

Universidad de las Ciencias Informáticas
Facultad 8



Título: Diseño del modelo de la Base de Datos de
Entidades y Cargos.

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

Autor: Damián Llópiz Oliva

Tutor: Ing. Dobranchy Jiménez Quesada

Julio, 2008

“Año 50 de la Revolución”

*Intenta no volverte un hombre de éxito, sino volverte un hombre
de valor.*

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Damián Llópiz Oliva

Ing. Dobranchy Jiménez Quesada

AGRADECIMIENTOS

A mí mamá y mi papá a mi hermano por el apoyo en todos estos años de esfuerzo y sacrificio.

A toda mi familia.

A mi tutor por su ayuda.

Y a Fidel y la revolución.

DEDICATORIA

*A mi mamá por todo el amor, su dedicación, su incansable lucha
en su afán porque terminares mis estudios.*

*A mi papá por haberme guiado por el camino correcto de la vida,
por todos los valores transmitidos y más que todo por ser para mí,
un héroe.*

*A mi hermano por su preocupación, por todo su cariño y gracias
por ser para mí una guía de ejemplo.*

*A mi abuela Aracelia por su preocupación diaria, por todo lo que
ha hecho por mí para lograr todo esto y por llevarme de la mano
cada día a la escuela.*

*A la memoria de mi abuelo Pedro por ser para mí como otro
padre.*

A mis tías por toda su ayuda, su preocupación y su afecto.

A toda mi familia por apoyarme.

A todos, gracias, Sin ellas nada de esto podría haber sido posible.

RESUMEN

Con el desarrollo de este trabajo, se persigue el objetivo de diseñar un modelo de Base de Datos que facilite el trabajo a todos los niveles con las entidades y cargos en las FAR.

Para estructurar y diseñar un modelo de Base de Datos de Entidades y Cargos que este de acuerdo y cumpla con lo establecido se utilizan diferentes herramientas de desarrollo: Visual Paradigm para el modelado de la misma, PostgreSQL 8.2 como Sistema Gestor de Base de Datos, EMS SQL Manager for PostgreSQL para la administración y desarrollo de la base de Datos y como metodología del desarrollo del software, RUP.

Con el desarrollo de la BD, se espera lograr el incremento de la capacidad organizativa de la información en lo que respecta a Entidades y Cargos en las FAR.

PALABRAS CLAVES

BD: Base de Datos

SGBD: Sistema Gestor Base de Datos.

ÍNDICE DE CONTENIDOS

AGRADECIMIENTOS	II
DEDICATORIA.....	III
RESUMEN.....	IV
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1 INTRODUCCIÓN.....	4
1.2 INTRODUCCIÓN A LAS BASES DE DATOS.....	4
1.2.1 DEFINICIÓN DE BASE DE DATOS	5
1.3 MODELOS DE BASES DE DATOS.....	6
1.3.1 BASES DE DATOS JERÁRQUICAS.....	6
1.3.2 BASES DE DATOS DE RED	7
1.3.3 BASES DE DATOS RELACIONALES.....	7
1.3.4 BASES DE DATOS ORIENTADAS A OBJETOS.....	7
1.4 TENDENCIAS ACTUALES Y FUTURAS	8
1.5 ANÁLISIS Y DESCRIPCIÓN DE LAS DIFERENTES HERRAMIENTAS A USAR	9
1.5.1 SISTEMAS GESTORES DE BASES DE DATOS (SGBD)	9
1.5.1.1 ORACLE	9
1.5.1.2 MYSQL	10
1.5.1.3 POSTGRESQL.....	11
1.5.2 METODOLOGÍAS DE DESARROLLO DE SOFTWARE	14
1.5.2.1 PROCESO UNIFICADO DE RATIONAL (RUP)	15
1.5.2.2 PROGRAMACIÓN EXTREMA (XP).....	15
1.5.3 HERRAMIENTAS PARA EL LENGUAJE DE MODELADO VISUAL.....	16
1.5.3.1 RATIONAL ROSE ENTERPRISE	16
1.5.3.2 VISUAL PARADIGM.....	17
1.6 HERRAMIENTA PARA LA ADMINISTRACIÓN Y DESARROLLO DE POSTGRESQL DATABASE SERVER..	19
1.6.1 EMS POSTGRESQL MANAGER	19
1.7 CONCLUSIONES.....	19
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.....	20
2.1 INTRODUCCIÓN.....	20
2.2 SELECCIÓN Y ARGUMENTACIÓN DE LOS REQUISITOS NO FUNCIONALES DEL SISTEMA PROPUESTO.	20
2.3 DESCRIPCIÓN DE LA REPRESENTACIÓN GRÁFICA DE LA ESTRUCTURA Y LA COMPOSICIÓN	21
2.4 DESCRIPCIÓN DE LA ARQUITECTURA Y FUNDAMENTACIÓN	32
2.5 DIAGRAMA DE CLASES PERSISTENTES.....	34
2.5.1 REPRESENTACIÓN DE LAS CLASES PERSISTENTES	35

2.6	DIAGRAMA ENTIDAD RELACIÓN DE LA BD.....	39
2.6.1	DESCRIPCIÓN DE LAS TABLAS.....	45
2.7	CONCLUSIONES.....	48
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA		49
3.1	INTRODUCCIÓN.....	49
3.2	VALIDACIÓN TEÓRICA DEL DISEÑO	49
3.2.1	INTEGRIDAD	49
3.2.2	NORMALIZACIÓN DE LA BASE DE DATOS.....	50
3.2.3	ANÁLISIS DE REDUNDANCIA DE INFORMACIÓN	52
3.2.4	ANÁLISIS DE LA SEGURIDAD DE LA BASE DE DATOS.....	52
3.3	VALIDACIÓN FUNCIONAL	53
3.4	CONCLUSIONES.....	54
CONCLUSIONES.....		55
RECOMENDACIONES.....		56
BIBLIOGRAFÍA.....		57
CITAS BIBLIOGRÁFICAS		58
ANEXOS		59
1	ANEXO 1. DESCRIPCIÓN DE LAS CLASES DEL DIAGRAMA DE CLASES PERSISTENTES.	59
2	ANEXO 2. DESCRIPCIÓN DE LAS TABLAS DEL DIAGRAMA ENTIDAD RELACIÓN.	66
3	ANEXO 3. REQUISITOS NO FUNCIONALES.....	77

ÍNDICE DE FIGURAS

FIGURA 1	6
FIGURA 2 ESQUEMA LÓGICO DE LA ESTRUCTURA.....	21
FIGURA 3 DIVISIÓN DE LA ESTRUCTURA	22
FIGURA 4 DISEÑO FÍSICO DE LA ESTRUCTURA Y COMPOSICIÓN.....	24
FIGURA 5 PRINCIPALES ENTIDADES DE LA ESTRUCTURA.	25
FIGURA 6 ESTRUCTURA DE ÁRBOL PARA LOS ÓRGANOS Y AGRUPACIONES.....	26
FIGURA 7 EJEMPLO DE UNA ESTRUCTURA DE ÁRBOL	27
FIGURA 8 ESTRUCTURA DE DATOS. ÓRGANOS Y AGRUPACIONES.	28
FIGURA 9 ENTIDAD "DAT_ENTIDAD".....	29
FIGURA 10 ESTRUCTURA DE DATOS. ÓRGANOS Y AGRUPACIONES.	30
FIGURA 11 COMPOSICIÓN DE LAS ENTIDADES.	31
FIGURA 12 DIAGRAMA DE CLASES PERSISTENTES.....	35
FIGURA 13 DIAGRAMA ENTIDAD RELACIÓN.....	39
FIGURA 14 SUB MODELO 1.....	40
FIGURA 15 SUB MODELO 2.....	41
FIGURA 16 SUB MODELO 3.....	42
FIGURA 16 SUB MODELO 4.....	43
FIGURA 18 SUB MODELO 5.....	44

ÍNDICE DE TABLAS

TABLA 1 AGRUPACIONEST.....	36
TABLA 2 AGRUPACIONESTOP.....	36
TABLA 3 CARGO	36
TABLA 4 ENTIDAD.....	37
TABLA 5 ORGANOEST.....	37
TABLA 6 ORGANOSTOP.....	37
TABLA 7 ESTRUCTURA	38
TABLA 8 ESTRUCTURAOP.....	38
TABLA 9 DAT_AGRUPACIONEST.....	45
TABLA 10 DAT_AGRUPACIONESTOP.....	45
TABLA 11 DAT_CARGO	45
TABLA 12 DAT_ESTRUCTURA	46
TABLA 13 DAT_ENTIDAD.....	47
TABLA 14 DAT_ESTRUCTURAOP	48
TABLA 15 CARGOCIVIL.....	59
TABLA 16 CARGOMILITAR	59
TABLA 17 MODULOS	59
TABLA 18 TECNICA.....	59
TABLA 19 NOM_AGRUPACION	60
TABLA 20 AMTOMAQ	60
TABLA 21 ARMTOSITUAC.....	60
TABLA 22 ARMTOTIPO	60
TABLA 23 GRADOMILITAR	61
TABLA 24 GTECNICO.....	61
TABLA 25 CARGOCIVIL.....	61
TABLA 26 CARGOMILITAR	62
TABLA 27 CATEGORIACIVIL.....	62
TABLA 28 ESPECIALIDAD.....	62
TABLA 29 GINSTRUCTOR	62
TABLA 30 GPROFECIONAL	63
TABLA 31 MODCARGO	63
TABLA 32 GSUBCATEGORIA.....	63
TABLA 33 MODULO.....	63
TABLA 34 NIVELESTR.....	63
TABLA 35 PREFIJO	64
TABLA 36 NIVELPREPARACION	64
TABLA 37 ORGANO.....	64
TABLA 38 PREPMILITAR.....	64
TABLA 39 TIPOCIFRA.....	65
TABLA 40 TECNICA.....	65
TABLA 41 CATEGORIACUP	65
TABLA 42 DAT_CARGOCIVIL.....	66
TABLA 43 DAT_CARGOMTAR	66
TABLA 44 DAT_MODCARGO	67
TABLA 45 DAT_MODULOS	67
TABLA 46 DAT_ORGANOEST.....	67

TABLA 47 DAT_ORGANOESTOP.....	68
TABLA 48 DAT_TECNICA.....	68
TABLA 49 NOM_AGRUPACION.....	68
TABLA 50 NOM_AMTOMAQ.....	69
TABLA 51 NOM_ARMTOSITUAC.....	69
TABLA 52 NOM_ARMTOTIPO.....	69
TABLA 53 NOM_CARGOCIVIL.....	70
TABLA 54 NOM_CARGOMILITAR.....	70
TABLA 55 NOM_CATEGOCUP.....	70
TABLA 56 NOM_CATEGCIVIL.....	71
TABLA 57 NOM_ESPECIALIDAD.....	71
TABLA 58 NOM_GINSTRUCTOR.....	72
TABLA 59 NOM_GPROFECIONAL.....	72
TABLA 60 NOM_GTECNICO.....	72
TABLA 61 NOM_GRADOMILIT.....	73
TABLA 62 NOM_GSUBCATEG.....	73
TABLA 63 NOM_MODULO.....	74
TABLA 64 NOM_NIVELESTR.....	74
TABLA 65 NOM_NIVELPREP.....	74
TABLA 66 NOM_ORGANO.....	75
TABLA 67 NOM_PREFIJO.....	75
TABLA 68 NOM_PREPMILITAR.....	75
TABLA 69 NOM_TECNICA.....	76
TABLA 70 NOM_TIPOCIFRA.....	76

INTRODUCCIÓN

Desde hace un tiempo atrás las FAR (Fuerzas Armadas Revolucionarias) ha invertido una cantidad considerable de recursos para el desarrollo de la informática en esta organización. Esto ha desarrollado en gran medida las redes de datos del país y a la vez ha surgido un mayor flujo de información.

Históricamente, cuando se desarrollaban los sistemas o aplicaciones cada desarrollador definía sus propias estructuras, esto generalmente no creaba grandes dificultades ya que la interacción entre sistemas era poca o en algunos de los casos, nula y casi siempre el proceso de desarrollo corría a cargo de una sola persona es decir la información se encontraba esparcida.

En las FAR, así como en nuestro país, no existe un modelo general, que permita el trabajo con las estructuras organizativas de los cargos, los organismos y las entidades, que facilite con esto el trabajo a todos los niveles, garantizando algún nivel de compartimentación, y disponibilidad además que garantice que las estructuras sean creadas por los órganos facultados para este fin, lo cual garantiza la uniformidad y la veracidad de la información.

Gracias al desarrollo vertiginoso de la tecnología y las grandes posibilidades que brinda la revolución, en las FAR y en el país, los sistemas que se comienzan a desarrollar, tienden a la integración, por lo que se exige que los datos que obviamente van a ser comunes a muchos de los sistemas sean registrados y controlados mediante una sola aplicación, que garantice el satisfactorio uso de estos por las demás aplicaciones.

Teniendo en cuanto lo anteriormente planteado nos planteamos el siguiente **problema**: ¿Cómo resolver el problema de integridad y compatibilidad en los sistemas informáticos, en lo que respecta a la estructura y composición?

A raíz de esto se identificó como **problema a resolver** la creación de un modelo de datos informático que facilite el trabajo con la estructura y composición de cualquier organismo del país.

El **objeto de estudio** consiste en un estudio de la estructura y de dichas estructuras en las FAR y nuestro país, del cual se deriva que su **campo de acción** sea el proceso de modelado de un modelo de datos informático para gestionar las entidades y cargos del país y las FAR.

La **idea a defender** es que si se hace un modelo de datos informático para el trabajo con las estructuras y composiciones del país, entonces se logrará con esto un trabajo más eficiente, rápido y fácil para las entidades necesitadas de dicho modelo.

El **objetivo general** de este trabajo es la creación un modelo de datos informático que permita la gestión de las estructuras de los organismos e instituciones del país.

Por lo tanto se identifica como **objetivo específico**:

1. El diseño de una Base de Datos de Entidades y Cargos.

Para poder cumplir con los objetivos planteados y darle solución al problema se definen la siguientes **Tareas**:

1. Un estudio de los modelos de estructuras existentes ya en las FAR y en nuestro país.
2. Evaluar el contenido de la información obtenida sobre la forma en que se usan las estructuras en los sistemas actuales, con el objetivo de buscar puntos coincidentes.
3. A partir del estudio crear un modelo genérico, que soporte todas las formas de estructuras encontradas.
4. Realizar el análisis para el modelo propuesto.
5. Realizar el diseño del modelo.
6. Diseñar e implementar la base de datos.
7. Pruebas a la Base de Datos.

La tesis está estructurada en los capítulos siguientes:

Capítulo1

El primer capítulo presenta el estado del arte sobre el tema de las Bases de Datos en la actualidad. Se hace una breve explicación de las tecnologías y tendencias actuales. También se especifican las herramientas utilizadas para diseñar Bases de Datos y en particular las que se utilizaron para implementar y diseñar la BD de Entidades y Cargos.

Capítulo2

En el segundo capítulo se describe y analiza la solución del diseño propuesto. Primeramente se argumentan los principales requisitos no funcionales. Luego se analiza diagrama de clases persistente, se crea el modelo de Datos, y se genera la BD. Se hace una descripción de cada una de sus tablas.

Capítulo3

En el tercer capítulo se examina la validación teórica y funcional del diseño realizado, a través de la integridad, normalización y seguridad de la Base de Datos. También se tratan otros aspectos como el análisis de redundancia en la información, la trazabilidad de las acciones y las pruebas empleadas para comprobar la consistencia y buena estructuración de la BD.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Este capítulo presenta los principales fundamentos teóricos en los que se basa la presente investigación. Además se hace un análisis de las tecnologías y las tendencias actuales principalmente de las Bases de Datos (BD), donde se abundan los temas de las tendencias, técnicas, tecnologías y metodologías que se utilizan en el mundo para el tratamiento de las mismas. También se especifican las herramientas que son utilizadas para diseñar Bases de Datos y en particular las que se utilizaron para implementar y diseñar la BD de Entidades y Cargos.

