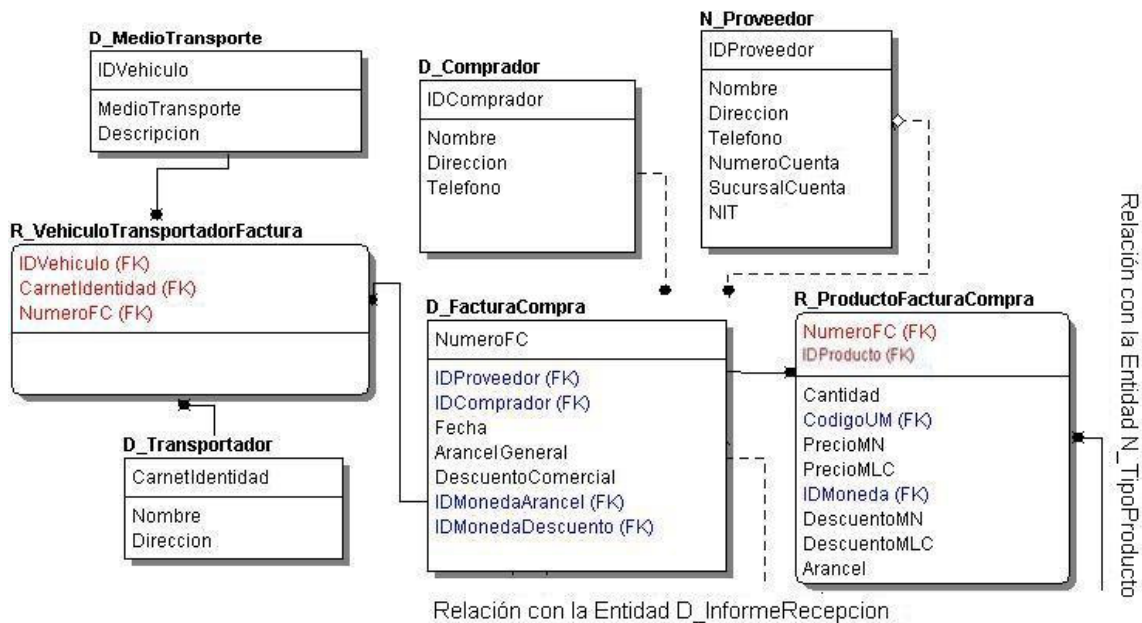


Figura 2.1: Diseño lógico para el proceso de Entrada de Productos por Compra.

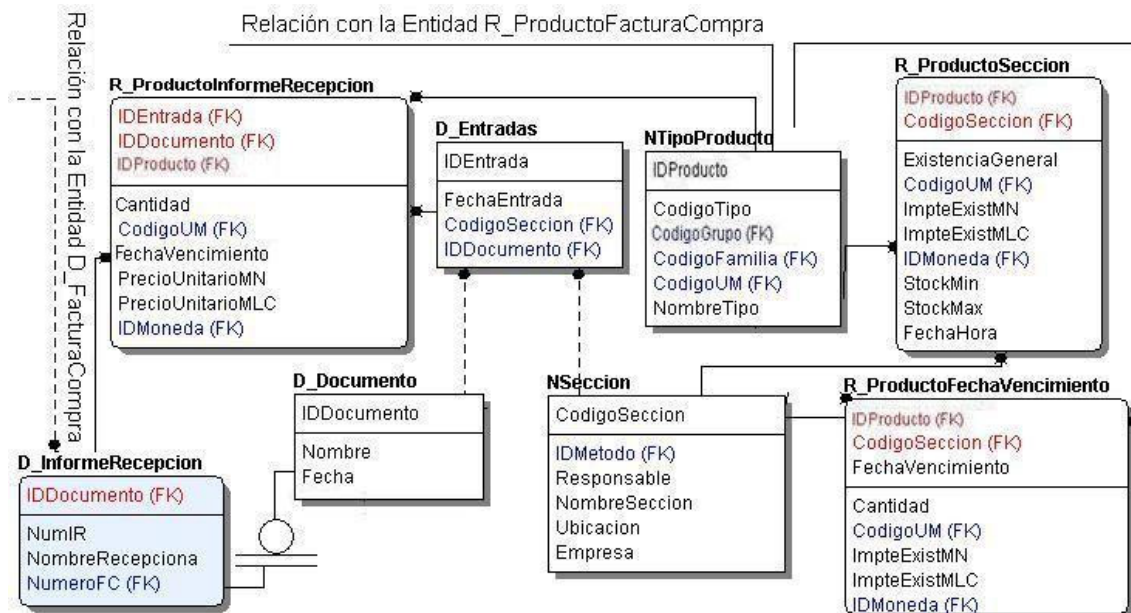


Figura 2.2: Diseño lógico para el proceso de Entrada de Productos por Compra (continuación figura2.1).

2.3.2 Salida de productos.

El objetivo de este proceso es formalizar las salidas de productos, ya sean para insumos o ventas, así como las entregas de productos en consignación o en depósito.

El proceso de salida de productos transcurre de la siguiente manera:

1. Se produce el documento primario “Vale de Salida”, el cual contiene el destinatario del producto, cada producto con su cantidad, costo y precio de venta cuando proceda. (Ver anexo 5, 9, 10)
2. El sistema genera la Factura de Venta cuando se refiera a una operación de venta, teniendo en cuenta los precios de venta establecidos para los productos.
3. Se define la sección de donde se va realizar la salida de los productos.

Las entidades fuertes que se requieren para que la aplicación informatices este proceso son:

14. D_ValeSalida

Entidad:	D_ValeSalida	
Descripción	Recoge los datos del encabezado del vale de salida.	
Atributos:	Tipo	Descripción de los Atributos:
IDDocumento	Integer	Identificador del documento de Vale de salida
NumSalida	Integer	Número del modelo de la factura de compra
TipoSalida	Varchar	Especifica el tipo de salida, ejemplo: salida para insumo, ajuste de inventario, venta, etc.
Destinatario	Varchar	Destino que lleva la salida de productos.
Observaciones		

15. D_PreciosdeVenta

Entidad:	D_PreciosdeVenta	
Descripción	Contiene los precios de venta que se establecen para cada producto.	
Atributos:	Tipo	Descripción de los Atributos:
IDProducto	Integer	Identificador del producto
CodigoUM	Varchar	Código de la Unidad de Medida

PrecioMN	Float	Precio de Venta de cada unidad del producto en moneda nacional
PrecioMLC	Float	Precio de Venta de cada unidad del producto en moneda libremente convertible.
IDMoneda	Integer	Identificador de la moneda en la que está expresado el valor del producto.
Observaciones		

16. D_FacturaVenta

Entidad:	D_FacturaVenta	
Descripción	Almacena la información sobre el encabezado de la factura de venta.	
Atributos:	Tipo	Descripción de los Atributos:
IDDocumento	Integer	Identificador del documento de factura de venta
NumFV	Integer	Número del modelo de la factura de Venta
IDCliente	Integer	Identificador del cliente
Observaciones		

17. D_Cliente

Entidad:	D_Cliente	
Descripción	Recoge los datos de los clientes a quienes se les vende.	
Atributos:	Tipo	Descripción de los Atributos:
IDCliente	Integer	Identificador del cliente
Nombre	Varchar	Nombre del cliente (persona natural o jurídica)
Direccion	Varchar	Dirección del cliente
Observaciones		

18. D_Salida

Entidad:	D_Salida	
Descripción	Recoge la información de las salidas de productos que se realicen por la sección. Es una entidad base que se relaciona con todos los tipos de salidas posibles.	
Atributos:	Tipo	Descripción de los Atributos:
IDSalida	Integer	Identificador de la salida
FechaSalida	Date	Fecha en que se produce la salida.
CodigoSeccion	Varchar	Código de la sección de la que se produce la salida.
IDDocumento	Integer	Identificador del documento que produce la salida, en este caso el vale de salida.
Observaciones		

Las entidades que representan interrelaciones son:

19. R_ProductosSalida

Entidad:	R_ProductosSalida	
Descripción	Relaciona los productos contenidos como detalle del vale de salida.	
Atributos:	Tipo	Descripción de los Atributos:
IDSalida	Integer	Identificador de la salida
IDProducto	Integer	Identificador del producto
IDDocumento	Integer	Identificador del documento que produce la salida, en este el vale de salida.
Cantidad	Float	Cantidad del producto
CodigoUM	Varchar	Código de la Unidad de Medida
PrecioMN	Float	Precio de costo por cada unidad del producto en moneda nacional
PrecioMLC	Float	Precio de costo por cada unidad del producto en moneda libremente convertible.
IDMoneda	Integer	Identificador de la moneda en la que está expresado el valor del producto.
Observaciones		

20. R_ProductoFacturaVenta

Entidad:	R_ProductoFacturaVenta	
Descripción	Contiene los productos que están en el detalle de la factura de venta.	
Atributos:	Tipo	Descripción de los Atributos:
IDProducto	Integer	Identificador del producto
IDDocumento	Integer	Identificador de documento de factura de venta
Cantidad	Float	Cantidad del producto
CodigoUM	Varchar	Código de la Unidad de Medida.
PrecioMN	Float	Precio de venta en moneda nacional, por Unidad del producto, tomado de D_PreciosdeVenta
PrecioMLC	Float	Precio de venta en divisa, por Unidad del producto, tomado de D_PreciosdeVenta.
IDMonedaMLC	Integer	Identificador de la moneda libremente convertible
Observaciones		

En este proceso interviene la entidad D_Documento (especificado en la tabla 2 del proceso 2.3.1) debido a que es base para todo tipo de documento primario que se maneja en el almacén, en este caso los vales de salida y facturas de ventas. También intervienen las entidades ProductoFechaVencimiento y ProductoSeccion (detalladas en las tablas 12 y 13 respectivamente del proceso 2.3.1), las mismas se deben actualizar ya que se ha realizado una operación sobre el inventario de los almacenes.