1.2 Introducción a las Bases de Datos

La información ha sido siempre un recurso muy valorado. Desde las primeras civilizaciones se han archivado, manipulado y creado grandes cantidades de información en bibliotecas, monasterios por determinadas personas. En esos lugares se atesoraban conocimientos científicos, de medicina, literatura y otros más.

Los datos son la parte principal de todos los sistemas de información. El almacenamiento, la manipulación y recuperación de información en forma eficiente, es vital y estratégico para cualquier organización. Las bases de datos juegan un rol crítico en casi todas las áreas donde las computadoras son usadas. Sin embargo el soporte lógico que hace operables los Sistemas de Información y cooperan en la consistencia de los datos que se manejan, así como el acceso a los mismos son las Bases de Datos (BD).

Las Bases de Datos son el soporte fundamental de empresas e instituciones en sus cotidianos procesos de gestión, ya que manejan gran cantidad de información valiosa y diversa, que necesitan organizar para tener disponible en el momento de tomar decisiones y brindar servicios.

Las Bases de Datos son componentes esenciales de los Sistemas de Información, pero son conocidas de mucho antes en las ciencias de la informática. Las BD surgen desde mediados de los años setenta en una conferencia organizada por el System Development Corporation en Santa Mónica (California) para referirse a “un almacén de todos los datos de interés y valor para los usuarios del sistema de información”. Codd propuso el modelo relacional, este modelo es el que ha marcado la línea de investigación por muchos años, ahora se encuentran los modelos orientados a objetos.

El avance tecnológico de la computación ha alterado la vida cotidiana de la sociedad y cambiado los modos de proceder e inclusive de vivir. Las computadoras se han convertido en una herramienta indispensable para la realización de tareas y la solución de problemas, tanto así, que sin ellas en la actualidad muchas de las actividades colapsarían.

1.2.1 Definición de Base de Datos

Una Base de Datos es un conjunto de datos relacionados entre sí. Se entiende por *dato*, los hechos conocidos, que pueden registrarse y que tienen significado implícito. Ejemplo: Nombre del Producto, Código del Producto. Según esta definición el conjunto de palabras que conforman un libro es una BD pues estarían todas relacionadas entre sí. Una definición más rigurosa que la anterior es la siguiente:

Una *BD* es un conjunto de datos que tiene las siguientes propiedades implícitas:

- Representa algún aspecto del mundo real, llamado mini mundo o universo de discurso. Las modificaciones del mini mundo se reflejan en la BD.
- Es un conjunto de datos lógicamente coherentes, con un cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una BD.
- Una BD se diseña, construye y puebla con datos para propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a distintos usuarios.

O sea, una BD tiene:

- Una fuente de la cual se derivan los datos.
- Cierta grado de interacción con los hechos del mundo real.
- Un público activamente interesado en el contenido de la BD.
- Su tamaño es variado.
- Debe ser posible buscar, obtener y actualizar los datos siempre que sea necesario. (1)

El software que permite la utilización y/o actualización de los datos almacenados en una o varias Bases de Datos, desde diferentes puntos de vista a la vez, por uno o varios usuarios, se denomina Sistema Gestor de Bases de Datos. (5)

1.3 Modelos de Bases de Datos.

1.3.1 Bases de Datos Jerárquicas

Este modelo utiliza árboles para la representación lógica de los datos. Este árbol está compuesto de unos elementos llamados nodos. El nivel más alto del árbol se denomina raíz. Cada nodo representa un registro con sus correspondientes campos.

La representación gráfica de este modelo se realiza mediante la creación de un árbol invertido, los diferentes niveles quedan unidos mediante relaciones.

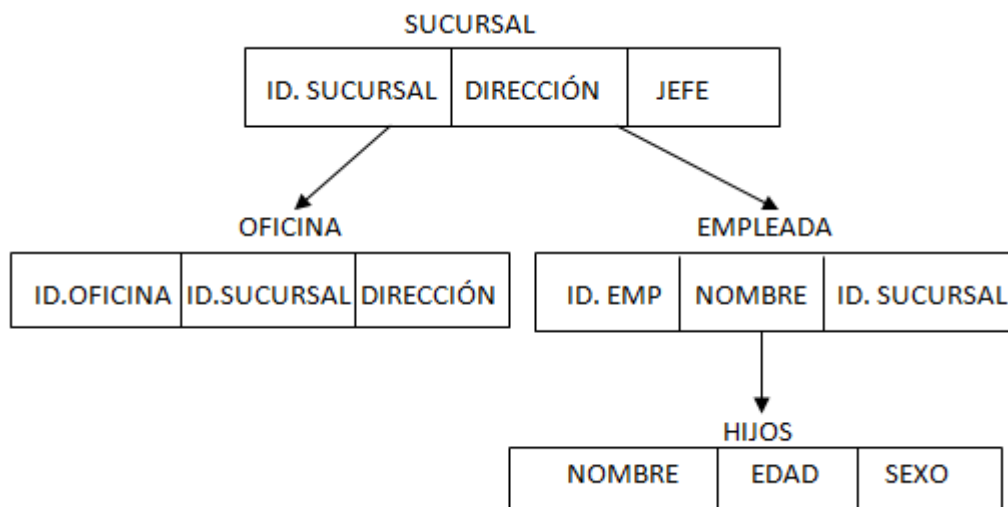


Figura 1

En este modelo solo se pueden representar relaciones 1: M, por lo que presenta varios inconvenientes:

- No se admiten relaciones N:M
- Un segmento hijo no puede tener más de un padre.
- No se permiten más de una relación entre dos segmentos.
- Para acceder a cualquier segmento es necesario comenzar por el segmento raíz

El árbol se debe de recorrer en el orden designado.

1.3.2 Bases de Datos de Red

En este modelo las entidades se representan como nodos y sus relaciones son las líneas que los unen. En esta estructura cualquier componente puede relacionarse con cualquier otro. A diferencia del modelo jerárquico, en este modelo, un hijo si puede tener varios padres.

Los conceptos básicos en el modelo en red son:

- El tipo de registro, que representa un nodo.
- Elemento, que es un campo de datos.
- Agregado de datos, que define un conjunto de datos con nombre.

Este modelo de datos permite representar relaciones N: M.

1.3.3 Bases de Datos Relacionales

Este modelo es el más utilizado actualmente ya que utiliza tablas bidimensionales para la representación lógica de los datos y sus relaciones.

Algunas de sus principales características son:

- Puede ser entendido y usado por cualquier usuario.
- Permite ampliar el esquema conceptual sin modificar las aplicaciones de gestión.
- Los usuarios no necesitan saber donde se encuentran los datos físicamente.

El elemento principal de este modelo es la relación que se representa mediante una tabla. (2)

1.3.4 Bases de Datos Orientadas a Objetos

A finales de los 80's aparecieron las primeras BDOO, es una base de datos inteligente. Soporta el paradigma orientado a objetos almacenando datos y métodos, y no sólo datos. Está diseñada para ser

eficaz, desde el punto de vista físico, para almacenar objetos complejos. Evita el acceso a los datos; esto es mediante los métodos almacenados en ella. Es más segura ya que no permite tener acceso a los datos (objetos); esto debido a que para poder entrar se tiene que hacer por los métodos que haya utilizado el programador (3)

La Orientación a Objetos puede describirse como el conjunto de disciplinas (ingeniería) que desarrollan y modelizan software que facilita la construcción de sistemas complejos a partir de componentes, llamados objetos.

- En la programación orientada a objetos todo se organiza en torno a los objetos.
- Un objeto tiene un nombre y posee datos y procedimientos. En cierto sentido, un objeto es un programa virtual.

Dado ciertos datos de entrada, el objeto aplica los procedimientos sobre dichos datos y genera una salida (4)

1.4 Tendencias actuales y futuras

La utilización de Bases de Datos como plataforma para el desarrollo de Sistemas de Aplicación en las organizaciones se ha incrementado notablemente en nuestros días. Actualmente las BD ofrecen numerosas funcionalidades de gran importancia para el almacenamiento y la manipulación de la información en las instituciones, pero siguen creciendo las demandas de nuevas operaciones que faciliten una mayor eficiencia en el manejo de los datos. Esto trae consigo el desarrollo de técnicas y herramientas que brinden una solución eficiente a las solicitudes planteadas por los usuarios.

Algunas de las tendencias actuales y futuras de Bases de Datos son citadas a continuación.

- La explotación efectiva de la información brinda ventajas competitivas a las organizaciones.
- El uso de las Bases de Datos Distribuidas y Multidimensionales se incrementa de manera considerable en la medida en que la tecnología de comunicación de datos brinde más facilidades para ello.
- El uso de las Bases de Datos facilita y soporta en gran medida a los sistemas de información para la toma de decisiones.

- Los lenguajes de consulta (SQL) permitirán el uso del lenguaje natural para solicitar información de la Base de Datos, haciendo más rápido y fácil su manejo.

En los últimos años se ha visto un gran crecimiento en la capacidad de generación y almacenamiento de información, debido a la creciente automatización de procesos. Desgraciadamente no ha existido un desarrollo equivalente en las técnicas de análisis de información, por lo que existe la necesidad de una nueva generación de técnicas y herramientas computacionales con la capacidad de asistir a usuarios en el análisis automático e inteligente de datos. El procesar automáticamente grandes cantidades de datos para encontrar conocimiento útil para un usuario y satisfacerle sus metas, es el objetivo principal del área de Descubrimiento de Conocimiento en Bases de Datos o KDD (Knowledge Discovery from Data Base). Este es el campo que está evolucionando para proporcionar soluciones al análisis automatizado de datos.

1.5 Análisis y descripción de las diferentes herramientas a usar

1.5.1 Sistemas Gestores de Bases de Datos (SGBD)

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.

1.5.1.1 Oracle

Es un sistema de gestión de base de datos fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos que existe.

Ventajas:

- Soporte de transacciones.

- Gran estabilidad y seguridad.
- Escalabilidad.
- Es multiplataforma.

Inconveniente:

- Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias).
- Otro aspecto a criticar es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.
- Aunque ha tenido un amplio dominio en el mercado de servidores sufre la competencia del Microsoft SQL Server de Microsoft y con licencia libre como PostgreSQL, MySql o Firebird. (6)

1.5.1.2 MySQL

Es uno de los Sistemas Gestores de bases de Datos más populares desarrollados bajo la filosofía de código abierto. MySQL tiene como una de sus principales ventajas la velocidad en la lectura de datos, pero a costa de eliminar un conjunto de facilidades que presentan otros SGBD: integridad referencial, bloqueo de registros, procedimientos almacenados, entre otros. En recientes versiones de MySQL se incluyen algunas de estas características, pero indudablemente esto va en detrimento de la velocidad, como se expresa anteriormente es un SGBD basado en Open Source (Código abierto) diseñado para los sistemas Unix, aunque existen versiones para Windows.

Ventajas:

- Diseñado en vistas a la velocidad.
- Consume muy pocos recursos de CPU y memoria. Muy buen rendimiento.
- Tamaño del registro sin límite.
- Buena integración con PHP.
- Utilidades de administración (phpMyAdmin).

- Buen control de acceso usuarios-tablas-permisos.
- Trabaja bajo diferentes plataformas: AIX 4x 5x, Amiga, BSDI, Digital Unix 4x, FreeBSD 2x 3x 4x, HP-UX 10.20 11x, Linux 2x, Mac OS, NetBSD, Novell NetWare 6.0, OpenBSD 2.5, OS/2, SCO OpenServer, SCO UnixWare 7.1.x, SGI Irix 6.x, Solaris 2.5, SunOS 4.x, Tru64 Unix y Windows 9x, Me, NT, 2000, XP, 2003.
- Desarrollo de APIs para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.

Inconvenientes:

- No soporta subconsultas.
- No soporta transacciones.
- No soporta vistas.
- Se hace inestable cuando contiene gran cantidad de datos.

1.5.1.3 PostgreSQL

Está considerado uno de los SGBD de código abierto más avanzado del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales de alto calibre tales como Oracle. Es un SGBD objeto-relacional, ya que aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Su avanzada funcionalidad se pone de manifiesto con las consultas SQL declarativas, el control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia y valores no atómicos (atributos basados en vectores y conjuntos). Es altamente extensible: soporta operadores y tipos de datos definidos por el usuario. Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92. Cuenta con una API (del inglés Application Program Interface) flexible lo cual ha permitido dar soporte para el desarrollo con PostgreSQL en diversos lenguajes de programación.

Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta

estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- o Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Otras características

- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (*foreign keys*).
- Disparadores (*triggers*).
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientación a objetos o la programación funcional. Los disparadores (*triggers* en inglés) son funciones enlazadas a operaciones sobre los datos. Algunos de los lenguajes que se pueden usar son los siguientes:

- Un lenguaje propio llamado PL/pgSQL (similar al PL/SQL de oracle).

- C.
- C++.
- o Java (via PL/Java).
- PL/Perl.
- o PL/PHP.
- o PL/Python.
- o PL/Ruby.
- o PL/sh.
- o PL/Tcl.
- o PL/Scheme. PL/Scheme.
- Lenguaje para aplicaciones estadísticas R through PL/R.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta en SQL (query en inglés)

Ventajas con respecto a otros Sistemas gestores de base de datos.

- COSTO.
- Estabilidad, Confiabilidad.
- Funcionalidad (es un motor de bases de datos *Avanzado* y con herramientas gráficas como PgAdmin, phpPgAdmin la administración de la base de datos es sencilla).

Ventajas:

- Soporta la configuración automática y la auto-optimización.
- Administración multiservidor para un gran número de servidores.
- Gran variedad de opciones de duplicación de cualquier base de datos.
- Acceso universal a los datos (Universal Data Access).
- Fácil de usar.
- Escalabilidad. Permite realizar un escalamiento hasta 32 CPU y 64 gigabytes (GB) de RAM, siendo capaz de manejar al máximo multiprocesamiento simétrico aprovechando al máximo el hardware.

- Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server.
- Posee los mejores registros de los benchmarks independientes (TCP) tanto en transacciones totales como en coste por transacción.
- Gestión: Con una completa interfaz gráfica que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.
- Múltiples instancias y Failover.
- Integración con la Web
- Acceso Web a datos
- Soporte para XML (cláusula FOR XML)

Desventajas:

- Licencias con costos altos.
- Plataformas Windows13

1.5.2 Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software, es decir, van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Estas tecnologías se dividen en dos grandes grupos; las metodologías ligeras, en la que encontramos a XP (extreme Programming), Crystal Light Methods y metodologías pesadas como son, SW-CMM (SoftWare Capability Maturity Model), RUP (Rational Unified Process). Las primeras se basan en la idea de conseguir el objetivo común por medio de orden y documentación, mientras que las segundas tratan de lograrlo por medio de la comunicación directa e inmediata entre aquello que intervienen en el proceso. Analizaremos a continuación tres de las más conocidas con sus características, ventajas y desventajas.

1.5.2.1 Proceso Unificado de Rational (RUP)

Proceso Unificado de Rational, en inglés Rational Unified Process es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos en la actualidad, pues está pensada para adaptarse a una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipo de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. Es el resultado de la experiencia de más de 30 años de trabajo y los autores [James Rumbaugh, Grady Booch e Ivar Jacobson] confirman que es la solución al problema del software. RUP se divide en cuatro fases (Inicio, Elaboración, Construcción, Transición) y presenta nueve flujos de trabajos, de ellos seis son de ingeniería (Modelamiento del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, Instalación o Distribución) y tres de apoyo (Configuración y administración del cambio, Administración de proyectos, Ambiente). Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software y tiene la ventaja de venir acompañada de una potente herramienta que soporta todos los procesos básicos de RUP: Rational Rose Enterprise Edition 2003.

1.5.2.2 Programación Extrema (XP)

La metodología Programación Extrema, en inglés Extreme Programming consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Ha sido diseñada para solucionar el eterno problema del desarrollo de software por encargo: entregar el resultado que el cliente necesita a tiempo.

La metodología se basa en:

- Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir.
- Re fabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Entre sus características podemos resaltar: los diseñadores y programadores se comunican efectivamente con el cliente y entre ellos mismos, los diseños del software se mantienen sencillos y libres de complejidad o pretensiones excesivas, el software es liberado en entregas frecuentes tan pronto como sea posible, los cambios se implementan rápidamente tal y como sea posible, es usada en proyectos de corto plazo, entre otras. Con estas características no es sorprendente que XP sea la metodología más apropiada para un entorno caracterizado por requerimientos cambiantes originados por un mercado fluctuante y los propios avances de la tecnología y los negocios.

1.5.3 Herramientas para el lenguaje de modelado visual

1.5.3.1 Rational Rose Enterprise

Herramienta de desarrollo basada en modelos que se integra con las bases de datos y los IDE de las principales plataformas del sector

Si necesita soporte para UML 2.0 y el mejor modelado de datos, incluido el modelo de entidad-relación, consulte IBM Rational Software Architect, IBM Rational Software Modeler o IBM Rational Data y Application Modeling Bundle.

IBM Rational Rose Enterprise es uno de los productos más completos de la familia Rational Rose.

Todos los productos de Rational Rose dan soporte a Unified Modeling Language (UML), pero no son compatibles con las mismas tecnologías de implementación.

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

Incluye también estas funciones:

- Soporte a modelos de análisis, ANSI C++, Rose J y Visual C++ según el documento "Design Patterns: Elements of Reusable Object-Oriented Software".

- Los componentes del modelo se pueden controlar independientemente, lo que permite una gestión y un uso de modelos más granular.
- NUEVO: Soporte para compilación y descompilación de las construcciones más habituales de Java 1.5.
- Generación de código en lenguaje Ada, ANSI C++, C++, CORBA, Java y Visual Basic, con funciones configurables de sincronización entre los modelos y el código.
- Soporte para Enterprise Java Beans 2.0.
- Funciones de análisis de calidad de código.
- Complemento de modelado Web que incluye funciones de visualización, modelado y herramientas para desarrollar aplicaciones Web.
- Modelado en UML para diseñar bases de datos, que integra los requisitos de datos y aplicaciones mediante diseños lógicos y analíticos.
- Creación de definiciones de tipo de documento DTD en XML.
- Integración con otras herramientas de desarrollo de IBM Rational.
- Integración con cualquier sistema de control de versiones compatible con SCC, como IBM Rational ClearCase.
- Posibilidad de publicar en la Web, modelos e informes para mejorar la comunicación entre los miembros del equipo.

1.5.3.2 Visual Paradigm

Visual Paradigm, herramienta CASE que ofrece un entorno de creación de diagramas para UML. El diseño es centrado en casos de uso y enfocado al negocio, lo que permite generar software de gran calidad. Esta herramienta usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación. Tiene capacidad para la ingeniería directa e inversa en Java, C++, PHP, entre otros lenguajes y disponibilidad de múltiples versiones para cada necesidad. Es multiplataforma y muy útil para la generación de código fuente en PHP. Tiene la capacidad de crear el esquema de clases a partir de una Base de Datos y crear la de Base de Datos a partir del esquema de clases. Incorpora el soporte para trabajo en equipo, proporcionando que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros de equipo.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y

despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Algunas de sus características:

- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo con CVS y Subversión (nueva característica).
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI (nueva característica).
 - Ingeniería de ida y vuelta.
 - Ingeniería inversa - Código a modelo, código a diagrama.
 - Ingeniería inversa Java, C++, Esquemas XML, XML,.NET exe/dll, CORBA IDL.
 - Generación de código - Modelo a código, diagrama a código.
 - Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
 - Diagramas EJB - Visualización de sistemas EJB.
 - Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.
 - Diagramas de flujo de datos.
 - Soporte ORM - Generación de objetos Java desde la base de datos.
 - Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
 - Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
 - Generador de informes para generación de documentación.
 - Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.

- Importación y exportación de ficheros XMI.
- Integración con Visio - Dibujo de diagramas UML con plantillas (stencils) de MS Visio.
- Editor de figuras.

1.6 Herramienta para la administración y desarrollo de PostgreSQL Database Server

1.6.1 EMS PostgreSQL Manager

El encargado del ccsme SQL para PostgreSQL es una herramienta de gran alcance para la administración y el desarrollo del servidor de la base de datos de PostgreSQL. El encargado de PostgreSQL trabaja con cualesquiera versiones de PostgreSQL hasta 8.1 y ayuda todo de las características más últimas de PostgreSQL incluyendo los tablespaces, nombres de la discusión en funciones, etcétera. Ofrece a un montón de herramientas de gran alcance para los usuarios experimentados tales como diseñador visual de la base de datos, constructor visual de la pregunta, y redactor de gran alcance de la GOTA para satisfacer todas sus necesidades. El encargado de PostgreSQL tiene un nuevo interfaz utilizador gráfico avanzado con el sistema bien-descrito del mago, así que claro en uso que incluso no confundirán a un newbie con él.

1.7 Conclusiones

Por todo lo expuesto anteriormente se ha coincidido con los lineamientos, principios y normativas para el desarrollo del software en las FAR que establece que todas sus aplicaciones informáticas utilicen RUP como Proceso de Desarrollo de Software en la creación de sus proyectos. Además por sus características y ventajas como Gestor de Base de datos a PostgreSQL, como Herramienta Case, Visual Paradigm y para la administración y desarrollo de la BD, EMS SQL Manager for PostgreSQL.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

En este capítulo se definen las principales actividades que posibilitaron la construcción de la Base de Datos de Entidades y Cargos, obteniendo al final de las mismas un resultado final, el diseño de la BD.

Se seleccionaron los requisitos no funcionales del modelo de datos informático propuesto y se argumentó dicha selección, los cuales permitieron reconocer las entidades y atributos de la BD. Luego se describió y fundamentó la arquitectura a utilizar para este modelo para una buena comprensión del tema del acceso a datos desde la BD.

También se obtuvo el diagrama de clases persistentes a través del diagrama de clases del diseño del cual se hizo un estudio para poder obtener un eficiente Modelo de Datos. Por último se elaboró el diagrama entidad relación de la BD y se describieron sus tablas.

2.2 Selección y argumentación de los requisitos no funcionales del sistema propuesto

A continuación se muestran los requisitos no funcionales que corresponden con el desarrollo de la Base de Datos de Entidades y Cargos.

1. Rendimiento.

La base de datos debe consumir lo mas mínimo posible de recursos de la maquina para que sea capaz de dar respuestas rápidas.

2. Confiabilidad.

La información almacenada deberá estar protegida de acceso no autorizado.

3. Fiabilidad.

Debe ser garantizado el resguardo de los datos, así como una copia de la base de datos periódicamente, de forma tal que no haya una pérdida total de datos en casos de accidentes.

4. Legales.

El modelo se debe basar en el documento que expone las normativas para el desarrollo de software en las FAR. La mayoría de las herramientas de desarrollo serán libres y del resto, las licencias deben estar avaladas.

2.3 Descripción de la representación gráfica de la estructura y la composición

Con el objetivo de dar continuidad a los subsistemas en desarrollo, a continuación detallamos las estructuras de datos que representan la **estructura** y **composición** de las entidades de las FAR, que constituye el punto de partida para estos y otros sistemas que se desarrollen.

La **estructura** está formada por un esquema jerárquico de órganos y entidades agrupadas por su tipo, que técnicamente se representan en un árbol. La **composición** por su parte representa los recursos materiales y humanos que deben llevar una entidad, lo que comúnmente denominamos cargos de la plantilla (Figura. 2).

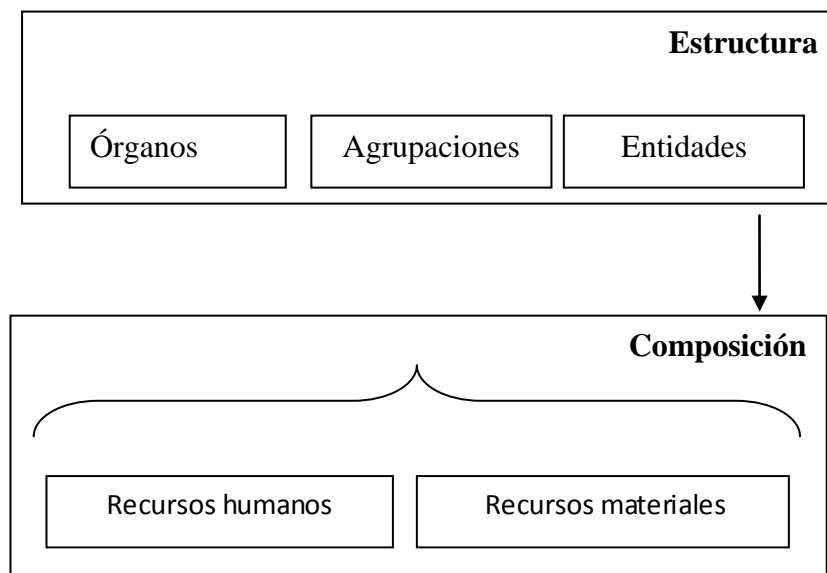


Figura 2 Esquema lógico de la estructura.

Por razones de integridad y rendimiento se decidió dividir la estructura en tres partes: las entidades, los órganos y agrupaciones que están jerárquicamente por encima de las entidades y los órganos y agrupaciones que están jerárquicamente por debajo de las entidades. Las entidades son aquellos órganos para los cuales se crea una plantilla (Figura 3).

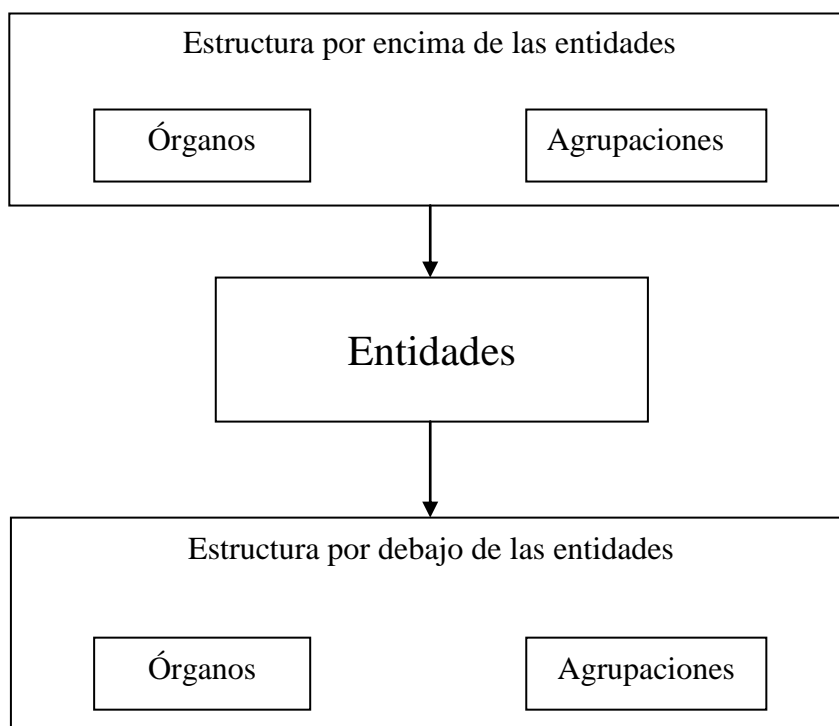


Figura 3 División de la estructura

A continuación graficaremos un ejemplo para poder comprender la idea descrita anteriormente.

- * Ejército
- ** Unidades de aseguramiento
- *** Batallón de Construcción 1010
- **** Órganos de dirección
- ***** Jefatura Batallón
- **** Órganos de producción

- ***** Brigada de carpintería
- ***** Brigada de albañilería
- ***** Brigada de electricidad
- ** Unidades combativas
 - *** Batallón de Infantería 9010
 - **** Órganos de dirección
 - ***** Jefatura Batallón
 - **** Órganos de combate
 - ***** Compañía Infantería
 - ***** Compañía exploración

- * Órganos por encima de las entidades
- ** Agrupaciones por encima de las entidades
- *** Entidades
 - **** Agrupaciones por debajo de las entidades
 - ***** Órganos por debajo de las entidades

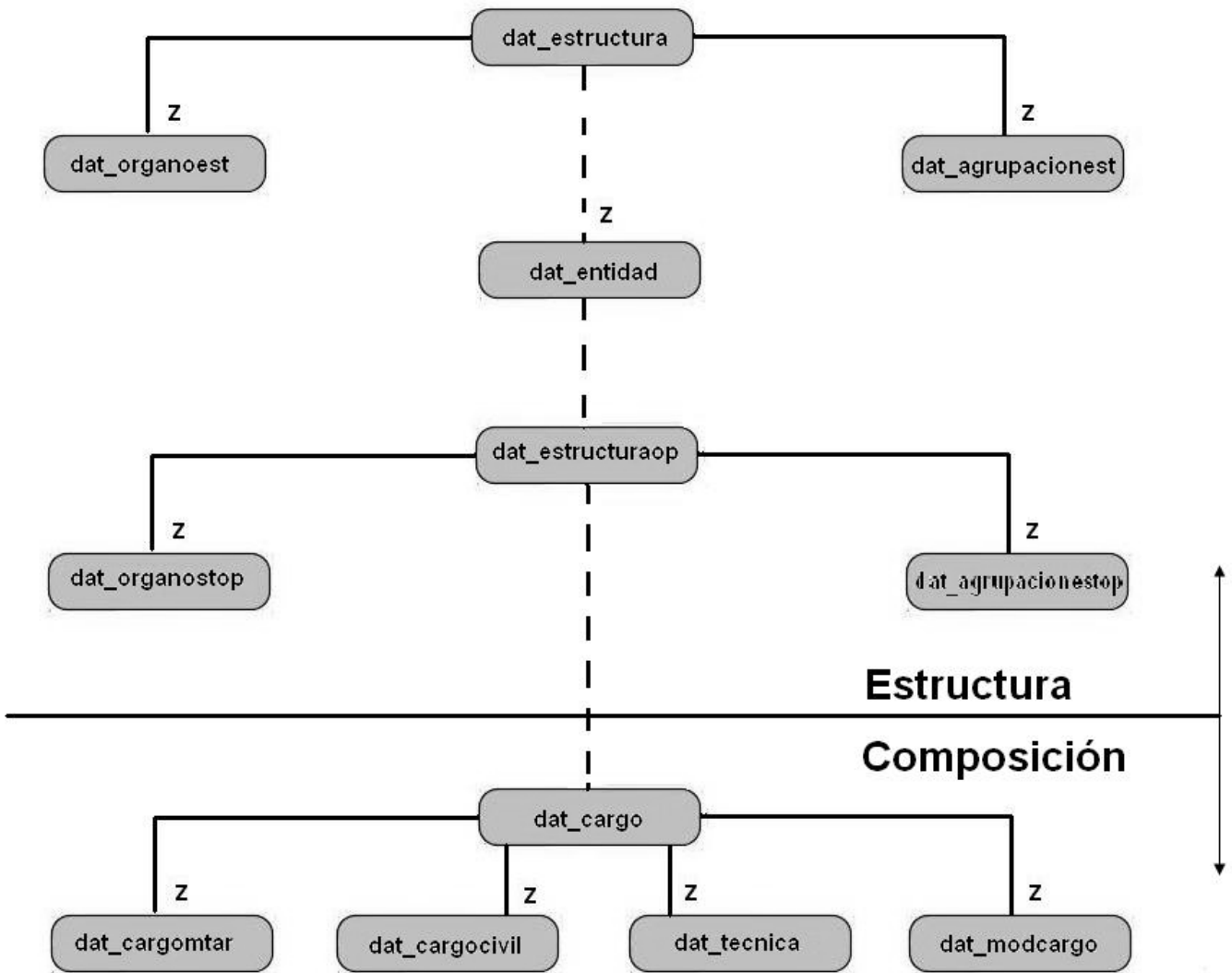


Figura 4 Diseño físico de la estructura y composición.