En este proceso intervienen las entidades N_Seccion (tabla 43) y N_TipoProducto (tabla 41), que constituyen los nomencladores para las Secciones y para los Productos, respectivamente. Nota: Todos los nomencladores son descritos en el epígrafe 2.4 de este capítulo.

El diseño lógico que representa las necesidades de almacenamiento de información del proceso de salida de productos para insumos o ventas, se presenta en las siguientes figuras.

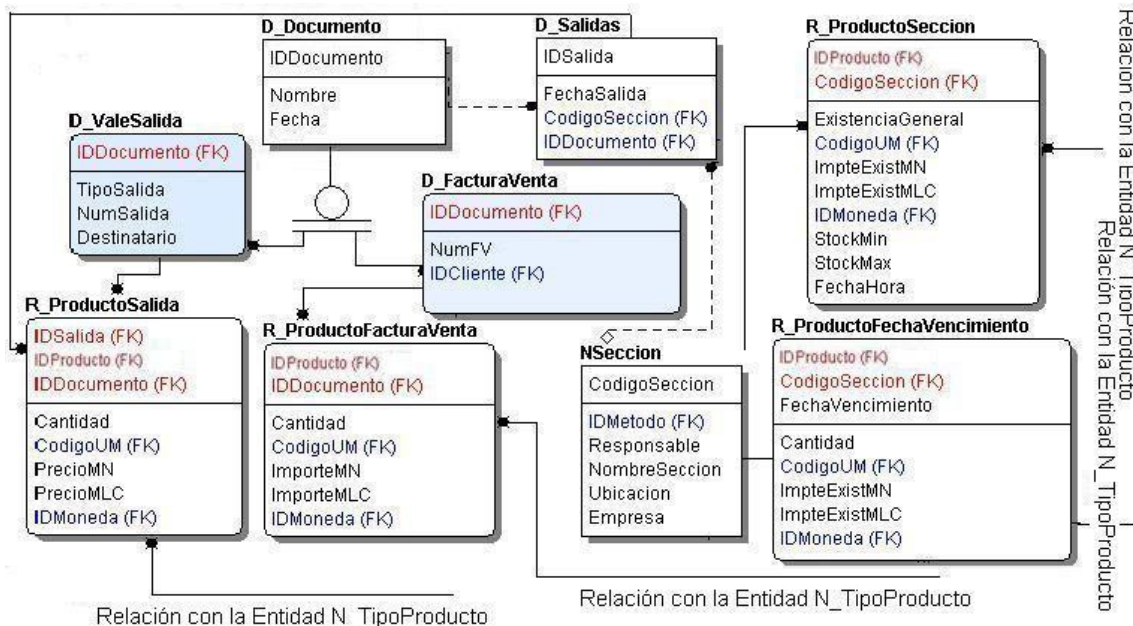


Figura 2.3: Diseño lógico para el proceso Salida de Productos por Venta a Clientes.

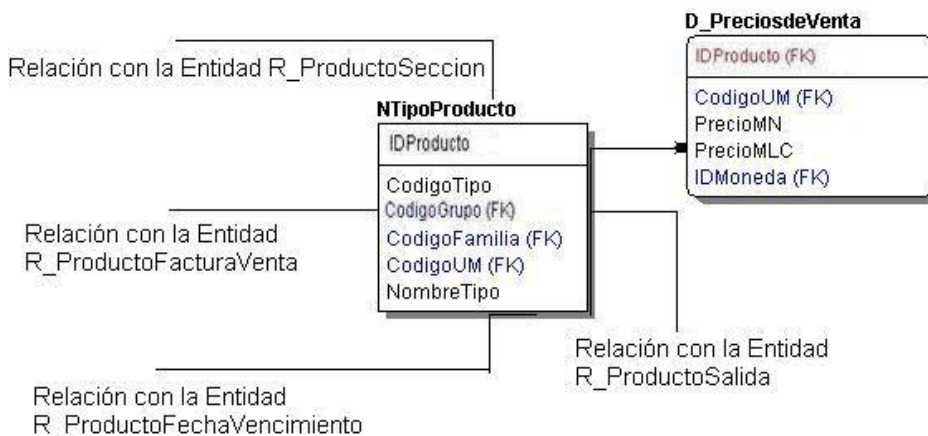


Figura 2.4: Diseño lógico para el proceso Salida de Productos por Venta a Clientes (continuación figura 2.3).

2.3.3 Solicitud de materiales desde un departamento solicitante.

El objetivo de este proceso es formalizar el pedido de productos al almacén con destino a la satisfacción de las necesidades de la unidad solicitante.

El proceso de solicitud de materiales transcurre de la siguiente manera:

1. Se elabora el documento Solicitud de materiales. (Ver anexo 4, 7, 10)
2. Se registran los productos que se solicitan.

CAPÍTULO II

Las entidades fuertes que se requieren para que la aplicación informaticice este proceso son:

21. D_ValeSolicitud

Entidad:	D_ValeSolicitud	
Descripción	Almacena el encabezado del Vale de la Solicitud.	
Atributos:	Tipo	Descripción de los Atributos:
IDDocumento	Integer	Identificador del documento de solicitud
NumVSM	Integer	Número del modelo de la Solicitud
CodigoSeccion	Varchar	Código de la sección que recibe la solicitud
Solicitante	Varchar	Departamento que realiza la solicitud
Autoriza	Varchar	Nombre y apellidos de la persona que está facultado en el Área para autorizar la solicitud de materiales.
Recibe	Varchar	Nombre y apellidos de la persona autorizada a recibir los materiales solicitados
Observaciones		

Las entidades que representan interrelaciones son:

22. R_ProductoSolicitud:

Entidad:	R_ProductoSolicitud:	
Descripción	Relaciona los productos contenidos en el detalle de la solicitud.	
Atributos:	Tipo	Descripción de los Atributos:
IDProducto	Integer	Identificador del producto
IDDocumento	Integer	Identificador de documento de solicitud
Cantidad	Float	Cantidad del producto que se solicita
CodigoUM	Varchar	Código de la unidad de medida en que se solicita el producto.
Observaciones		

En este proceso interviene la entidad D_Documento (especificado en la tabla 2 del proceso 2.3.1) debido a que es base para todo tipo de documento primario que se maneja en el almacén, en este caso es base para el modelo de solicitud de materiales.

En este proceso intervienen las entidades N_Seccion (tabla 43) y N_TipoProducto (tabla 41), que constituyen los nomencladores para las Secciones y para los Productos, respectivamente. Nota: Todos los nomencladores son descritos en el epígrafe 2.4 de este capítulo.

El diseño lógico que representa las necesidades de almacenamiento de información del proceso de solicitud de materiales se presenta en la siguiente figura.

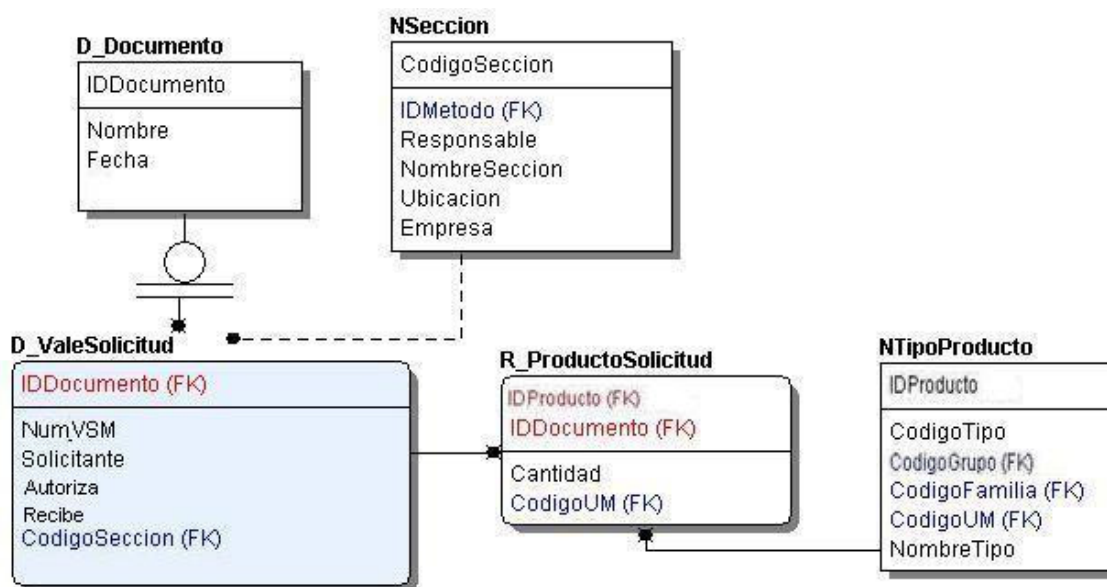


Figura 2.5: Diseño lógico para el proceso de Solicitud de Materiales.