En lo adelante se divide la explicación en dos partes: estructura y composición, definiendo en cada caso la estructura de datos y el diseño lógico de las entidades. Al final del documento podremos encontrar un resumen de todas las entidades con sus atributos.

Estructura.

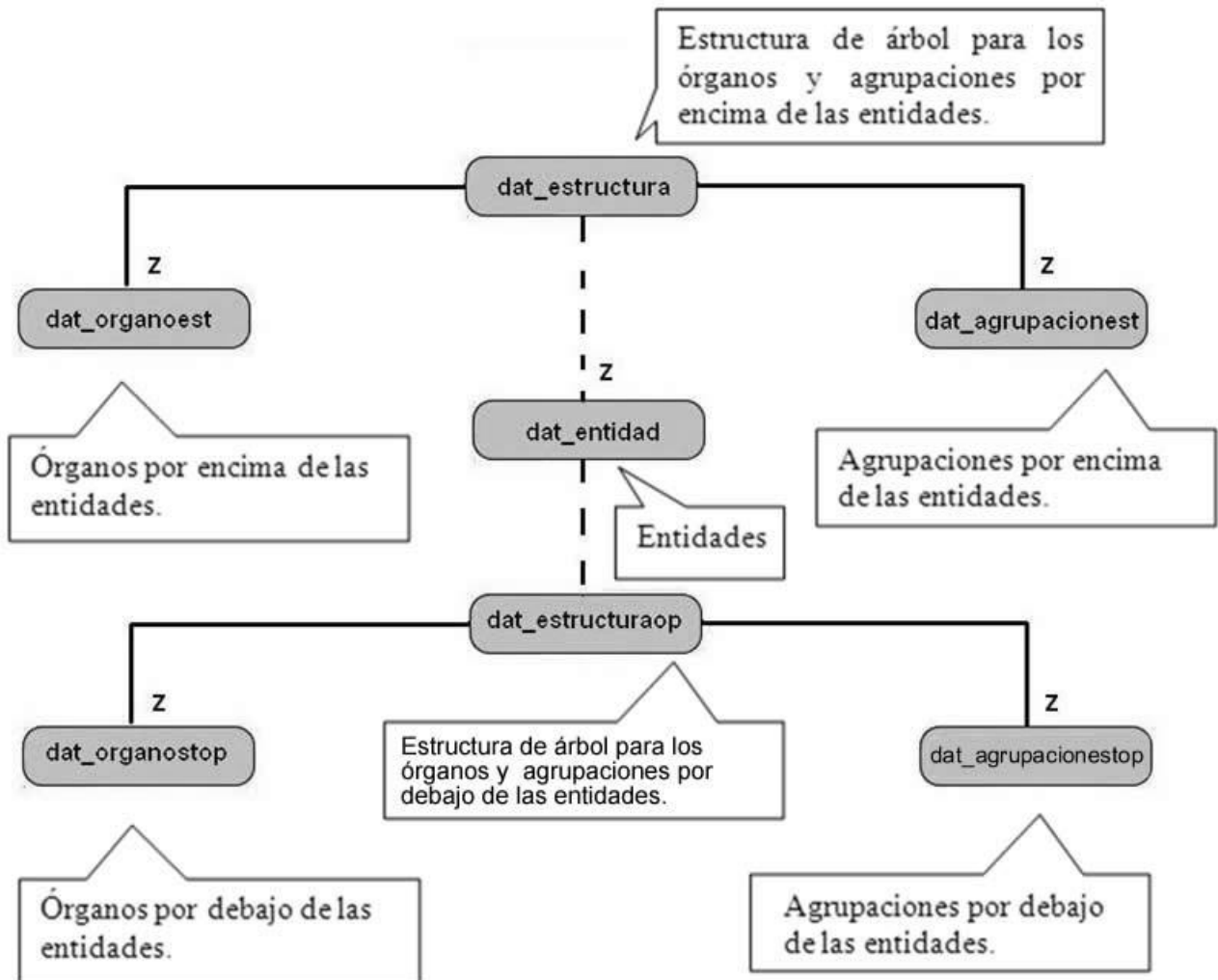


Figura 5 Principales entidades de la estructura.

Las entidades “dat_estructura” y “dat_estructuraop” almacenan la estructura de árbol para los órganos y agrupaciones.

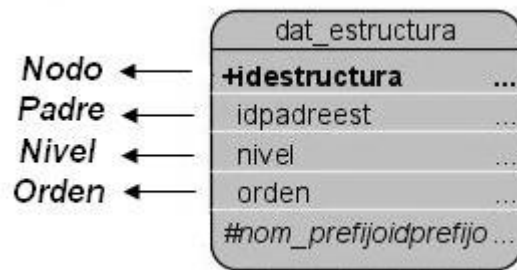


Figura 6 Estructura de árbol para los órganos y agrupaciones

El atributo “idestructura” constituye el identificador del órgano o agrupación (nodo en el árbol) e “idpadreest” el identificador del nodo padre en el árbol, “nivel” el nivel de profundidad del nodo y “orden” el orden del nodo. El atributo “orden” no se repite en la estructura y toma su valor teniendo en cuenta el nivel y posición de los nodos, de arriba hacia abajo y de izquierda a derecha, tal y como se muestra en la Figura 7.

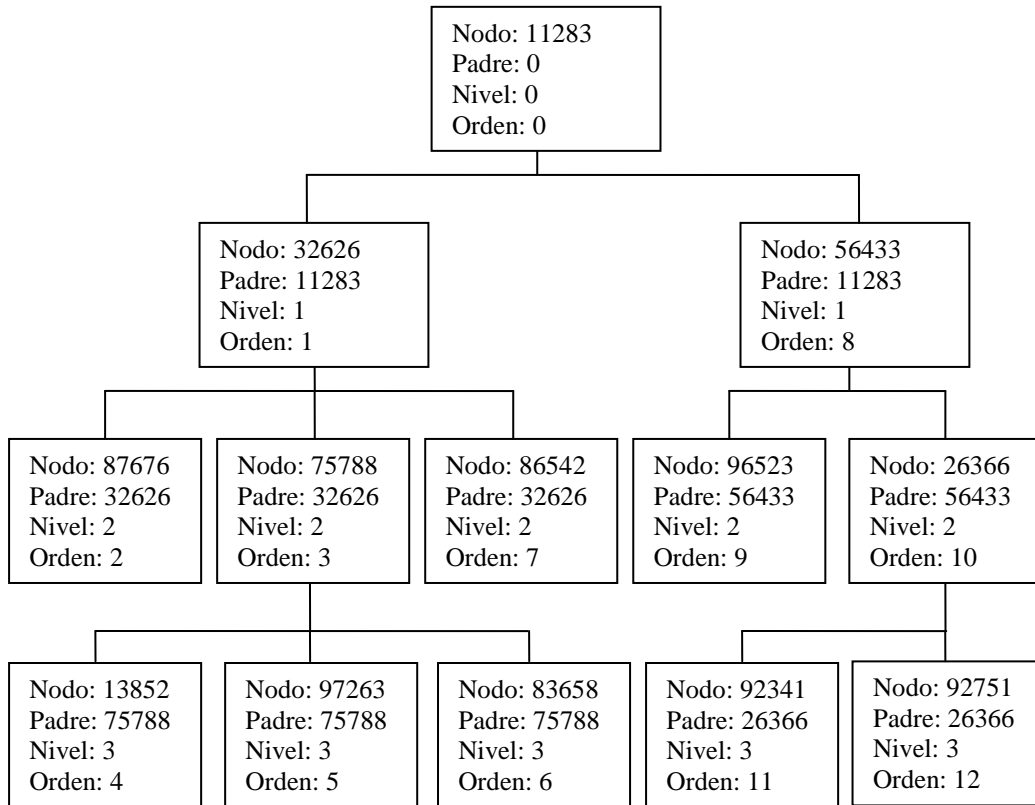


Figura 7 Ejemplo de una estructura de árbol

Esta estructura de árbol brinda todas las necesidades informativas que se requieren y se aplica a las entidades que almacenan los órganos y agrupaciones. Por ejemplo si queremos conocer todos los nodos subordinados al nodo 32626, buscamos el orden del nodo a la derecha en el mismo nivel (nodo 56433 orden 8) y entonces el resultado son todos los nodos con orden mayor que 1 y menor que 8.

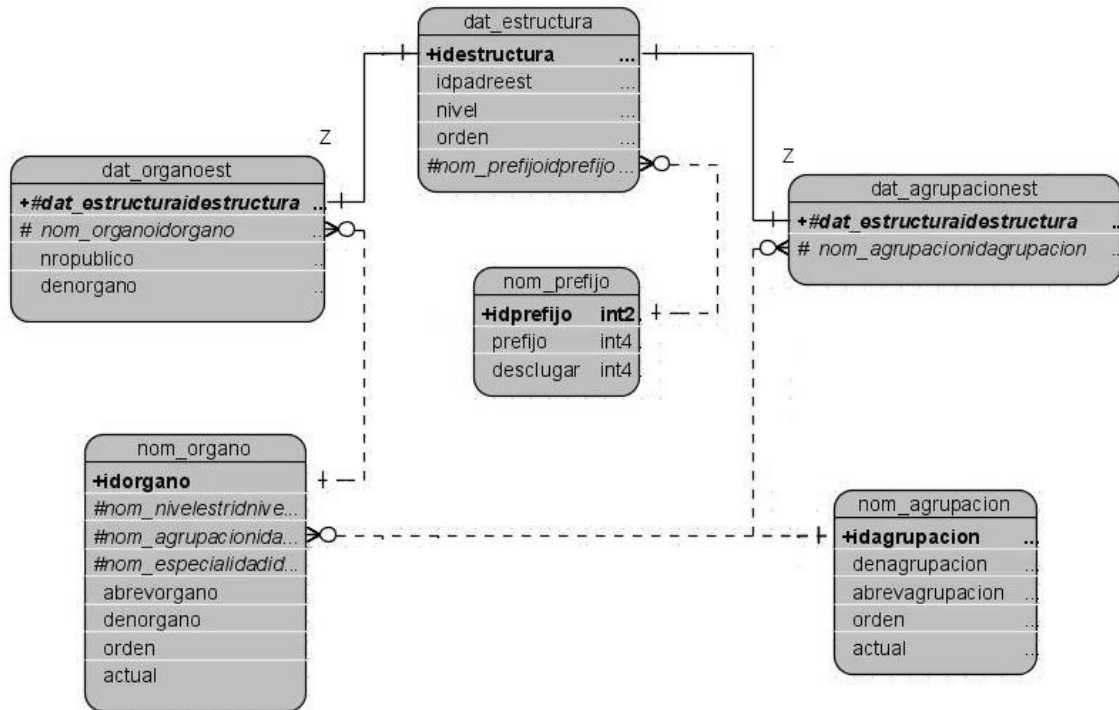


Figura 8 Estructura de datos. Órganos y agrupaciones.

En la Figura 8 se muestra la estructura de datos que almacena los órganos y agrupaciones correspondientes a la jerárquica que existe por encima de las entidades. Entre las entidades “dat_estructura” y “dat_organos”-“dat_agrupaciones” existe una relación uno a uno. El número público que aparece en la entidad “dat_organos” se adquiere o hereda (o sea es igual) del número de su estado mayor, con el objetivo de poder conocer el número de la entidad superior.



Figura 9 Entidad “dat_entidad”.

La tabla “dat_entidad” contiene todas las entidades y sus atributos principales (Figura 9). Entre la entidad “dat_estructura” y “dat_entidad” existe una relación uno a muchos. El atributo “orden” es usado para especificar el orden de salida de los datos para mostrarlos o imprimirlos. El prefijo se utiliza para definir el lugar donde se genera la información.

La estructura que almacena los órganos y agrupaciones por debajo de las entidades (Figura 10) es similar a la mostrada en la Figura 8 especificándose en el caso de los órganos la cantidad para tiempo de paz (ctp) y de guerra (ctg). La relación existente entre la entidad “dat_entidad” y “dat_estructuraop” es de uno a muchos.

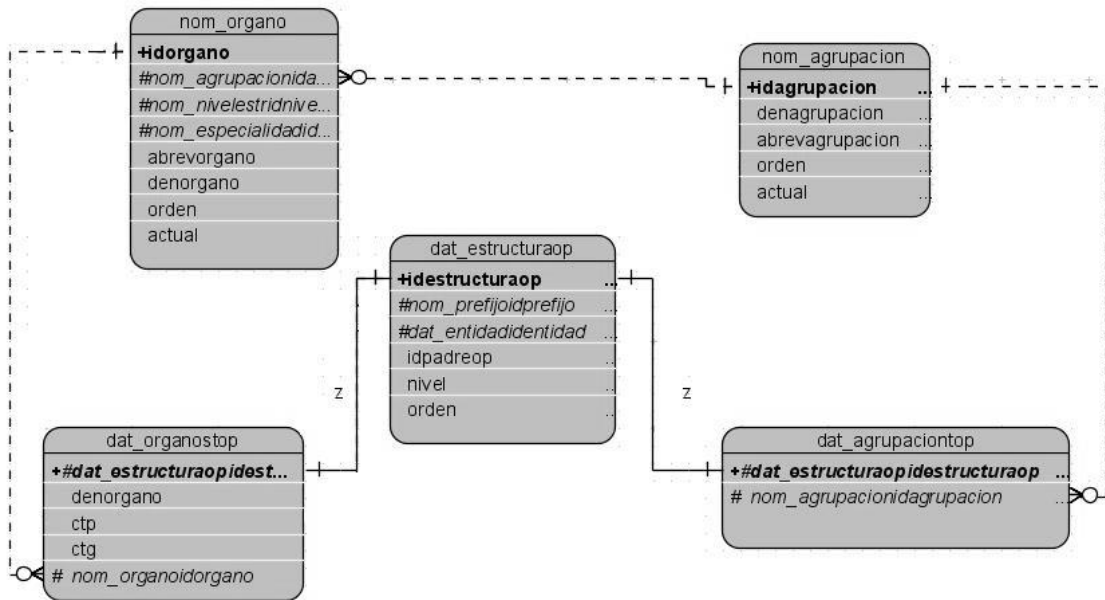


Figura 10 Estructura de datos. Órganos y agrupaciones.

Composición.

La estructura de datos que se muestra a continuación contiene las entidades donde se almacenan la cantidad de recursos materiales y humanos que deben llevar una entidad (Plantilla).

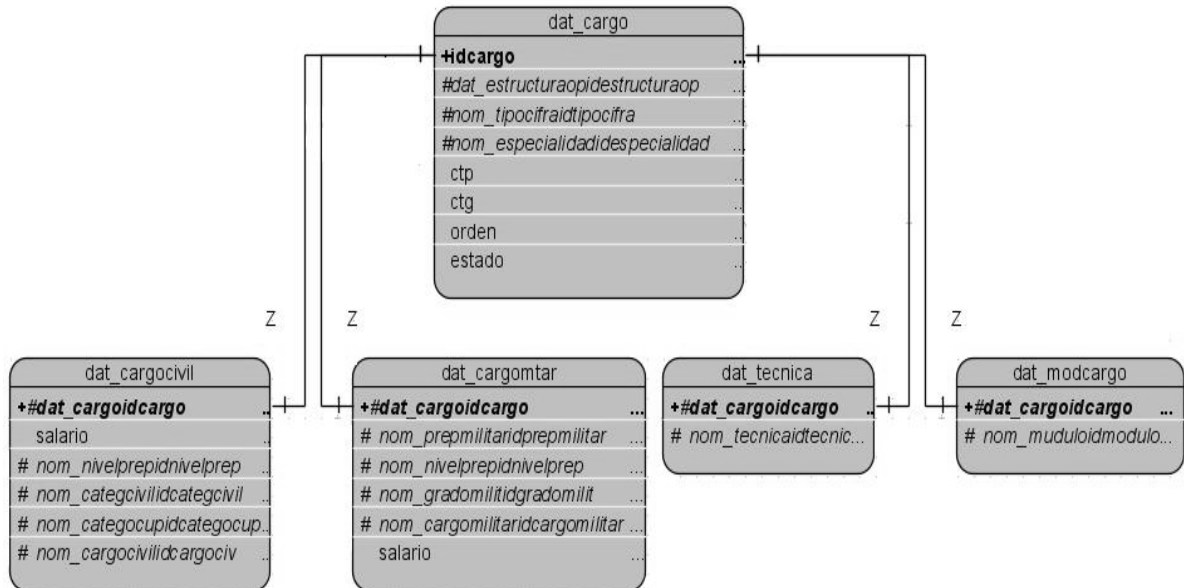


Figura 11 Composición de las entidades.