2.3.4 Transferencia de productos entre secciones.

El proceso de transferencia de productos entre secciones transcurre de la siguiente manera:

1. Se elabora el documento "Transferencia de Productos" entre Secciones. (Ver anexo 3, 8, 10, 11, 12)
2. Se define la sección de origen y de destino
3. Se registran qué productos físicos se les dan entrada y salida a la sección que corresponde la entrada y la salida, respectivamente.

Las entidades fuertes que se requieren para que la aplicación informatices este proceso son:

23. D_ValeTransferencia

Entidad:	D_ValeTransferencia:	
Descripción	Almacena el encabezado del vale de transferencia	
Atributos:	Tipo	Descripción de los Atributos:
IDDocumento	Integer	Identificador del documento de transferencia
NumTranferencia	Integer	Número del modelo de transferencia
CodigoSeccionQueTransfiere	Varchar	Código de la sección que transfiere.
CodigoSeccionATransferir	Varchar	Código de la sección a transferir.
Despachador	Varchar	Nombre y apellidos de la persona que despacha los productos
Autorizado	Varchar	Nombre y Apellidos, de la persona que autoriza la transferencia.
Transportado	Varchar	Nombre, y Apellidos del personal que realiza la operación de transportación.
Observaciones		

Las entidades que representan interrelaciones son:

24. R_TransferenciaProducto

Entidad:	R_TransferenciaProducto	
Descripción	Relaciona los productos contenidos en el detalle de la transferencia.	
Atributos:	Tipo	Descripción de los Atributos:
IDEntrada	Integer	Identificador de la entrada de ese producto en la entidad D_Entradas
IDSalida	Integer	Identificador de salida de ese producto en la entidad D_Salidas

CAPÍTULO II

IDProducto	Integer	Identificador del producto
IDDocumento	Integer	Identificador del documento de transferencia
Cantidad	Float	Cantidad del producto que se transfiere
CodigoUM	Varchar	Código de la unidad de medida a que se refiere la cantidad del producto.
PrecioMN	Float	Precio del producto en Moneda Nacional
PrecioMLC	Float	Precio del producto en Moneda Libremente Convertible
IDMoneda	Integer	Identificador de la moneda libremente convertible.
FechaVencimiento	Date	Fecha de vencimiento del producto que se transfiere
Observaciones		

En este proceso interviene la entidad D_Documento (especificado en la tabla 2 del proceso 2.3.1) debido a que es base para todo tipo de documento primario que se maneja en el almacén, en este caso la transferencia entre secciones. También intervienen las entidades ProductoFechaVencimiento y ProductoSeccion (detalladas en las tablas 12 y 13 respectivamente del proceso 2.3.1), las mismas se deben actualizar ya que se ha realizado una operación sobre el inventario de los almacenes.

Así como también las entidades D_Salidas (tabla 18) para especificar la sección de la que se le da salida al o a los productos para otra sección, detallada en el proceso 2.3.2 y la entidad D_Entradas (tabla 4) para especificar la sección en la que se da entrada, detallada en el proceso 2.3.1.

En este proceso intervienen las entidades N_Seccion (tabla 43) y N_TipoProducto (tabla 41), que constituyen los nomencladores para las Secciones y para los Productos, respectivamente. Nota: Todos los nomencladores son descritos en el epígrafe 2.4 de este capítulo.

El diseño lógico que representa las necesidades de almacenamiento de información del proceso de transferencia de productos entre secciones se presenta en las siguientes figuras.

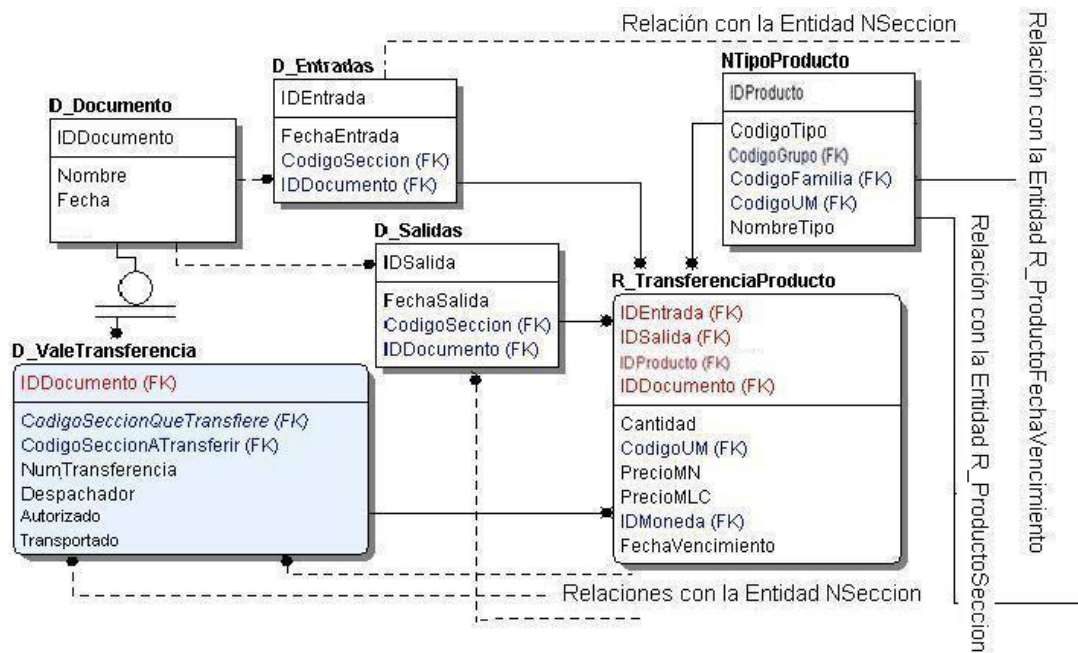


Figura 2.6: Diseño lógico para el proceso de Transferencia entre Secciones.

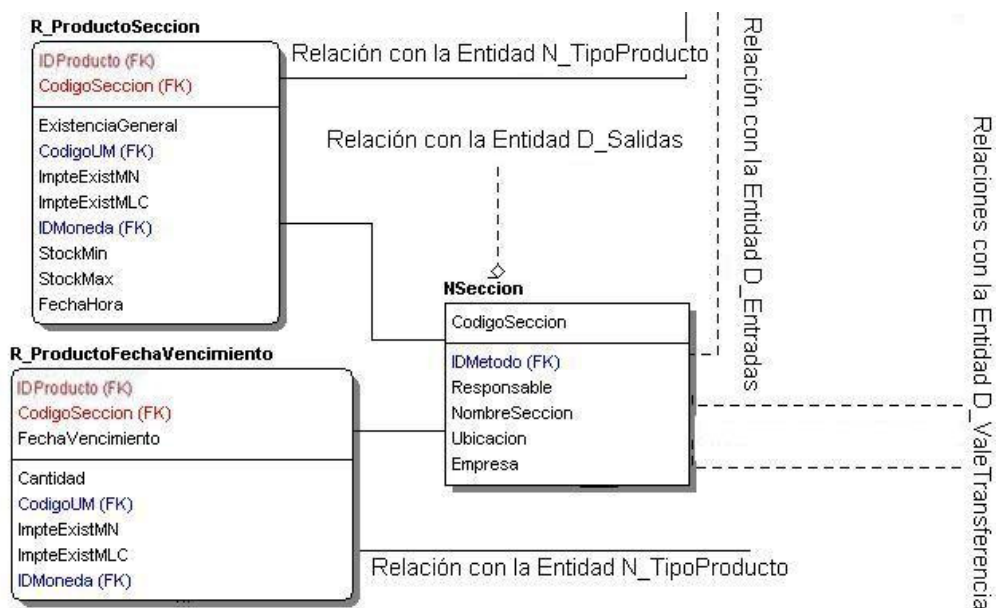


Figura 2.7: Diseño lógico para el proceso de Transferencia entre Secciones. (continuación figura 2.6).

2.3.5 Devolución de entradas de productos.

El proceso de devolución de entradas de productos transcurre de la siguiente manera:

1. Se elabora el documento primario de devolución, en donde se especifica el documento que originó la entrada, del cual se van a devolver los productos. (Ver anexo 5, 9)
2. Se registran los productos físicos que se van a devolver del documento que originó su entrada en la sección.

Las entidades fuertes que se requieren para que la aplicación informaticice este proceso son:

25. D_ValeDevolucion

Entidad:	D_ValeDevolucion:	
Descripción	Almacena los datos del encabezado del modelo Vale de Devolución.	
Atributos:	Tipo	Descripción de los Atributos:
IDDocumento	Integer	Identificador del documento Vale de devolución
IDDocumentoADevolver	Integer	Identificador del documento al que se refiere la devolución.
NumVR	Integer	Número del modelo de devolución.
NombreRecibe	Varchar	Nombre y apellidos de la persona que recibe la devolución.
NombreEmisor	Varchar	Nombre y apellidos de la persona que emite la devolución.
Observaciones		

Las entidades que representan interrelaciones son:

26. R_ProductosADevolver

Entidad:	R_ProductosADevolver:	
Descripción	Relaciona los productos que están en el detalle del vale de devolución.	
Atributos:	Tipo	Descripción de los Atributos:
IDProducto	Integer	Identificador del producto a devolver.
IDDocumento	Integer	Identificador del documento de devolución.