La entidad “dat_cargo” almacena los datos que son comunes para todos los cargos que forman la plantilla. El resto de las entidades almacenan la información que es específica para cada caso:

dat_cargocivil: información relativa a los cargos civiles.

dat_cargomtar: información relativa a los cargos militares.

dat_tecnica: información relativa al material de guerra (armamento, carros...). Se especifica uno a uno

dat_modcargo: información relativa al material de guerra (armamento, carros...). Se especifica mediante módulos.

La relación que existe entre la entidad “dat_cargo” y el resto de las entidades es uno a uno. Un cargo se identifica de forma única mediante el atributo “idcargo”.

La diferencia entre especificar la plantilla del material de guerra uno a uno y mediante módulos consiste en: uno a uno es cuando se define la plantilla por cada medio material, ejemplo- Carro de carga...4. Por módulos es cuando se define la cantidad para un grupo de medios agrupados en un

módulo, ejemplo-Modulo personal del soldado...6. En este último caso el modulo personal del soldado esta formado por: Fusil Automático, Mochila, municiones.

2.4 Descripción de la arquitectura y fundamentación

El diseño del modelo de datos informático de entidades y cargos está basado en una arquitectura orientada a objetos y servicios, para que pueda ampliarse con facilidad en caso de que las necesidades aumenten.

Se utiliza la arquitectura en capas, facilitando el desarrollo y posibilitando que si se produce algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Hay tres características importantes inherentes a los sistemas de bases de datos: la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos. En 1975, el comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los sistemas de bases de datos, que resulta muy útil a la hora de conseguir estas tres características.

El objetivo de la arquitectura de tres niveles es el de separar los programas de aplicación de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción distintos:

1. En el nivel interno se describe la estructura física de la base de datos mediante un esquema interno. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
2. En el nivel conceptual se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.

3. En el nivel externo se describen varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinados y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

La mayoría de los SGBD no distinguen del todo los tres niveles. Algunos incluyen detalles del nivel físico en el esquema conceptual. En casi todos los SGBD que se manejan vistas de usuario, los esquemas externos se especifican con el mismo modelo de datos que describe la información a nivel conceptual, aunque en algunos se pueden utilizar diferentes modelos de datos en los niveles, conceptual y externo.

Hay que destacar que los tres esquemas no son más que descripciones de los mismos datos pero con distintos niveles de abstracción. Los únicos datos que existen realmente están a nivel físico, almacenados en un dispositivo como puede ser un disco. En un SGBD basado en la arquitectura de tres niveles, cada grupo de usuarios hace referencia exclusivamente a su propio esquema externo. Por lo tanto, el SGBD debe transformar cualquier petición expresada en términos de un esquema externo a una petición expresada en términos del esquema conceptual, y luego, a una petición en el esquema interno, que se procesará sobre la base de datos almacenada. Si la petición es de una obtención (consulta) de datos, será preciso modificar el formato de la información extraída de la base de datos almacenada, para que coincida con la vista externa del usuario. El proceso de transformar peticiones y resultados de un nivel a otro se denomina correspondencia o transformación. Estas correspondencias pueden requerir bastante tiempo, por lo que algunos SGBD no cuentan con vistas externas.

La arquitectura de tres niveles es útil para explicar el concepto de independencia de datos que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior. Se pueden definir dos tipos de independencia de datos:

- La independencia lógica es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla. Si, por ejemplo, se reduce la base de datos eliminando una entidad, los esquemas externos que no se refieran a ella no deberán verse afectados.

· La independencia física es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Por ejemplo, puede ser necesario reorganizar ciertos ficheros físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos. Dado que la independencia física se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento, es más fácil de conseguir que la independencia lógica.

En los SGBD que tienen la arquitectura de varios niveles es necesario ampliar el catálogo o diccionario, de modo que incluya información sobre cómo establecer la correspondencia entre las peticiones de los usuarios y los datos, entre los diversos niveles. El SGBD utiliza una serie de procedimientos adicionales para realizar estas correspondencias haciendo referencia a la información de correspondencia que se encuentra en el catálogo. La independencia de datos se consigue porque al modificarse el esquema en algún nivel, el esquema del nivel inmediato superior permanece sin cambios, sólo se modifica la correspondencia entre los dos niveles. No es preciso modificar los programas de aplicación que hacen referencia al esquema del nivel superior.

Por lo tanto, la arquitectura de tres niveles puede facilitar la obtención de la verdadera independencia de datos, tanto física como lógica. Sin embargo, los dos niveles de correspondencia implican un gasto extra durante la ejecución de una consulta o de un programa, lo cual reduce la eficiencia del SGBD. Es por esto que muy pocos SGBD han implementado esta arquitectura completa.

2.5 Diagrama de clases persistentes

<i>Diagrama de clases persistentes</i>

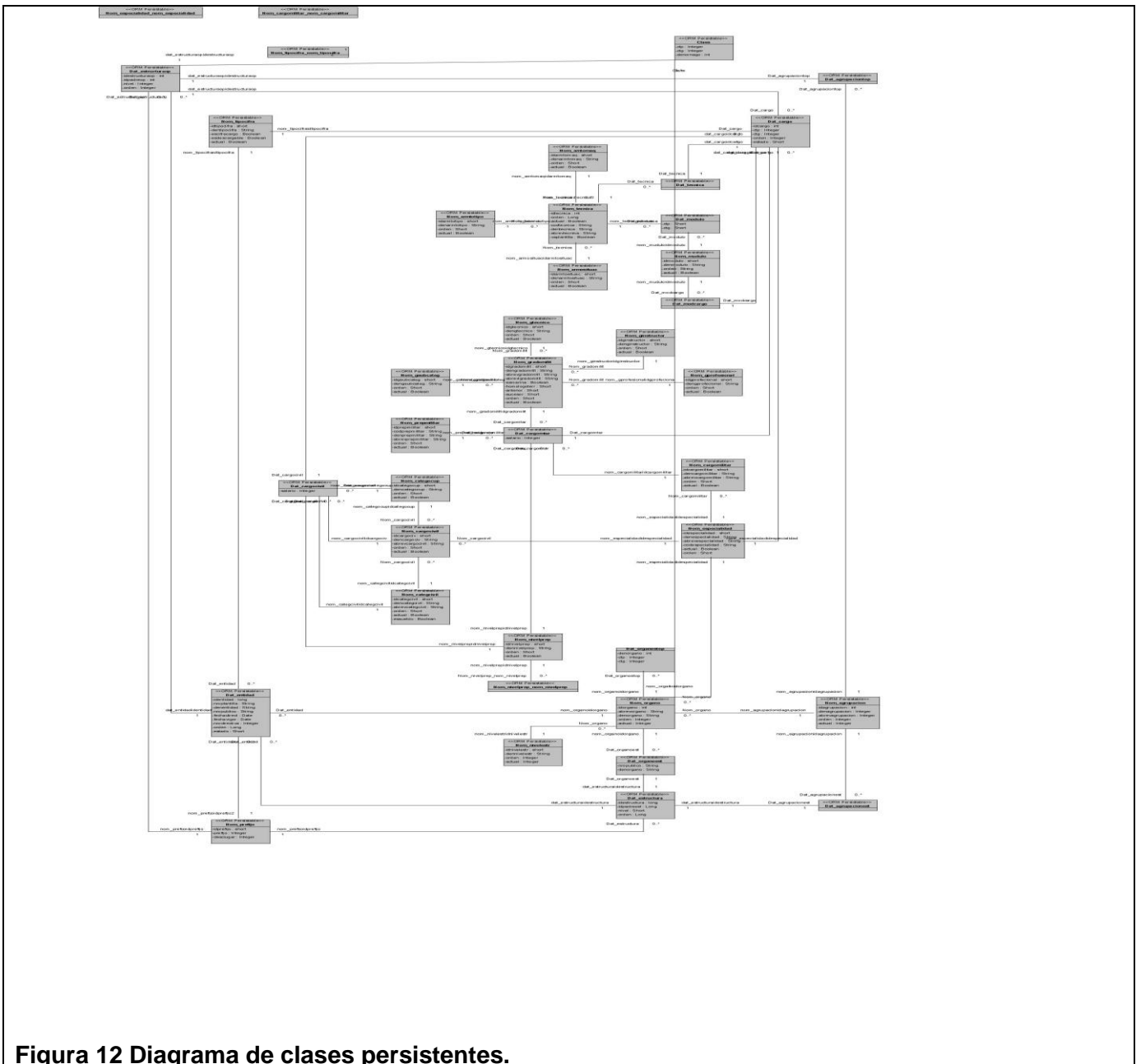


Figura 12 Diagrama de clases persistentes.

2.5.1 Representación de las clases persistentes

Tabla 1 Agrupacionest

Nombre: dat_agrupacionest	
Tipo de Clase: Entidad	
Atributo	Tipo
Idestructura	BIGINT
Idagrupacion	SMALLINT

Tabla 2 Agrupacionestop

Nombre: dat_agrupacionestop	
Tipo de Clase: Entidad	
Atributo	Tipo
Idestructuraop	BIGINT
Idagrupacion	SMALLINT

Tabla 3 Cargo

Nombre: dat_cargo	
Tipo de Clase: Entidad	
Atributo	Tipo
Idcargo	BIGINT
Idestructuraop	BIGINT
Idespecialidad	SMALLINT
Ctp	SMALLINT
Ctg	SMALLINT
Idtipocifra	SMALLINT
Idprefijo	SMALLINT
Orden	BIGINT
Estado	SMALLINT

Tabla 4 Entidad

Nombre: dat_entidad	
Tipo de Clase: Entidad	
Atributo	Tipo
Identidad	BIGINT
Idestructura	BIGINT
Idorgano	SMALLINT
Nroplantilla	CHAR(8)
Nropublico	CHAR(5)
Denentidad	VARCHAR(40)
Fechadirectiva	DATE
Fechavigor	DATE
Nrodirectiva	SMALLINT
Idtipocifra	SMALLINT
Idprefijo	SMALLINT
Orden	BIGINT
Estado	SMALLINT

Tabla 5 organoest

Nombre: dat_organosest	
Tipo de Clase: Entidad	
Atributo	Tipo
idestructura	BIGINT
idorgano	SMALLINT
nropublico	CHAR(4)
denorgano	VARCHAR(50)

Tabla 6 organostop

Nombre: dat_organostop	
Tipo de Clase: Entidad	
Atributo	Tipo
idestructuraop	BIGINT
idorgano	SMALLINT
denorgano	VARCHAR(50)
ctp	SMALLINT
ctg	SMALLINT

Tabla 7 Estructura

Nombre: dat_estructura	
Tipo de Clase: Entidad	
Atributo	Tipo
idestructura	BIGINT
idpadreest	BIGINT
nivel	SMALLINT
orden	BIGINT
idprefijo	SMALLINT

Tabla 8 Estructuraop

Nombre: dat_estructuraop	
Tipo de Clase: Entidad	
Atributo	Tipo
Idestructuraop	BIGINT
Identidad	BIGINT
Idpadreestop	BIGINT
Nivel	SMALLINT
Orden	BIGINT
Idprefijo	SMALLINT

2.6 Diagrama Entidad Relación de la BD

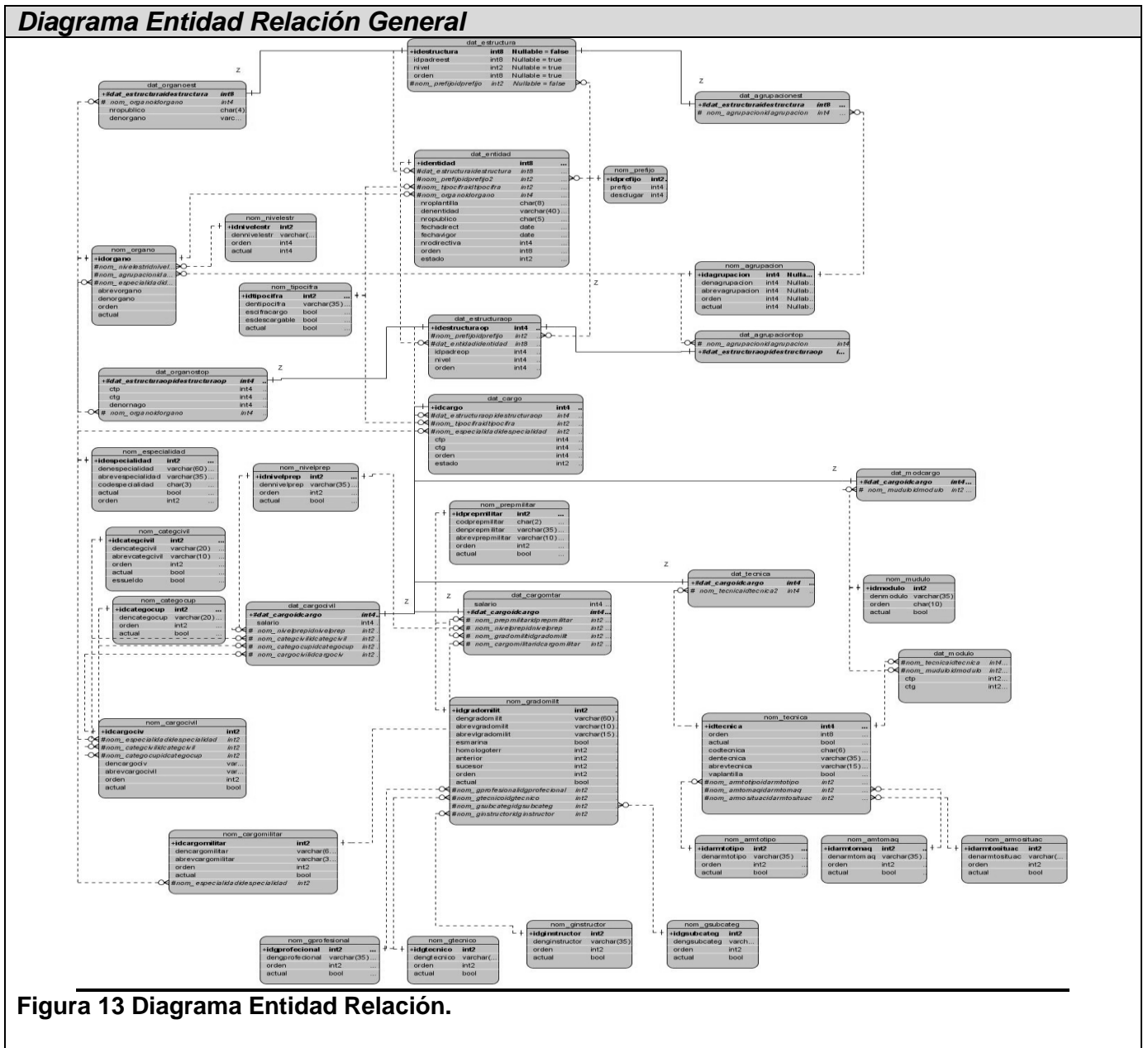


Figura 13 Diagrama Entidad Relación.