Cantidad	Float	Cantidad del producto que se esta devolviendo
CodigoUM	Varchar	Código de la unidad de medida en que esta expresado el producto que se devuelve
Observaciones		

En este proceso interviene la entidad D_Documento (especificado en la tabla 2 del proceso 2.3.1) debido a que es base para todo tipo de documento primario que se maneja en el almacén, en este caso el vale de devolución. También interviene las entidades ProductoFechaVencimiento y ProductoSeccion (detalladas en las tablas 12 y 13 respectivamente del proceso 2.3.1), las mismas se deben actualizar ya que se ha realizado una operación sobre el inventario de los almacenes. Así como también la entidad D_Entradas (tabla 4) para especificar la sección a la que se devuelven los productos y volver a actualizar los valores antes de la devolución, detallada en el proceso 2.3.1.

En este proceso intervienen las entidades N_Seccion (tabla 43) y N_TipoProducto (tabla 41), que constituyen los nomencladores para las Secciones y para los Productos, respectivamente. Nota: Todos los nomencladores son descritos en el epígrafe 2.4 de este capítulo.

El diseño lógico que representa las necesidades de almacenamiento de información del proceso de devolución de productos se presenta en la siguiente figura.

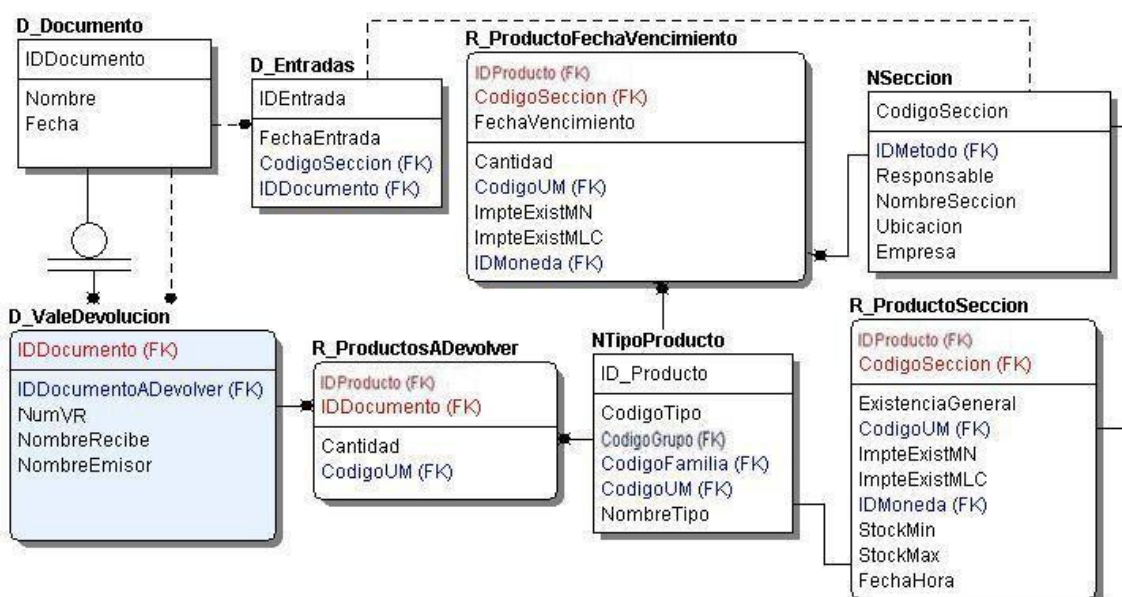


Figura 2.8: Diseño lógico para el proceso de Devoluciones de Entrada de Productos.

2.3.6 Realizar Ajustes de Inventarios en una sección.

El proceso de Ajuste de Inventarios transcurre de la siguiente manera:

1. Se elabora el documento “Ajuste de Inventarios”, el cual contiene el número del documento, la fecha y el concepto por el cual se ajusta el inventario: por conteo físico u otra causa, con la relación de productos a ajustar.
2. Se relaciona la cantidad del producto a ajustar en la sección donde se encuentre.
3. Se actualizan las existencias y su valor.

Las entidades fuertes que se requieren para que la aplicación informaticice este proceso son:

27. D_AjusteInventario

Entidad:	D_AjusteInventario	
Descripción	Almacena los datos del encabezado del modelo Ajuste de Inventarios	
Atributos:	Tipo	Descripción de los Atributos:
IDDocumento	Integer	Identificador del documento de ajuste de inventario
NumAI	Integer	Número del modelo de ajuste de inventario
IDConceptoAjuste	Integer	Identificador del concepto de ajuste de inventario
CodigoSeccion	Varchar	Código de la sección en la que se produce el ajuste de inventario.
Observaciones		

28. D_ConceptoAjuste

Entidad:	D_ConceptoAjuste	
Descripción	Almacena los datos de los diferentes conceptos de ajuste de inventarios.	
Atributos:	Tipo	Descripción de los Atributos:
IDConceptoAjuste	Integer	Identificador para el concepto de ajuste
NombreAjuste	Varchar	Nombre del concepto por el cual se ajusta el inventario, ejemplo: merma, rotura, etc.
Observaciones		

Las entidades que representan interrelaciones son:

29. R_ProductoAjusteSalida

Entidad:	R_ProductoAjusteSalida	
Descripción	Relaciona los productos que están en el modelo de ajuste de inventarios, que deben ser disminuidos de la existencia del almacén cuando el ajuste se produce por un faltante.	
Atributos:	Tipo	Descripción de los Atributos:
IDSalida	Integer	Identificador de salida de ese producto en la entidad D_Salidas
ID_Producto	Integer	Identificador del producto a ser ajustado.
IDDocumento	Integer	Identificador del documento de ajuste de inventario
Cantidad	Float	Cantidad del producto
CodigoUM	Varchar	Código de la unidad de medida
PrecioMN	Float	Importe del producto que se ajusta en Moneda Nacional
PrecioMLC	Float	Importe del producto que se ajusta en Moneda Librementemente Convertible
IDMoneda	Integer	Identificador de la moneda librementemente convertible.
Observaciones		

30. R_ProductoAjusteEntrada

Entidad:	R_ProductoAjusteEntrada	
Descripción	Relaciona los productos que están en el modelo de ajuste de inventarios, que deben ser aumentados en la existencia del almacén cuando el ajuste se produce por un sobrante.	
Atributos:	Tipo	Descripción de los Atributos:
IDEntrada	Integer	Identificador de entrada de ese producto en la entidad D_Entradas

CAPÍTULO II

ID_Producto	Integer	Identificador del producto a ser ajustado.
IDDocumento	Integer	Identificador del documento de ajuste de inventario
Cantidad	Float	Cantidad del producto
CodigoUM	Varchar	Código de la unidad de medida
PrecioMN	Float	Importe del producto que se ajusta en Moneda Nacional
PrecioMLC	Float	Importe del producto que se ajusta en Moneda Librementemente Convertible
IDMoneda	Integer	Identificador de la moneda libremente convertible.
Observaciones		

En este proceso interviene la entidad D_Documento (especificado en la tabla 2 del proceso 2.3.1) debido a que es base para todo tipo de documento primario que se maneja en el almacén, en este caso el vale de ajuste de inventarios. También interviene las entidades ProductoFechaVencimiento y ProductoSeccion (detalladas en las tablas 12 y 13 respectivamente del proceso 2.3.1), las mismas se deben actualizar ya que se ha realizado una operación sobre el inventario de los almacenes. Así como también las entidades D_Salidas (tabla 18) para especificar la sección de la que se le da salida al o a los productos por ajuste de inventarios, detallada en el proceso 2.3.2 y la entidad D_Entradas (tabla 4) para especificar la sección en la que se da entrada de productos por ajuste de inventarios, detallada en el proceso 2.3.1.

En este proceso intervienen las entidades N_Seccion (tabla 43) y N_TipoProducto (tabla 41), que constituyen los nomencladores para las Secciones y para los Productos, respectivamente. Nota: Todos los nomencladores son descritos en el epígrafe 2.4 de este capítulo.

El diseño lógico que representa las necesidades de almacenamiento de información del proceso de ajuste de inventarios en una sección se presenta en las siguientes figuras.

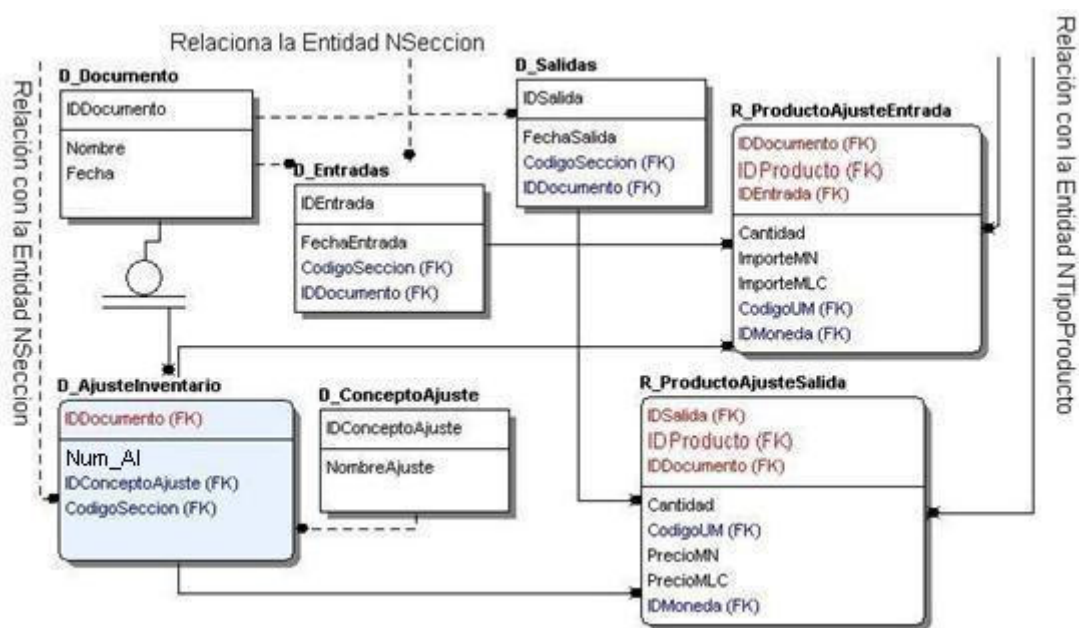


Figura 2.9: Diseño lógico para el proceso de Ajuste de Inventario de Productos.

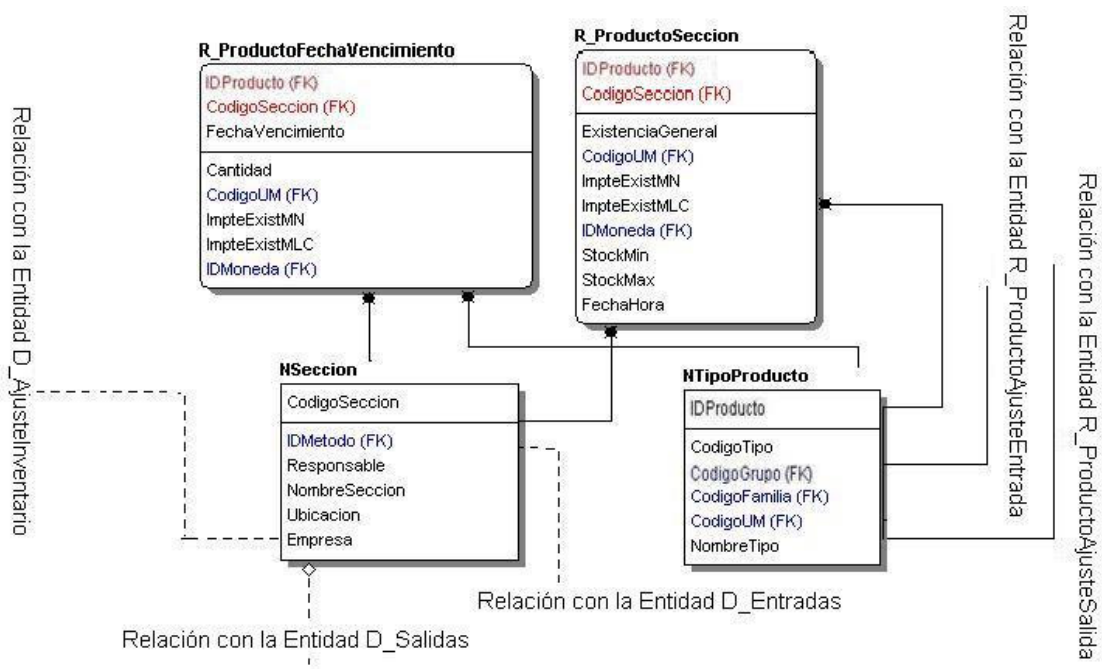


Figura 2.10: Diseño lógico para el proceso de Ajuste de Inventarios de Productos. (continuación figura 2.8).

2.3.7 Reflejar entradas de productos elaborados.

El objetivo de este proceso es formalizar la entrega de productos terminados al almacén, bien sean con destino a la comercialización o al insumo, provenientes de las áreas de producción en que se les da terminación a los mismos.

El proceso de Entradas al Almacén de Productos Elaborados transcurre de la siguiente forma:

1. El documento “Entrega de Productos Terminados” se elabora por el área productora, el cual contiene la Fecha de emisión del modelo, Nombre y código del área productora, Nombre y código del almacén receptor, el detalle de cada producto, Nombre y firma del empleado que entrega por el área productora, Nombre y firma del empleado del almacén receptor, Número de la Orden de Producción o del Lote.
2. Se seleccionan las secciones que recibirán los productos terminados.
3. Se procede a dar entrada de cada producto en o las secciones correspondientes.

Las entidades fuertes que se requieren para que la aplicación informatice este proceso son:

31. D_ProductoTerminado

Entidad:	D_ProductoTerminado	
Descripción	Almacena los datos del encabezado del modelo “Entrega de productos terminados”	
Atributos:	Tipo	Descripción de los Atributos:
IDDocumento	Integer	Identificador del documento de “Entrega de productos terminados”
NumPT	Integer	Número que posee el modelo de productos terminados
NombreRecepciona	Varchar	Nombre y apellidos de la persona que recepciona los productos terminados
NombreEntrega	Varchar	Nombre y apellidos de la persona que entrega los productos terminados
AreaProductora	Varchar	Área de la que provienen los productos terminados
Observaciones		

Las entidades que representan interrelaciones son:

32. R_ProductoTerminado

Entidad:	R_ProductoTerminado	
Descripción	Registra los productos que están en el detalle del modelo “Entrega de productos terminados”, que aumentarán las existencias del almacén.	
Atributos:	Tipo	Descripción de los Atributos:
IDEntrada	Integer	Identificador de la entrada de ese producto en la sección.
IDProducto	Integer	Identificador del producto elaborado
IDDocumento	Integer	Identificador del documento de productos terminados
Cantidad	Float	Cantidad del producto
CodigoUM	Varchar	Identificador de la unidad de medida del producto
PrecioMN	Float	Precio del producto en Moneda Nacional
PrecioMLC	Float	Precio del producto en Moneda Libremente Convertible
IDMoneda	Integer	Identificador de la moneda
Observaciones		

En este proceso interviene la entidad D_Documento (especificado en la tabla 2 del proceso 2.3.1) debido a que es base para todo tipo de documento primario que se maneja en el almacén, en este caso para el modelo “Entrega de productos terminados”. También interviene las entidades ProductoFechaVencimiento y ProductoSeccion (detalladas en las tablas 12 y 13 respectivamente del proceso 2.3.1), las mismas se deben actualizar ya que se ha realizado una operación sobre el inventario de los almacenes. Se incluye en este proceso la entidad D_Entradas (tabla 4) para especificar la sección en la que se da entrada a los productos terminados, detallada en el proceso 2.3.1.

En este proceso intervienen las entidades N_Seccion (tabla 43) y N_TipoProducto (tabla 41), que constituyen los nomencladores para las Secciones y para los Productos, respectivamente. Nota: Todos los nomencladores son descritos en el epígrafe 2.4 de este capítulo.

El diseño lógico que representa las necesidades de almacenamiento de información del proceso de Entrada de productos elaborados se presenta en la siguiente figura.

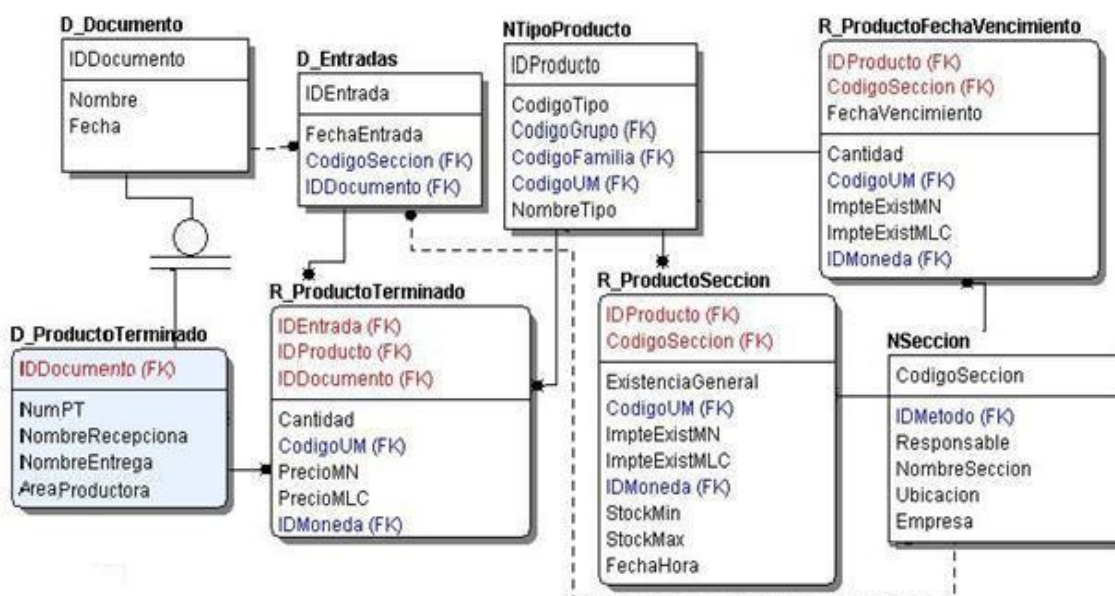


Figura 2.11: Diseño lógico para el proceso de Entrada de Productos Elaborados.