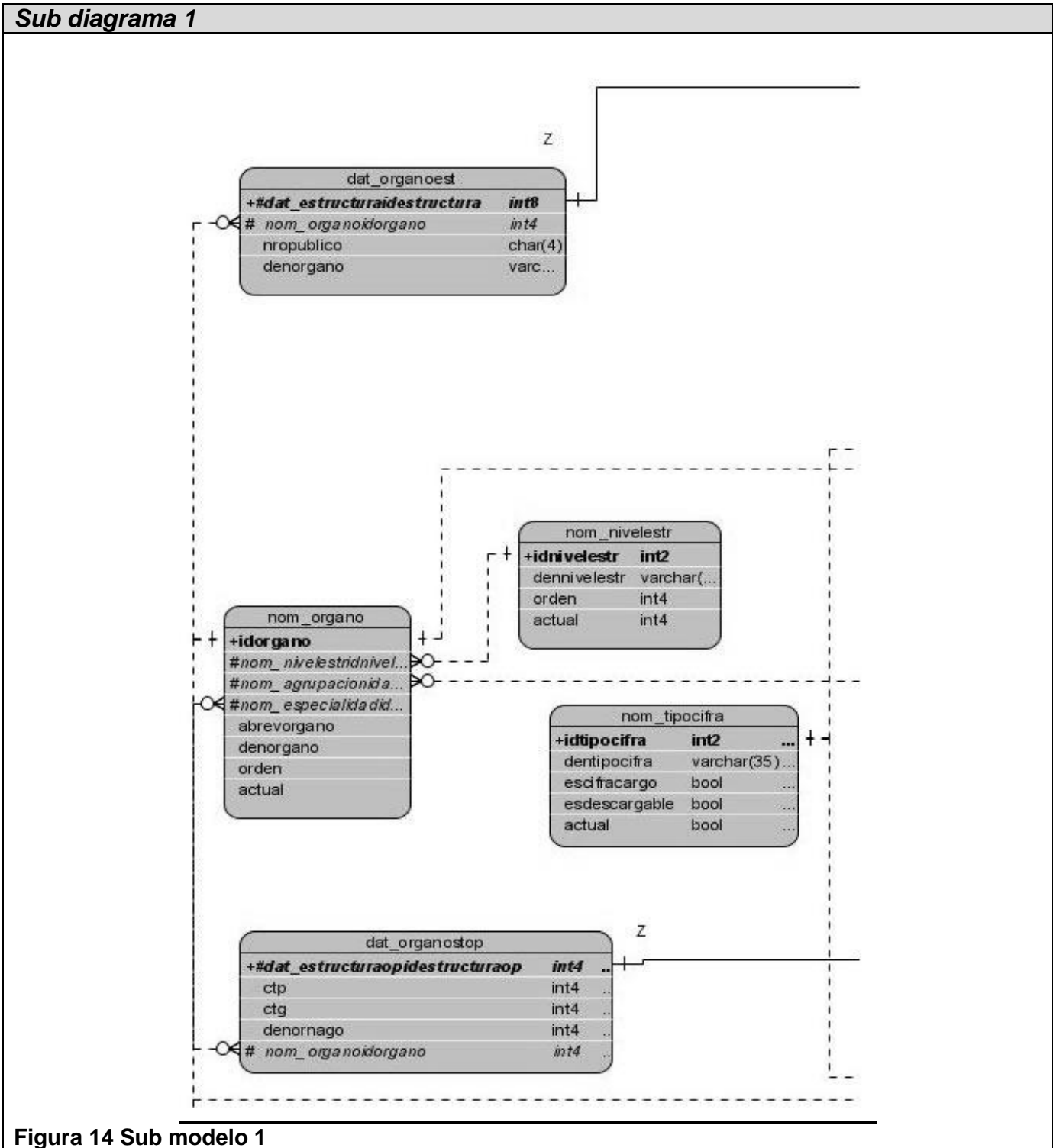
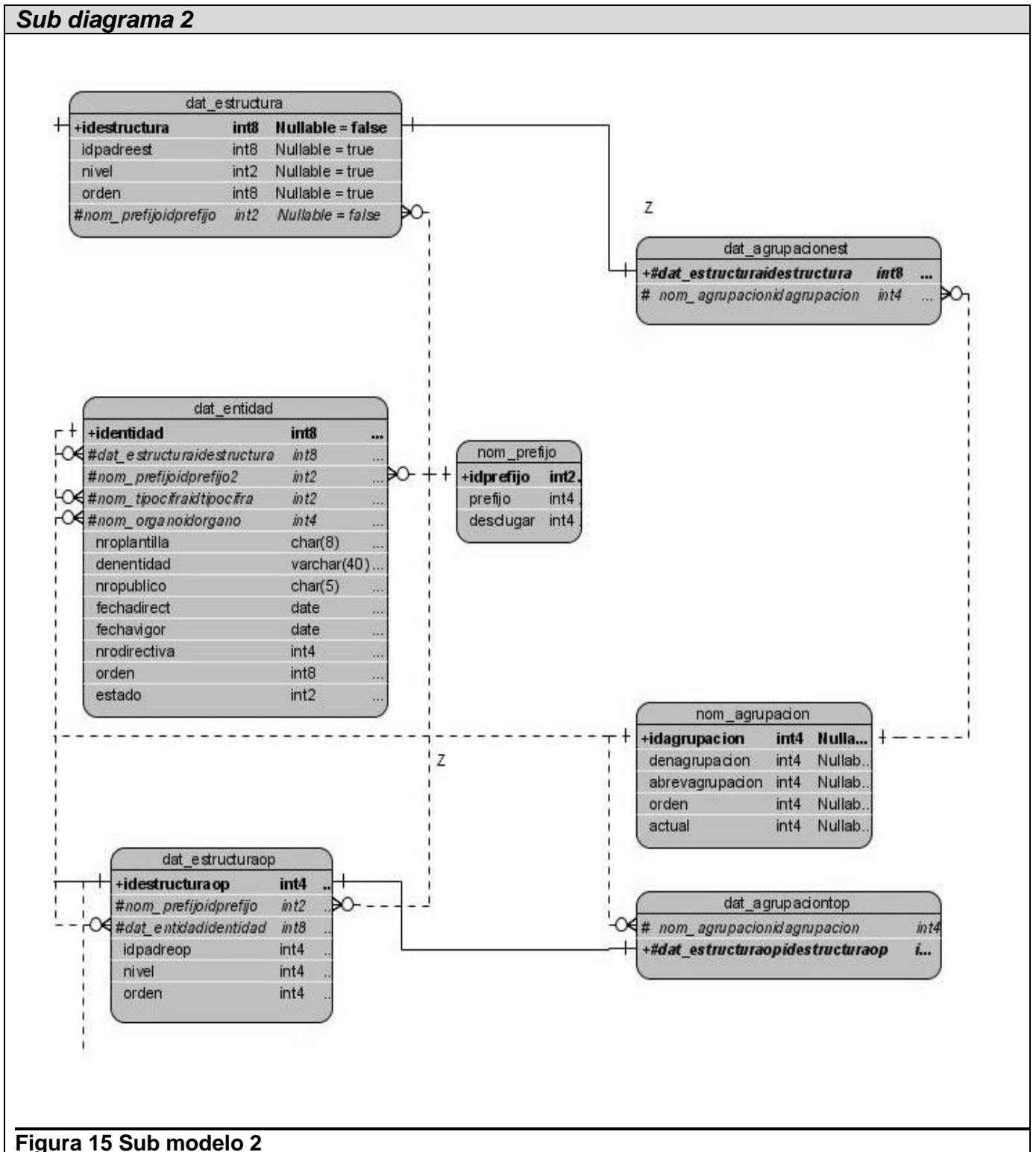