2.4 Descripción de la Aplicación.

La aplicación para el Sistema de Gestión de Inventarios y Almacenes (SIGIA), consta de los siguientes módulos:

1. Administración
2. Nomencladores
3. Control de Inventarios
4. Contabilidad
5. Manejo del Stock.
6. Reportes Generales.

La base de datos diseñada permite el desarrollo de todos los módulos excepto el de Contabilidad, el cual se ha planificado para la versión 2 del sistema.

2.4.1 Administración

El control de acceso a la aplicación, así como los privilegios con los que contara cada usuario, constituyen elementos indispensable en sistemas informáticos en especial los relacionados con el control y la gestión de almacenes e inventarios ya que a través de ellos se manejan componentes importantes de los que depende el éxito o fracaso de determinada empresa.

Los requisitos que se definen para permitir la administración del sistema son los siguientes [Liz Domenech y Cordero 2007]:

1. Administrar accesos al sistema.
 - 1.1. Permitir autenticación de los usuarios.
 - 1.2. Crear usuarios del sistema.
 - 1.3. Eliminar usuarios del sistema.
 - 1.4. Crear roles de usuarios con sus permisos y niveles de acceso.
 - 1.5. Registrar estaciones de trabajo.
2. Verificar que los usuarios solo accedan a determinadas funcionalidades según los niveles de acceso a la información previamente establecidos.

Las entidades fuertes que se requieren para que la aplicación informatices estos requisitos son:

33. D_Usuario

Entidad:	D_Usuario	
Descripción	Usuarios que pueden utilizar el “Sistema de Gestión de Inventarios y Almacenes”	
Atributos:	Tipo	Descripción de los Atributos:
Usuario	Varchar	Nombre que identifica al usuario que está utilizando el sistema.
IDRol	Integer	Identificador del rol. Es un valor auto incremental. Sirve para identificar el rol que tiene asociado el usuario.
Contrasena	Varchar	Clave con la que el sistema identificará a cada usuario.
Nombre	Varchar	Nombre de la persona usuaria.
PrimerApellido	Varchar	Primer apellido de la persona usuaria
SegundoApellido	Varchar	Segundo apellido de la persona usuaria
CarnetIdentidad	Numeric	Carnet de identidad del usuario.
CargoUsuario	Varchar	Responsabilidad laboral del usuario.
Observaciones		

34. D_Rol

Entidad:	D_Rol	
Descripción	Comportamiento y responsabilidades de un individuo que interactúa con el sistema.	
Atributos:	Tipo	Descripción de los Atributos:
IDRol	Integer	Identificador del rol
Nombre	Varchar	nombre que identifica el rol
Observaciones		

35. D_Opciones

Entidad:	D_Opciones	
Descripción	Acciones que puede ejecutar cada rol asignado a un usuario.	
Atributos:	Tipo	Descripción de los Atributos:
IDOpcion	Integer	Identificador de la opción.
Descripcion	Varchar	Descripción de cada acción
Observaciones		

36. D_Terminal

Entidad:	D_Terminal	
Descripción	Estaciones de trabajo desde las cuales es posible tener acceso al Sistema de Gestión de Inventarios y Almacenes.	
Atributos:	Tipo	Descripción de los Atributos:
IDTerminal	Integer	identificador de la terminal
Nombre	Varchar	Nombre de la estación de trabajo desde la que se ejecutará la aplicación.
Activa	Bit	Indica si la estación de trabajo está registrada o no para el trabajo con la aplicación.
EnUso	Bit	Define si la terminal está haciendo uso de la aplicación.
Observaciones		

Las entidades que representan interrelaciones son:

37. R_RolOpciones

Entidad:	R_RolOpciones	
Descripción	Contempla para cada rol las opciones que tiene asociadas.	
Atributos:	Tipo	Descripción de los Atributos:
IDRol	Integer	Identificador del rol.
IDOpcion	Integer	Identificador de la opción.
Observaciones	Tabla resultado de la relación Muchos a Muchos que se establece entre las entidades A_Rol y A_Opciones.	

38. R_OpcionOpciones

Entidad:	R_OpcionOpciones	
Descripción	Cada opción contempla un conjunto de facultades que se asocian a una misma opción.	
Atributos:	Tipo	Descripción de los Atributos:
IDOpcion	Integer	Identificador de la opción padre.
IDOpciones	Integer	Identificador de la opción hija.
Observaciones	Resultado de la relación recursiva que establece A_Opciones debido a que una opción puede tener un conjunto de opciones hija.	

El diseño lógico que representa las necesidades de almacenamiento de información del modulo de administración se presenta en la siguientes figura.

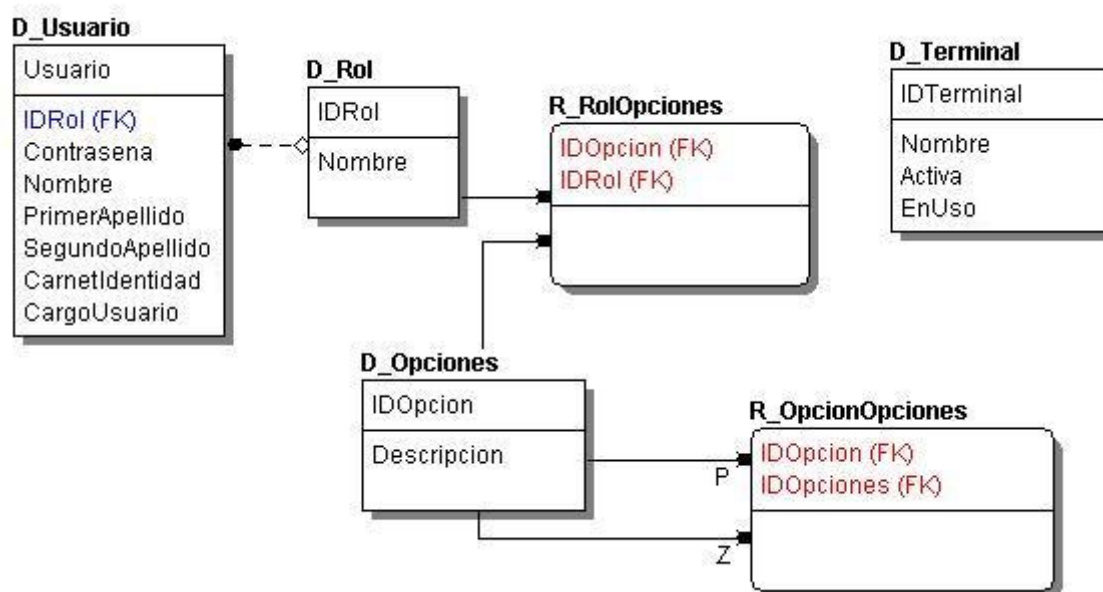


Figura 2.12: Diseño lógico para los requerimientos del Modulo de Administración.

2.4.2 Nomencladores

El módulo para el mantenimiento de los Nomencladores básicos en que se registran los datos descriptivos de cada elemento que interviene en la gestión de inventarios y almacenes.

Las entidades fuertes que se requieren para que la aplicación informatices estos requisitos son:

39. N_Grupo

Entidad:	N_Grupo	
Descripción	Conjunto de productos con características similares.	
Atributos:	Tipo	Descripción de los Atributos:
CodigoGrupo	Varchar	Código del grupo
Nombre	Varchar	Nombre
Descripcion	Varchar	Descripción del grupo
Observaciones		

40. N_Familia

Entidad:	N_Familia	
Descripción	Subdivisión de productos que pertenecen a un mismo grupo.	
Atributos:	Tipo	Descripción de los Atributos:
CodigoGrupo	Varchar	Código del grupo
CodigoFamilia	Varchar	Código de la familia
Nombre	Varchar	Nombre de la familia
Observaciones		

41. N_TipoProducto

Entidad:	N_TipoProducto	
Descripción	Almacena la información de las unidades de medida.	
Atributos:	Tipo	Descripción de los Atributos:
IDProducto	Integer	Identificador de Producto.
CodigoTipo	Varchar	Código del producto.
NombreTipo	Varchar	Nombre del producto.
CodigoGrupo	Varchar	Código del Grupo.
CodigoFamilia	Varchar	Código de la Familia.
CodigoUM	Varchar	Código de la Unidad de Medida Base.
Observaciones		

42. N_Empresa

Entidad:	N_Empresa	
Descripción	Entidad o empresa en la que se utiliza el sistema.	
Atributos:	Tipo	Descripción de los Atributos:
CodigoEmpresa	Varchar	Código de la Empresa.
Nombre	Varchar	Nombre de la empresa.
Organismo	Varchar	Organismo superior al que pertenece la entidad.
Logo	Image/Long Binary	Logo de la empresa.
Observaciones		

43. N_Seccion

Entidad:	N_Seccion	
Descripción	Sección o almacén de la entidad o empresa en la que se utiliza el sistema.	
Atributos:	Tipo	Descripción de los Atributos:
CodigoSeccion	Varchar	Define el código de la sección.
NombreSeccion	Varchar	Nombre de la sección definida.
Responsable	Varchar	Nombres y apellidos de la persona responsable de la sección.
Ubicacion	Varchar	Localización geográfica de la sección.
IDMetodo	Integer	Referencia al método que emplea la sección para realizar la valoración de sus inventarios.
Observaciones		

44. N_MetodoValoracionInventario

Entidad:	N_MetodoValoracionInventario	
Descripción	Método de Valoración de Inventario. Esto se refiere a los costos de los productos de salida del almacén o sección atendiendo a diferentes criterios, <i>FIFO, LIFO, PONDERADO</i> .	
Atributos:	Tipo	Descripción de los Atributos:
IDMetodo	Integer	Identificador de metodo
nombre	Varchar	Nombre del método de valoración de inventario (LIFO, FIFO, PONDERADO)
Observaciones		

45. N_SeccionFamilia

Entidad:	N_SeccionFamilia	
Descripción	Vincula determinadas familias de productos a una sección	
Atributos:	Tipo	Descripción de los Atributos:
CodigoFamilia	Varchar	Código de la Familia.
CodigoGrupo	Varchar	Código del grupo.
CodigoSeccion	Varchar	Código de la sección.

Operacion	Varchar	Control de movimiento de esa familia en esa sección, ejemplo entrada, salida o ambos.
Observaciones		

46. N_Moneda

Entidad:	N_Moneda	
Descripción	almacena las monedas que son utilizadas por el sistema	
Atributos:	Tipo	Descripción de los Atributos:
IDMoneda	Integer	Identificador de moneda.
NombreMoneda	Varchar	Nombre de la moneda (CUC, EURO, etc.)
TasaCambio	Float	Tasa de cambio de la moneda
Observaciones		

47. N_UnidadMedida

Entidad:	N_UnidadMedida	
Descripción	Almacena la información de las unidades de medida.	
Atributos:	Tipo	Descripción de los Atributos:
CodigoUM	Varchar	Código de la Unidad de Medida
NombreUM	Varchar	Nombre de la Unidad de Medida
TipoDato	Varchar	Define las métricas para la unidad de medida, estas métricas son (ENTERO, REAL, etc.) Ejemplo: la unidad de medida caja tendrá como métrica ENTERO, pero la unidad de medida Kg. tendrá métrica REAL.
Observaciones		

48. N_EquivalenciaUM

Entidad:	N_EquivalenciaUM	
Descripción	Almacena la información que permita establecer las equivalencias de unidades de medida.	
Atributos:	Tipo	Descripción de los Atributos:
CodigoUM	Varchar	Código unidad Medida Base

CodigoUM2	Varchar	Código de la unidad de medida al que se quiere establecer la equivalencia.
Operacion	Char	Operación matemática que se debe establecer para lograr la equivalencia (Multiplicar(*), Dividir(/))
Cantidad	Integer	Cifra que se debe multiplicar o dividir para lograr la conversión a la unidad de medida deseada.
Observaciones		

El diseño lógico que representa las necesidades de almacenamiento de información del módulo de nomencladores se presenta en la siguiente figura.

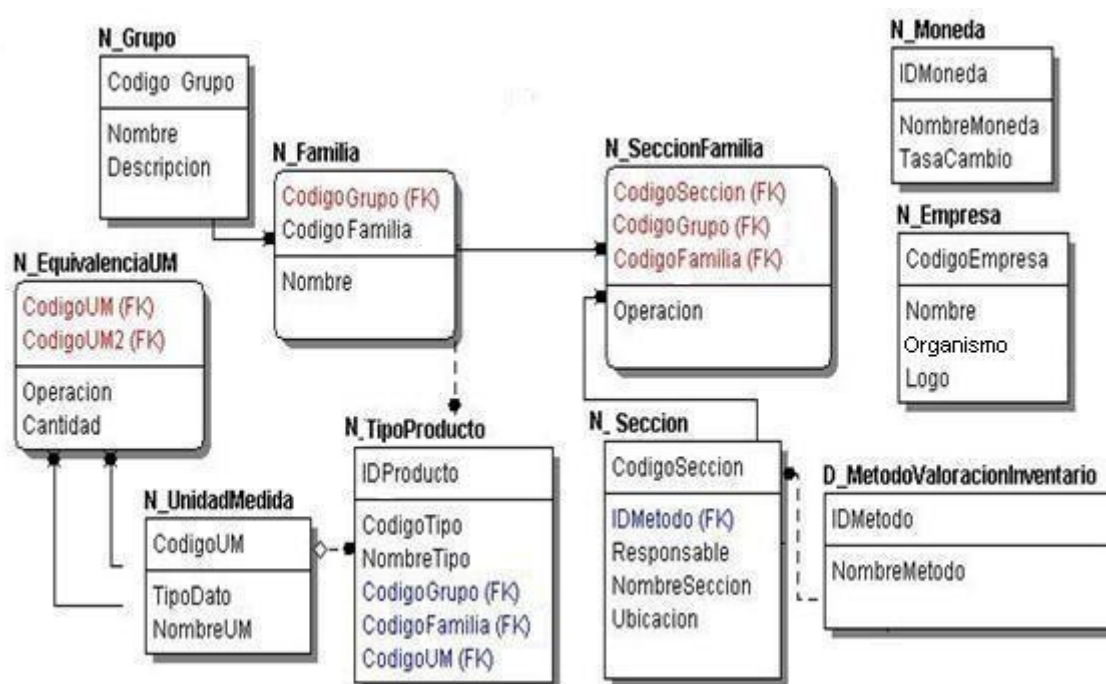


Figura 2.13: Diseño lógico para los Nomencladores Básicos.

2.5 Caracterización y análisis crítico de la propuesta de diseño lógico.

El modelo lógico obtenido posee las siguientes propiedades:

- Integridad
- Normalización de la Base de Datos
- Análisis de redundancia de información
- Análisis de la seguridad de la base de datos
- Trazabilidad de las acciones.

La integridad de la información almacenada deberá garantizarse en la aplicación utilizando los mecanismos que posea el gestor de bases de datos que se emplee para la implementación del sistema. El modelo físico que se obtiene a partir de este modelo lógico contiene reglas de integridad referencial al contemplarse en cada tabla tipo interrelación el conjunto de llaves foráneas a otras tablas necesarias. No se ha nombrado llaves extranjeras con otros nombres, lo cual ayuda a que el programador pueda hacer coincidir en valor campos que deben ser iguales tanto en las operaciones de inserción como de eliminación o actualización.

La regla de integridad del dominio se garantiza a través de la definición en el modelo lógico de las llaves primarias, luego en el modelo físico esta regla se garantiza al comprobar que todo registro no posea nulo ningún campo de la llave.

El modelo físico que se genera a partir de este modelo lógico se encuentra en Tercera Forma Normal [Date2005] ya que se ha utilizado un diseño orientado a objetos a partir del descubrimiento de las entidades en el propio entorno del dominio del problema y utilizando los conceptos de la modelación entidad relación extendida. La Tercera Forma Normal evita:

- La redundancia de los datos: repetición de datos en un sistema.
- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

El proceso de normalización consta de varias fases denominadas formas normales, estas formas se detallan a continuación.

Es de vital importancia antes de proceder a la normalización de la tabla definir cual es su clave, esta clave deberá contener un valor único para cada registro y podrá estar formado por un único campo o por un grupo de campos. Elemento que es contemplado en la propuesta de diseño al tener cada entidad definida su llave primaria.

Una tabla se encuentra en primera forma normal (1FN) si y solo si cada uno de los campos contiene un único valor para un registro determinado. Cada una de las

CAPÍTULO II

entidades que conforman el diseño lógico solo admite un único valor por cada campo de registro realizado. Ejemplo:

La entidad N_Moneda, cuyos atributos son el identificador de moneda, Nombre de la moneda y la Tasa de cambio, para cada campo del registro contienen un único dato, lo que se puede apreciar en la siguiente tabla.

IDMoneda	NombreMoneda	TasaCambio
03	CUC	1.0000
04	CAD	0.8553
05	EUR	1.2455

El mismo análisis seguido para esta entidad se le realizó a cada una de las entidades que conforman la propuesta de diseño lógico y los resultados obtenidos indican que esta está en 1FN, lo que permite asegurar que una vez transformado el diseño lógico propuesto en una base de datos para un gestor determinado cada una de sus tablas cumpla con la 1FN.

Una vez alcanzado el diseño la 1FN, se debe continuar con el proceso de normalización y pasar a la segunda forma normal (2FN).

La 2FN compara todos y cada uno de los campos de la tabla con la clave definida. Si todos los campos dependen directamente de la clave se dice que la tabla está en 2FN. Elemento que se tiene en cuenta en el diseño pues cada atributo definido para una entidad depende directamente de la clave definida, ejemplo:

La entidad N_Empresa, tomando como punto de partida que la llave está formada por el campo `CodigoEmpresa`.

- El campo nombre depende funcionalmente de la llave definida, para la empresa con código 03 el nombre que le corresponde es Softel y no otro.
- El campo Organismo depende funcionalmente de la llave definida, para la empresa con código 03 el organismo es el MIC.
- El campo Logo depende funcionalmente de la llave definida, para la empresa 03 tiene un único logo definido.

El mismo análisis seguido para esta entidad se le realizó a cada una de las entidades que conforman la propuesta de diseño lógico y los resultados obtenidos indican que

esta en 2FN, lo que permite asegurar que una vez transformado el diseño lógico propuesto en una base de datos para un gestor determinado cada una de sus tablas cumpla con la 2FN.

Una vez alcanzado el diseño la 1FN y 2FN se debe continuar con el proceso de normalización y pasar a la tercera forma normal (3FN).