Figura 14 Sub modelo 1



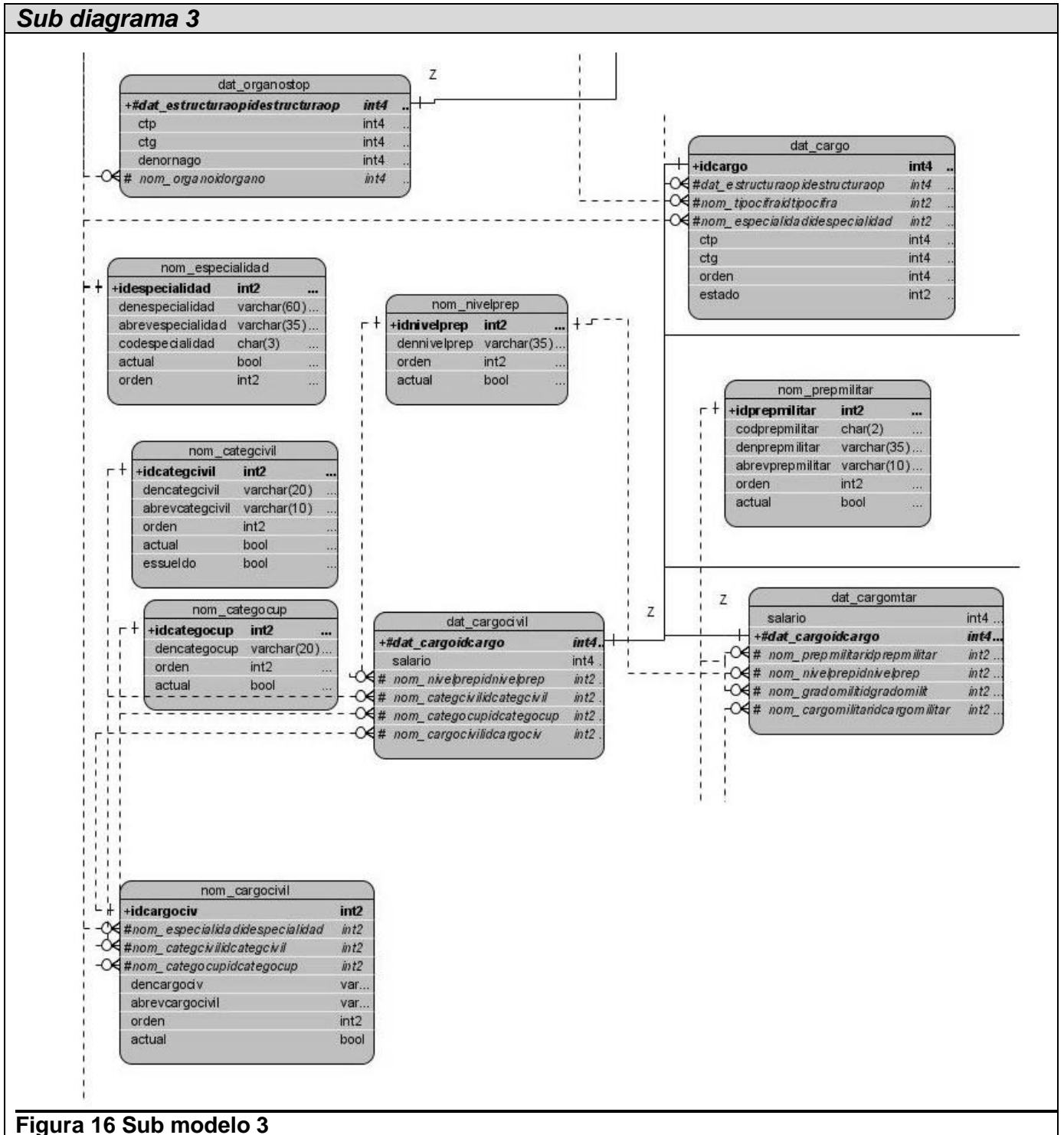


Figura 16 Sub modelo 3

Sub diagrama 4

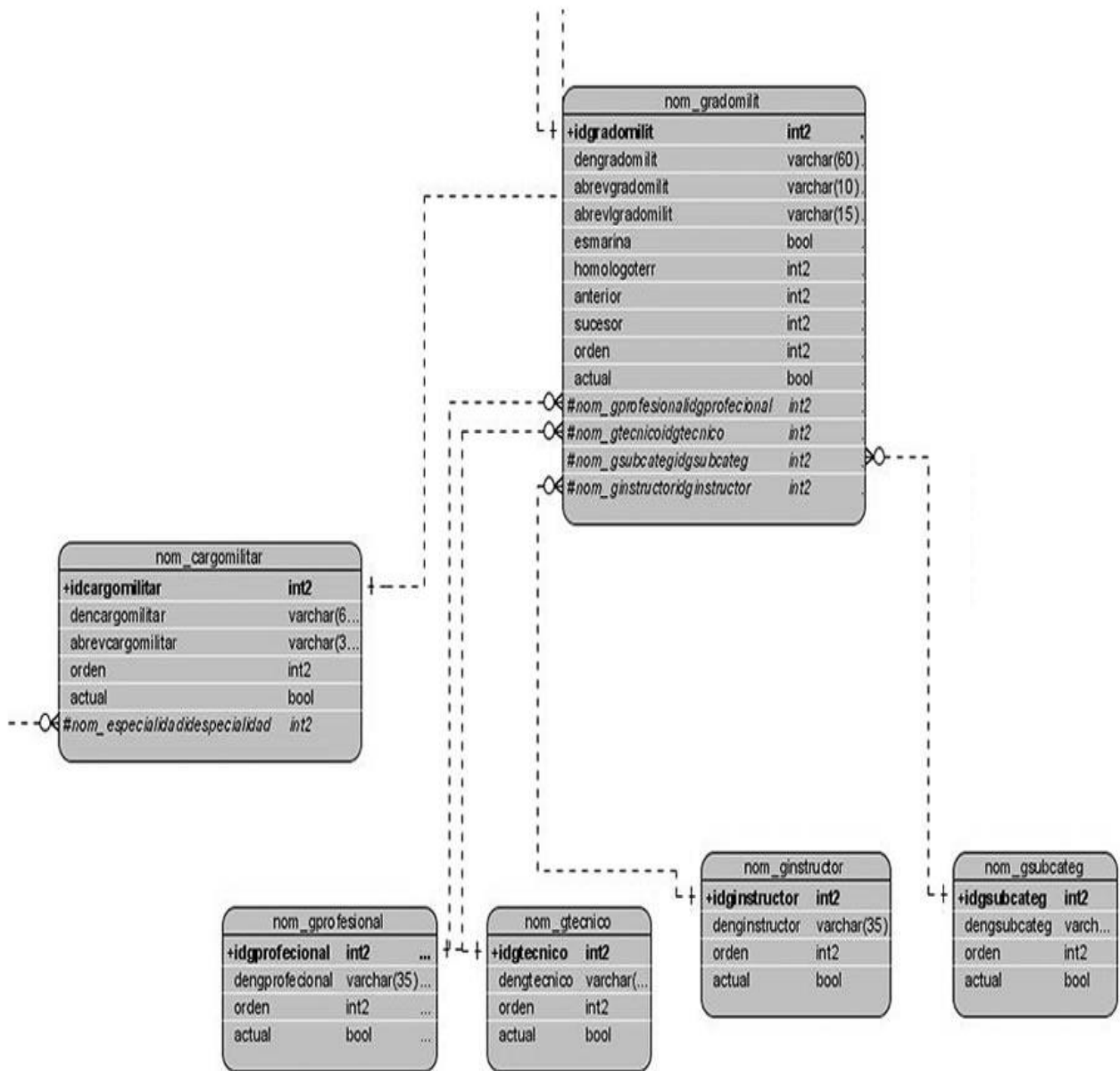


Figura 17 Sub modelo 4

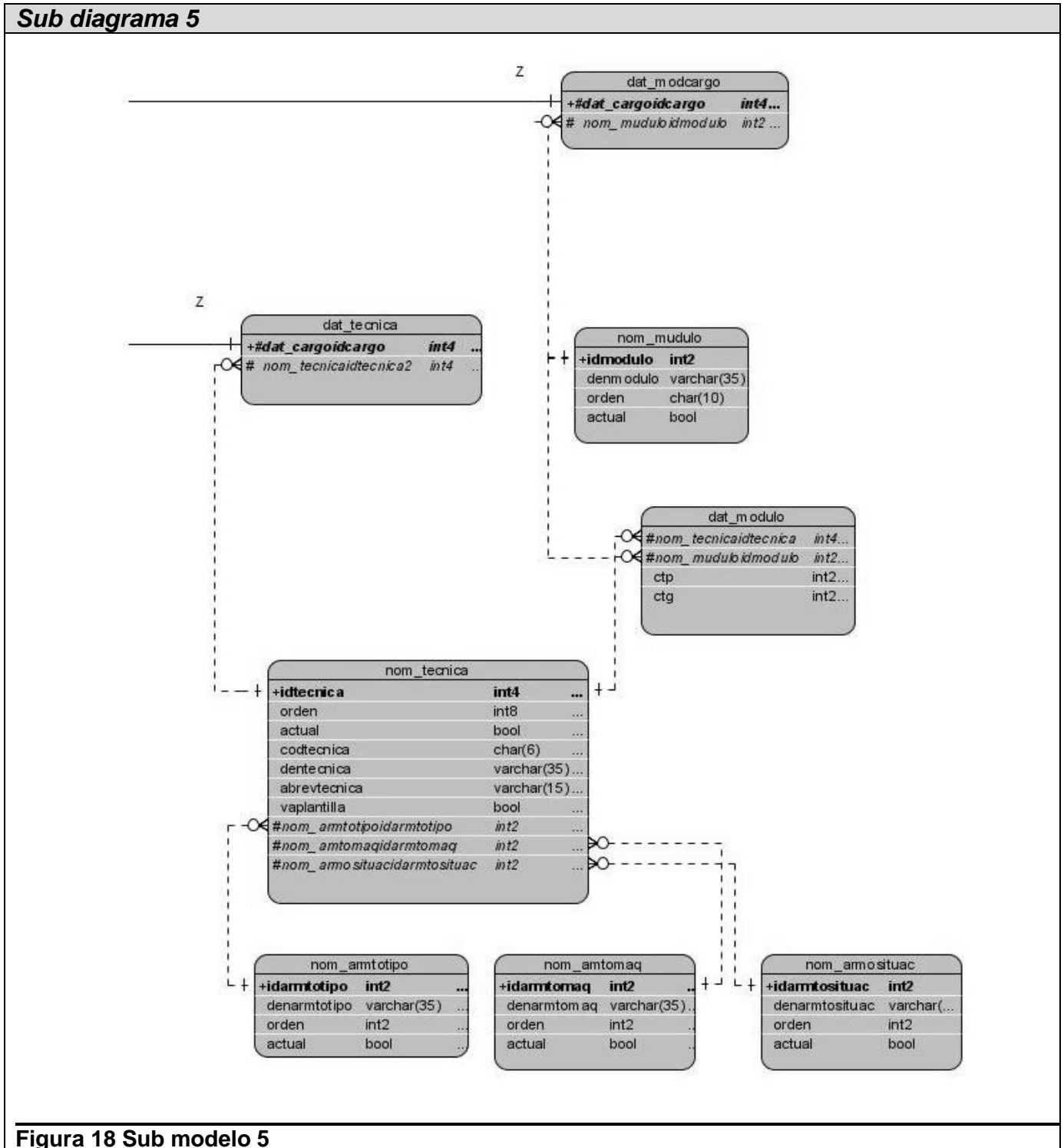


Figura 18 Sub modelo 5

2.6.1 Descripción de las tablas

Tabla 9 dat_agrupacionest

Nombre: dat_agrupacionest		
Descripción: Agrupaciones que aparecen por encima de las entidades.		
Atributo	Tipo	Descripción
Idagrupacion	SMALLINT	Tipo de agrupación.
Idagrupacion	SMALLINT	Tipo de agrupación.

Tabla 10 dat_agrupacionestop

Nombre: dat_agrupacionestop		
Descripción: Agrupaciones que aparecen por debajo de las entidades.		
Atributo	Tipo	Descripción
Idestructuraop	BIGINT	Identificador de la agrupación según la entidad dat_estructuraop.
Idagrupacion	SMALLINT	Tipo de agrupación.

Tabla 11 dat_cargo

Nombre: dat_cargo		
Descripción: Almacena los datos que son comunes para todos los cargos que forman la plantilla. (Recursos humanos y materiales).		
Atributo	Tipo	Descripción
Idcargo	BIGINT	Identificador del cargo.
Idestructuraop	BIGINT	Identificador de la estructura, a la cual pertenece el cargo.
Idespecialidad	SMALLINT	Especialidad a la que pertenece el cargo.
Ctp	SMALLINT	Cantidad de tiempo de paz.
Ctg	SMALLINT	Cantidad de tiempo de guerra.
Idtipocifra	SMALLINT	Identificador del tipo de cifra.
Idprefijo	SMALLINT	El prefijo se utiliza para definir el lugar donde se genera la información.
Orden	BIGINT	Orden en que aparece el cargo en la plantilla.
Estado	SMALLINT	Especifica el estado del cargo en la plantilla. (En elaboración, pendiente a aprobar, ..., eliminado)

Tabla 12 dat_estructura

Nombre: dat_estructura		
Descripción: Estructura de árbol para los órganos y agrupaciones por encima de las entidades.		
Atributo	Tipo	Descripción
idestructura	BIGINT	Identificador de la estructura.
idpadreest	BIGINT	Identificador del nodo padre en el árbol.
nivel	SMALLINT	Nivel de profundidad del nodo en el árbol.
orden	BIGINT	Orden del nodo. El atributo "orden" no se repite en la estructura y toma su valor teniendo en cuenta el nivel y posición de los nodos, de arriba hacia abajo y de izquierda a derecha
idprefijo	SMALLINT	Prefijo del lugar donde se elabora la plantilla.

Tabla 13 dat_entidad

Nombre: dat_entidad		
Descripción: Entidades. Las entidades son aquellos órganos para los cuales se crea una plantilla.		
Atributo	Tipo	Descripción
Identidad	BIGINT	Identificador de la entidad.
Iestructura	BIGINT	Identificador de la estructura, a la que pertenece la entidad.
Idorgano	SMALLINT	Identificador del órgano.
Nroplantilla	CHAR(8)	Número de la plantilla.
Nropublico	CHAR(5)	Número público de la unidad militar.
Denentidad	VARCHAR(40)	Denominación de la entidad.
Fechadirectiva	DATE	Fecha de la directiva que pone en vigor la plantilla.
Fechavigor	DATE	Fecha en que se pone en vigor la plantilla.
Nrodirectiva	SMALLINT	Número de la directiva que pone en vigor la plantilla.
Idtipocifra	SMALLINT	Identificador del tipo de cifra.
Idprefijo	SMALLINT	Identificador del lugar que elabora la plantilla.
Orden	BIGINT	Orden en que aparece la entidad.
Estado	SMALLINT	Especifica el estado de la entidad. (En elaboración, pendiente a aprobar,, eliminada)

Tabla 14 dat_estructuraop

Nombre: dat_estructuraop		
Descripción: Estructura de árbol para los órganos y agrupaciones por debajo de las entidades.		
Atributo	Tipo	Descripción
Idestructuraop	BIGINT	Identificador de la estructura.
Identidad	BIGINT	Identificador de la entidad a la que pertenece la estructura.
Idpadreestop	BIGINT	Identificador del nodo padre en el árbol.
Nivel	SMALLINT	Nivel de profundidad del nodo en el árbol.
Orden	BIGINT	Orden del nodo. El atributo "orden" no se repite en la estructura y toma su valor teniendo en cuenta el nivel y posición de los nodos, de arriba hacia abajo y de izquierda a derecha
Idprefijo	SMALLINT	Prefijo del lugar donde se elabora la plantilla.

2.7 Conclusiones

El diseño del modelo propuesto de la Base de Datos de Entidades y Cargos se hizo de acuerdo a los requisitos no funcionales. Las tablas se hicieron para que el tiempo de duración de las consultas fuera el menos posible. Las tablas fueron construidas tales que no exista redundancia en los datos aumentando con esto integridad en la información lo cual es uno de los objetivos principales de la BD.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

Durante el proceso de desarrollo de un software, los errores pueden comenzar a aparecer incluso desde el mismo momento en que fueron definidos los objetivos y estos a su vez especificados de forma errónea e imperfecta, de la misma forma en los posteriores pasos del diseño y desarrollo. A raíz de la incapacidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software debe ser acompañado de una actividad que garantice su calidad.

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

La creciente inclusión del software, como un elemento más de muchos sistemas y la importancia de los costos asociados a un fallo del mismo, han motivado la creación de pruebas más minuciosas y bien planificadas.

El presente capítulo trata de la validación teórica y funcional del diseño de la BD realizado. En la primera se abordan temas como la integridad, seguridad y normalización de la BD, así como del análisis de la redundancia de la información y la trazabilidad de las acciones. La segunda se basa en las pruebas realizadas a la BD para comprobar su funcionamiento.

3.2 Validación teórica del diseño

Un buen diseño de una Base de Datos no es suficiente para comenzar a utilizarla, también requiere de aspectos que garanticen la consistencia, integridad y seguridad de la misma. Los Sistemas Gestores de Bases de Datos deben cumplir estos objetivos para facilitar la manipulación y confidencialidad de los datos.

3.2.1 Integridad

La integridad de los datos se contempla en diferentes niveles. Las restricciones de dominio, transacciones y entidades definen las reglas para el mantenimiento de la integridad de las relaciones individuales. Las relaciones de integridad referencial aseguran que se mantienen las asociaciones necesarias entre las relaciones. Las restricciones de integridad de la Base de Datos gobiernan la BD

como un todo y las restricciones de integridad de transacciones controlan la forma en que se manipulan los datos, dentro de una o entre múltiples Bases de Datos.

3.2.2 Normalización de la Base de Datos

La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. Es una estrategia de diseño de abajo a arriba: se parte de los atributos y éstos se van agrupando en relaciones (tablas) según su afinidad. Aquí no se utilizará la normalización como una técnica de diseño de bases de datos, sino como una etapa posterior a la correspondencia entre el esquema conceptual y el esquema lógico, que elimine las dependencias entre atributos no deseadas. Las ventajas de la normalización son las siguientes:

- Evita anomalías en inserciones, modificaciones y borrados.
- Mejora la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.

Uno de los conceptos fundamentales en la normalización es el de dependencia funcional. Una dependencia funcional es una relación entre atributos de una misma relación (tabla). Si A y B son atributos de la relación, se dice que B es funcionalmente dependiente de A si cada valor de A tiene asociado un solo valor de B . A se le denomina determinante, ya que determina el valor de B . Se dice que el atributo B es completamente dependiente de A si depende funcionalmente de A y no depende de ningún subconjunto.

La dependencia funcional es una noción semántica. Si hay o no dependencias funcionales entre atributos no lo determina una serie abstracta de reglas, sino, más bien, los modelos mentales del usuario y las reglas de negocio de la organización o empresa para la que se desarrolla el sistema de información. Cada dependencia funcional es una clase especial de regla de integridad y representa una relación de uno a muchos.

En el proceso de normalización se debe ir comprobando que cada relación cumple una serie de reglas que se basan en la clave primaria y las dependencias funcionales. Cada regla que se cumple aumenta

el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que sí la cumplan.

La normalización se lleva a cabo en una serie de pasos. Cada paso corresponde a una forma normal que tiene unas propiedades. Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto y, por lo tanto, son menos vulnerables a las anomalías de actualización. El modelo relacional sólo requiere un conjunto de relaciones en primera forma normal. Las restantes formas normales son opcionales. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal.

Primera forma normal (1FN)

Una relación está en primera forma normal si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, no hay grupos repetitivos. Si se ve la relación gráficamente como una tabla, estará en 1FN si tiene un solo valor en la intersección de cada fila con cada columna.

Si una relación no está en 1FN, hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación. Hay dos formas de eliminar los grupos repetitivos. En la primera, se repiten los atributos con un solo valor para cada valor del grupo repetitivo. De este modo, se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarán después mediante las restantes formas normales. La segunda forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos en una relación aparte, heredando la clave primaria de la relación en la que se encontraban.

Un conjunto de relaciones se encuentra en 1FN si ninguna de ellas tiene grupos repetitivos.

Segunda forma normal (2FN)

Una relación está en segunda forma normal si, y sólo si, está en 1FN y, además, cada atributo que no está en la clave primaria es completamente dependiente de la clave primaria.

La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos. Si una relación está en 1FN y su clave primaria es simple entonces también está en 2FN. Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

Para pasar una relación en 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva relación con una copia de su determinante.

Tercera forma normal (3FN)

Una relación está en tercera forma normal si, y sólo si, está en 2FN y, además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. La dependencia es transitiva si existen las dependencias siendo, atributos o conjuntos de atributos de una misma relación.

Aunque las relaciones en 2FN tienen menos redundancias que las relaciones en 1FN, todavía pueden sufrir anomalías frente a las actualizaciones. Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de su determinante (el atributo o atributos no clave de los que dependen). (7)

3.2.3 Análisis de redundancia de información

Para que una base de datos sea efectiva hace falta eliminar en la medida de lo posible las redundancias, es decir, las repeticiones que puedan llevar a error, ya que si no existe el riesgo de inconsistencia entre las distintas versiones de los mismos datos, además para que la BD no aumente su volumen con datos ya existentes. Para que la redundancia de información no estuviese presente, esta BD paso por un proceso de normalización, logrando con esto que cada tabla este con los atributos que necesitan.

3.2.4 Análisis de la seguridad de la Base de Datos

La seguridad garantiza el acceso autorizado a los datos, para irrumpir cualquier intento de acceso no autorizado, ya sea por error del usuario o por mala intención.

La información debe ser recuperada en caso de que ocurra algún error en el servidor de la Base de Datos o alguna falla de energía. Deben realizarse copias de seguridad de las Bases de Datos regularmente. Dado que PostgreSQL gestiona sus propios ficheros en el sistema, no se recomienda confiar en los sistemas de copia de seguridad del sistema para las copias de respaldo de las BD; no hay garantía de que los ficheros estén en un estado consistente que permita su uso después de la restauración.

Para la realización de copias a la BD PostgreSQL cuenta con un comando denominado `pg_dump` que se encarga de hacerle un backup a la BD en el momento que el administrador decida, luego la información se recupera por medio de otro comando: `pg_restore`. Además, este SGBD se puede ejecutar en dos modos: `fsync` y `no fsync`. Si se ejecuta en el primer modo el gestor sería más lento, pero garantizaría que si el sistema operativo se bloquea o se produce una pérdida de energía, todos sus datos estarían almacenados y sin daños en el disco. El modo de ejecución de PostgreSQL en el servidor de BD de la aplicación será `fsync`, para evitar pérdida de la información en caso de que ocurra algún error inevitable en el sistema. Las contraseñas de los usuarios, tanto los de la aplicación como el administrador de la BD estarán encriptadas, para evitar que intrusos se apoderen de ellas con malos fines.

3.3 Validación funcional

Para garantizar que la Base de Datos cumpla con los requisitos de los usuarios y opere sin problemas se necesitan buenos procedimientos de pruebas. Las pruebas de validación normalmente se centran en las operaciones siguientes:

- La carga de la Base de Datos se realiza sin violar la integridad de los datos.
- La correcta interfaz de las aplicaciones con la Base de Datos.
- El rendimiento del sistema satisface las necesidades para las que se adquirió el SGBD.

El diseño de la Base de Datos de Entidades y Cargos incluye en su estructura tablas que contienen información que no varían frecuentemente, y que se encuentran en formato DBF. Para la carga de estos ficheros se utilizó la funcionalidad para importación de datos que tiene PostgreSQL por defecto, lo que permite disminuir el tiempo de llenado de las tablas y evitar equivocaciones en el momento de insertar los datos existentes en la aplicación anterior.