Una tabla está en 3FN si y solo si los campos de la tabla dependen únicamente de la clave, dicho en otras palabras los campos de las tablas no dependen unos de otros. El diseño lógico contempla que cada atributo no posea ninguna dependencia de otro atributo que no sea la llave definida para dicha entidad. Ejemplo:

La entidad N_EquivalenciaUM, para la cual se definieron los siguientes atributos:

- CódigoUM, como llave primaria
- CódigoUM2, como llave primaria
- Operación, este atributo solo depende de las llaves primarias, el resto de los atributos le es indiferente.
- Cantidad, este atributo solo depende de las llaves primarias, el resto de los atributos le es indiferente.

El mismo análisis seguido para esta entidad se le realizó a cada una de las entidades que conforman la propuesta de diseño lógico y los resultados obtenidos indican que esta en 3FN, lo que permite asegurar que una vez transformado el diseño lógico propuesto en una base de datos para un gestor determinado cada una de sus tablas cumpla con la 3FN.

Una vez alcanzado el diseño la 1FN, 2FN y 3FN puede considerarse que una vez generado la Base de datos para un gestor de base de datos determinado la misma se encuentre normalizada.

La seguridad para el sistema se ha contemplado en el modelo lógico al concebir tablas que almacenan los usuarios posibles de la aplicación y sus opciones permitidas, el resto de los mecanismos de seguridad son proporcionados por el gestor a los programadores.

Como principales bondades del diseño lógico de los datos podemos citar:

- El diseño permite la adición de nuevos tipos de documentos de entrada y salida sin tener que cambiar la estructura de las tablas para reflejar las operaciones de entrada y salida de productos a una sección. Esto se ha logrado al existir la tabla base DOCUMENTO que es una generalización de tipos de documentos.
- Se han incluido dos monedas en la valoración de los inventarios.
- Se han considerado equivalencias generales entre unidades de medida.
- El carácter genérico del modelo de datos permite ser adaptado a diferentes tipos de empresa, ya sean éstas de producción o de servicios.

El almacenamiento de la información de forma centralizada es la propuesta de modelo físico para la base de datos del Sistema de Gestión de Inventarios y Almacenes de manera que se permita la integración de todos los módulos de la aplicación por medio de la consulta y actualización a una misma base de datos relacional, lo que significa que la base de datos es compartida por todos los módulos que integran el Sistema de Gestión de Inventarios y Almacenes (SIGIA).

2.6 Conclusiones.

En este capítulo se han explicado los procesos de negocio de las operaciones que una Empresa realiza relacionadas con su gestión de inventarios.

Se describen las entidades e interrelaciones que se identificaron en el entorno del dominio del problema, así como las entidades e interrelaciones que se demandan por los módulos de administración y de nomencladores de SIGIA. Las entidades que corresponden al soporte de la automatización de los procesos de negocio constituyen la estructura estática de información para el desarrollo del resto de los módulos de la aplicación.

Cada diagrama constituye un subconjunto del Modelo Lógico de los Datos.

El análisis crítico de la propuesta de diseño lógico que se describió en el capítulo, constituye el último tema tratado en el mismo.

CONCLUSIONES

- Se diseñó el modelo lógico de la capa de datos, que permite a la aplicación SIGIA almacenar los datos que se utilizan o generan en la Gestión de Inventarios y Almacenes.
- Identificar las necesidades de los usuarios y los procesos que se realizan manualmente son imprescindibles para obtener las entidades y las relaciones entre ellas, lo que conformaría como un todo el diseño de una base de datos.
- Las metodologías de diseño de bases de datos no pueden ser suprimidas cuando se enfrenta un diseño para una base de datos de una aplicación profesional.

RECOMENDACIONES

- Validación del modelo de datos en las empresas que fueron objeto de estudio.
- Refinamiento constante del modelo lógico propuesto, durante los ciclos de desarrollo restantes del proyecto de Sistema de Gestión de Inventarios y Almacenes (SIGIA).
- Tomar el modelo lógico como base para permitir que el SIGIA en un futuro forme parte de una aplicación de planificación de recursos empresariales, (ERP, de sus siglas en inglés: Enterprise Resource Planning) para lograr abarcar la gestión de todas las áreas de la organización, así como su integración a un GIS (sistema de información geográfica) de manera que se pueda visualizar en mapas digitales la ubicación geográfica de los productos en los almacenes.

REFERENCIAS BIBLIOGRÁFICAS

[Chen 1976] Chen, P.P. *“The entity-relationship model —toward a unified view of data”*. ACM Transactions on Database Systems. 1976. pp 9-36.

[Codd 1970] Codd, E.F. *“A Relational Model for Large Shared Data Banks”* Communications of the ACM, 1970. pp 377-387, También reimpresso en [Stonebraker & Hellerstein 1998].

[Codd 1990] Codd, E.F. *“The Relational Model for Database Management”*. Version 2” Reading, Massachusetts, Addison Wesley, 1990.

[Copeland & Maier 1984] Copeland, D.; Maier *“Making SMALLTALK a database system”* ACM SIGMOD Conf. on the Management of Data, 1984.

[C. Batini, S. Ceri, S.B. Navathe 1994] C. Batini, S. Ceri, S.B. Navathe *“Diseño Conceptual de Bases de Datos. Un enfoque de entidades-interrelaciones.”* Addison-Wesley / Díaz de Santos, 1994.

[Date 1981] Date, C.J. *“An Introduction to Database Systems”*, Addison-Wesley, 1981.

[Date 1999] Date, C.J. *“An Introduction to Database Systems”*, 7th Edition (25th Anniversary Edition), Addison-Wesley, 1999.

[Decups ,2002] Decups. *“Base de datos”*. Tema 1: Introducción. Curso 2001-2002.

[Disponible en:
<http://www.upseros.com/fotocopiadora/ficheros/Bases%20de%20Datos/bbddtema1.pdf>
]

[Delobel & Adiba 1985] Delobel, C.; Adiba, M. *“Relational Database Systems”* North-Holland 1985.

[De Miguel, A. y Piattini, M. 1999]. De Miguel, A. y Piattini, M *“Fundamentos y modelos de bases de datos.”* 2^{da} edición. Editorial Ra-Ma, Madrid, 1999.

[Everest 1986] Everest, G.C. *“Database Management. Objectives, System Functions, and Administration”* McGraw-Hill Book Company, 1986.

REFERENCIAS BIBLIOGRÁFICAS

[Elmasri & Navathe 2000] Elmasri, R.; Navathe, S.B. *“Fundamentals of Database Systems”* 3rd Edition, Addison-Wesley, 2000.

[Gardarin 1988] Gardarin, G. *“Bases de Données. Les systèmes et leurs langages”* 5eme Édition, Eyrolles 1988.

[Gardarin & Valduriez 1987] Gardarin, G.; Valduriez P. *“Bases de Données Relationnelles. Analyse et Comparaison de Systèmes”* 2eme Édition, Eyrolles 1987.

[Hernández Orallo, 2002] José Hernández Orallo *“La disciplina de los sistemas de bases de datos. Historia, Situación actual y perspectiva”*. Universidad Politécnica de Valencia, mayo 2002. [Disponible en: <http://www.dsic.upv.es/~jorallo/docent/BDA/DisciplinaBD.pdf>]

[Hernández y Coello 2002] Rolando Alfredo Hernandez Leon y Sayda Coello González *“El paradigma cuantitativo de la investigación científica”* Editorial Universitaria (EDUNIV). Noviembre 2002.

[Jacobson et al 2000] Ivar Jacobson, Grady Booch, James Rumbaugh *“El Proceso Unificado de Desarrollo de Software”*. Addison Wesley, 2000.

[Liz Domenech y Cordero 2007] Grethel Liz Domenech y Mayuli Cordero Valido *“Análisis de Sistema de Gestión de Inventarios”*. Universidad de las Ciencias Informáticas. La Habana. 2007.

[Marqués Andrés, 2001] María Mercedes Marqués Andrés *“Historia de los sistemas de bases de datos”*. Universidad Jaume I en Castellón. 2001-02-12. [Disponible en: <http://www3.uji.es/~mmarques/f47/apun/node6.html>]

[MFP 2000]. *“Finanzas al Día. Control Interno”*. Disposiciones Vigentes del Ministerio de Finanzas y Precios, Cuba, 2000.

[Piattini Velthuis 2000] Mario Piattini Velthuis *“Líneas de evolución de las bases de datos”*. 2000.

[Rodríguez, 2004] Rubén Rodríguez *“Bases De Datos (La Historia)”*. [Disponible en: <http://www.sonria.com/notas/nota.php?id=22&a=Adim.>]

[Ro y Norick 2003] Ro y Norick *“Diseño de Bases de Datos Relacionales”* .2003[Disponible en: <http://usuarios.lycos.es/cursosgbd>]

REFERENCIAS BIBLIOGRÁFICAS

[Stonebraker & Hellerstein 1998] Stonebraker, M. and Hellerstein, J. (ed.) *“Readings in Database Systems”*, Third Edition, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, San Francisco, 1998. [Disponible en: <http://redbook.cs.berkeley.edu/>]

[Sánchez 2004] Jorge Sánchez *“Principios sobre Bases de Datos Relacionales”*, 2004.

[Ullman 1980] Ullman, J.D. *“Principles of Database Systems”*, Computer Science Press, 1980.