Para la realización de pruebas de un llenado voluminoso e inteligente de la Base de Datos, sin violar la integridad de la misma, se empleó PostgreSQL Data Generator. Esta herramienta es muy poderosa en la generación de datos de pruebas para Bases de Datos construidas en PostgreSQL. Permite seleccionar tablas y columnas para generar datos, definir rangos de valores, generar columnas de caracteres por máscara. También brinda la posibilidad de generar los datos a través de consultas SQL o de escogerlos por medio de listas de valores. Una de las ventajas que tiene es que genera los datos que provienen de otras tablas para evitar errores de integridad referencial.

Las pruebas realizadas a una BD nunca pueden tomarse como resultados reales, aunque dan un aproximado al mismo, dependiendo del grado de acercamiento que se utilice a los procesos de la vida real. Aún así, la implantación y utilización de una BD, está marcada por factores como, cantidad de usuarios a conectarse y el grado de concurrencia de las solicitudes hechas por los mismos, los cuales hay que tener en cuenta siempre. Además, es necesario velar el cumplimiento de todos los requerimientos no funcionales propuestos para la BD, ya que validan el funcionamiento correcto y máximo rendimiento de la misma. Aunque las pruebas realizadas brinden resultados que puedan parecer alentadores, no pueden ser motivo de confianza, todo lo contrario, la mejor prueba que se le puede realizar a cualquier resultado informático lo constituye la interacción directa del usuario, de aquí los procesos de mantenimiento de software y la realización de versiones del mismo buscando mejorar o resolver, posibles errores que se detecten durante su vida útil.

3.4 Conclusiones

Con la validación teórica del diseño realizado se chequearon parámetros muy importantes a tener en cuenta en el correcto funcionamiento de la Base de Datos. Se definieron restricciones de integridad para garantizar la validez de los datos introducidos o modificados en la misma. Fueron tomados en cuenta aspectos para garantizar la seguridad de los datos en el sistema y llevar a cabo la trazabilidad de las acciones en la BD. Por otra parte, las pruebas realizadas a la BD permitieron chequear la integridad y seguridad de los datos y el correcto funcionamiento de la misma.

CONCLUSIONES

La información ha sido siempre un recurso muy valorado. Desde las antiguas civilizaciones se ha tratado de manipular y guardar toda la información posible, puesto que esta información podría servir en algún momento para tomar decisiones. Las bases de datos son el soporte fundamental de la cualquier organización para esto.

El modelo de datos informático que se realizó da solución a la problemática que lo originó, y con esto traerá un uso más eficiente de la información en lo que respecta a entidades y cargos. Además se cumplió con el objetivo general el cual era crear un modelo de datos informático que permita la gestión de las estructuras de los organismos e instituciones de las FAR.

Se llegó a la conclusión de que la BD de Entidades y Cargos cumple con el proceso de normalización hasta la tercera forma normal, posibilitando que en las tablas se encuentren datos que realmente pertenecen a ellas y que se disminuya la redundancia en la información.

RECOMENDACIONES

- Con este trabajo se exhorta a una profundización del tema abordado para así poder dar la posibilidad de detectar debilidades.

BIBLIOGRAFÍA

1. <http://www.desarrolloweb.com>. [En línea] [Citado el: 09 de 02 de 2008.]
<http://www.desarrolloweb.com/articulos/modelos-base-datos.html>.
2. <http://www.monografias.com>. [En línea] [Citado el: 10 de 02 de 2008.]
<http://www.monografias.com/trabajos5/tipbases/tipbases.shtml#bases23>.
3. <http://teleformacion.uci.cu>. [En línea] [Citado el: 09 de 02 de 2008.]
<http://teleformacion.uci.cu/mod/resource/view.php?id=13590..>
4. <http://www.monografias.com/>. [En línea] [Citado el: 10 de 02 de 2008.]
<http://www.monografias.com/trabajos11/basda/basda.shtml#bas>.
5. **C.J.Date**. *Introducción a los Sistemas de Base de Datos*. La Habana : s.n., 2003. Vol. Primera Parte.
6. F., A. R. M. Y. M. Propuesta de arquitectura para un sistema de información hospitalaria. La Habana : UCI, 2007.
7. <http://www.mailxmail.com/>. [En línea] [Citado el: 20 de 02 de 08.]
<http://www.mailxmail.com/curso/informatica/disenobasesdatosrelacionales/capitulo8.htm>.

CITAS BIBLIOGRÁFICAS

1. <http://www.desarrolloweb.com>. [En línea] [Citado el: 09 de 02 de 2008.]
<http://www.desarrolloweb.com/articulos/modelos-base-datos.html>.
2. <http://www.monografias.com>. [En línea] [Citado el: 10 de 02 de 2008.]
<http://www.monografias.com/trabajos5/tipbases/tipbases.shtml#bases23>.
3. <http://teleformacion.uci.cu>. [En línea] [Citado el: 09 de 02 de 2008.]
<http://teleformacion.uci.cu/mod/resource/view.php?id=13590..>
4. <http://www.monografias.com/>. [En línea] [Citado el: 10 de 02 de 2008.]
<http://www.monografias.com/trabajos11/basda/basda.shtml#bas>.
5. **C.J.Date.** *Introducción a los Sistemas de Base de Datos*. La Habana : s.n., 2003. Vol. Primera Parte.
6. F., A. R. M. Y. M. *Propuesta de arquitectura para un sistema de información hospitalaria*. La Habana : UCI, 2007.
7. <http://www.mailxmail.com/>. [En línea] [Citado el: 20 de 02 de 08.]
<http://www.mailxmail.com/curso/informatica/disenobasesdatosrelacionales/capitulo8.htm>.

ANEXOS

1 Anexo 1. Descripción de las clases del diagrama de clases persistentes.**Tabla 15 Cargocivil**

Nombre: dat_cargocivil	
Tipo de Clase: Entidad	
Atributo	Tipo
Idcargo	BIGINT
Idcargociv	SMALLINT
Idnivelprep	SMALLINT
Idcategocivil	SMALLINT
Idcategocup	SMALLINT
Salario	MONEY(10, 0)

Tabla 16 Cargomilitar

Nombre: dat_cargomtar	
Tipo de Clase: Entidad	
Atributo	Tipo
Idcargo	BIGINT
Idcargomilitar	SMALLINT
Idnivelprep	SMALLINT
Idgradomilit	SMALLINT
Idprepmilitar	SMALLINT
Salario	MONEY(10, 0)

Tabla 17 Modulos

Nombre: dat_modulos	
Tipo de Clase: Entidad	
Atributo	Tipo
idtecnica	BIGINT
idmodulo	SMALLINT
ctp	SMALLINT
ctg	SMALLINT

Tabla 18 tecnica

Nombre: dat_tecnica	
Tipo de Clase: Entidad	
Atributo	Tipo
idcargo	BIGINT
idtecnica	BIGINT

Tabla 19 nom_agrupacion

Nombre: nom_agrupacion	
Tipo de Clase: Entidad	
Atributo	Tipo
idagrupacion	SMALLINT
denagrupacion	VARCHAR(60)
abrevagrupacion	VARCHAR(15)
orden	SMALLINT
actual	BIT

Tabla 20 Amtomaq

Nombre: nom_amtomaq	
Tipo de Clase: Entidad	
Atributo	Tipo
idarmtomaq	SMALLINT
denamtomaq	VARCHAR(35)
orden	SMALLINT
actual	BIT

Tabla 21 Armtosituac

Nombre: nom_armtsituac	
Tipo de Clase: Entidad	
Atributo	Tipo
idarmtsituac	SMALLINT
denarmtsituac	VARCHAR(35)
orden	SMALLINT
actual	BIT

Tabla 22 Armtotipo

Nombre: nom_armtotipo	
Tipo de Clase: Entidad	
Atributo	Tipo
idarmtotipo	SMALLINT
denarmtotipo	VARCHAR(35)
orden	SMALLINT
actual	BIT

Tabla 23 Gradomilitar

Nombre: nom_gradomilit	
Tipo de Clase: Entidad	
Atributo	Tipo
idgradomilit	SMALLINT
dengradomilit	VARCHAR(60)
abrevgradomilit	VARCHAR(10)
abrevlgradomilit	VARCHAR(15)
idgsubcateg	SMALLINT
esmarina	BIT
homologoterr	SMALLINT
anterior	SMALLINT
sucesor	SMALLINT
idgprofecional	SMALLINT
idgtecnico	SMALLINT
idginstructor	SMALLINT
orden	SMALLINT
actual	BIT

Tabla 24 Gtecnico

Nombre: nom_gtecnico	
Tipo de Clase: Entidad	
Atributo	Tipo
idgtecnico	SMALLINT
dengtecnico	VARCHAR(35)
orden	SMALLINT
actual	BIT

Tabla 25 Cargocivil

Nombre: nom_cargocivil	
Tipo de Clase: Entidad	
Atributo	Tipo
idcargociv	SMALLINT
dencargociv	VARCHAR(60)
abrevcargocivil	VARCHAR(15)
idespecialidad	SMALLINT
idcategocup	SMALLINT
idcategocivil	SMALLINT
orden	SMALLINT
actual	BIT

Tabla 26 Cargomilitar

Nombre: nom_cargomilitar	
Tipo de Clase: Entidad	
Atributo	Tipo
idcargomilitar	SMALLINT
dencargomilitar	VARCHAR(60)
abrevcargomilitar	VARCHAR(35)
idespecialidad	SMALLINT
orden	SMALLINT
actual	BIT

Tabla 27 CategoriaCivil

Nombre: nom_categcivil	
Tipo de Clase: Entidad	
Atributo	Tipo
idcategcivil	SMALLINT
dencategcivil	VARCHAR(20)
abrevcategcivil	VARCHAR(10)
essueldo	BIT
orden	SMALLINT
actual	BIT

Tabla 28 Especialidad

Nombre: nom_especialidad	
Tipo de Clase: Entidad	
Atributo	Tipo
idespecialidad	SMALLINT
denespecialidad	VARCHAR(60)
abrevespecialidad	VARCHAR(35)
codespecialidad	CHAR(3)
orden	SMALLINT
actual	BIT

Tabla 29 Ginstructor

Nombre: nom_ginstructor	
Tipo de Clase: Entidad	
Atributo	Tipo
idginstructor	SMALLINT
denginstructor	VARCHAR(35)
orden	SMALLINT
actual	BIT

Tabla 30 Gprofecional

Nombre: nom_gprofecional	
Tipo de Clase: Entidad	
Atributo	Tipo
idgprofecional	SMALLINT
dengprofecional	VARCHAR(35)
orden	SMALLINT
actual	BIT

Tabla 31 Modcargo

Nombre: dat_modcargo	
Tipo de Clase: Entidad	
Atributo	Tipo
idcargo	BIGINT
idmodulo	SMALLINT

Tabla 32 Gsubcategoria

Nombre: nom_gsubcateg	
Tipo de Clase: Entidad	
Atributo	Tipo
idgsubcateg	SMALLINT
densubcateg	VARCHAR(35)
orden	SMALLINT
actual	BIT

Tabla 33 Modulo

Nombre: nom_modulo	
Tipo de Clase:	
Atributo	Tipo
idmodulo	SMALLINT
denmodulo	VARCHAR(35)
orden	CHAR(10)
actual	BIT

Tabla 34 Nivelestr

Nombre: nom_nivelestr	
Tipo de Clase: Entidad	
Atributo	Tipo
idnivelestr	SMALLINT
dennivelestr	VARCHAR(35)
orden	SMALLINT
actual	BIT

Tabla 35 Prefijo

Nombre: nom_prefijo	
Tipo de Clase: Entidad	
Atributo	Tipo
idprefijo	SMALLINT
prefijo	CHAR(7)
desclugar	VARCHAR(30)

Tabla 36 Nivelpreparacion

Nombre: nom_nivelprep	
Tipo de Clase: Entidad	
Atributo	Tipo
idnivelprep	SMALLINT
dennivelprep	VARCHAR(35)
orden	SMALLINT
actual	BIT

Tabla 37 Organo

Nombre: nom_organo	
Tipo de Clase: Entidad	
Atributo	Tipo
idorgano	SMALLINT
denorgano	VARCHAR(60)
abrevorgano	VARCHAR(15)
idnivelestr	SMALLINT
idespecialidad	SMALLINT
idagrupacion	SMALLINT
orden	SMALLINT
actual	BIT

Tabla 38 Prepmilitar

Nombre: nom_prepmilitar	
Tipo de Clase: Entidad	
Atributo	Tipo
idprepmilitar	SMALLINT
codprepmilitar	CHAR(2)
denprepmilitar	VARCHAR(35)
abrevprepmilitar	VARCHAR(10)
orden	SMALLINT
actual	BIT

Tabla 39 Tipocifra

Nombre: nom_tipocifra	
Tipo de Clase: Entidad	
Atributo	Tipo
idtipocifra	SMALLINT
dentipocifra	VARCHAR(35)
escifracargo	BIT
esdescargable	BIT
actual	BIT

Tabla 40 Tecnica

Nombre: nom_tecnica	
Tipo de Clase: Entidad	
Atributo	Tipo
idtecnica	BIGINT
codtecnica	CHAR(6)
dentecnica	VARCHAR(35)
abrevtecnica	VARCHAR(15)
idarmtosituac	SMALLINT
idarmtotipo	SMALLINT
idarmtomaq	SMALLINT
vaplantilla	BIT
orden	BIGINT
actual	BIT

Tabla 41 Categoriacup

Nombre: nom_categocup	
Tipo de Clase: Entidad	
Atributo	Tipo
idcategocup	SMALLINT
dencatocup	VARCHAR(20)
orden	SMALLINT
actual	BIT

2 Anexo 2. Descripción de las tablas del diagrama entidad relación.

Tabla 42 dat_cargocivil

Nombre: dat_cargocivil		
Descripción: Información relativa a los cargos civiles		
Atributo	Tipo	Descripción
Idcargo	BIGINT	Identificador del cargo según la entidad dat_cargo.
Idcargociv	SMALLINT	Cargo según el nomenclador nom_cargocivil.
Idnivelprep	SMALLINT	Nivel de preparación requerido para ocupar el cargo.
Idcategcivil	SMALLINT	Identificador de categoría civil del cargo. Según entidad nom_categcivil (Directivos, Ejecutivos, etc.)
Idcategocup	SMALLINT	Identificador de categoría ocupacional del cargo. Según entidad nom_categocup (Administrativo, Obrero, etc.)
Salario	MONEY(10, 0)	Salario por el cargo.

Tabla 43 dat_cargomtar

Nombre: dat_cargomtar		
Descripción: Información relativa a los cargos militares.		
Atributo	Tipo	Descripción
Idcargo	BIGINT	Identificador del cargo según la entidad dat_cargo.
Idcargomilitar	SMALLINT	Cargo según el nomenclador nom_cargomilitar.
Idnivelprep	SMALLINT	Nivel de preparación requerido para ocupar el cargo.
Idgradomilit	SMALLINT	Grado militar por plantilla.
Idprepmilitar	SMALLINT	Preparación militar requerida para el cargo.
Salario	MONEY(10, 0)	Salario por cargo.

Tabla 44 dat_modcargo

Nombre: dat_modcargo		
Descripción: Información relativa al material de guerra (armamento, carros...). Se especifica mediante módulos. Por módulos es cuando se define la cantidad para un grupo de medios agrupados en un módulo, ejemplo-Modulo personal del soldado. En este caso el modulo personal del soldado esta formado por: Fusil Automático, Mochila, municiones.		
Atributo	Tipo	Descripción
idcargo	BIGINT	Identificador del cargo según la entidad dat_cargo.
idmodulo	SMALLINT	Identificador del módulo según entidad nom_modulo.

Tabla 45 dat_modulos

Nombre: dat_modulos		
Descripción: Entidad para la formación de los módulos.		
Atributo	Tipo	Descripción
idtecnica	BIGINT	Identificador del material de guerra (armamento o técnica). Según nom_tecnica
idmodulo	SMALLINT	Identificador del módulo según entidad nom_modulo.
ctp	SMALLINT	Cantidad de tiempo de paz.
ctg	SMALLINT	Cantidad de tiempo de guerra.

Tabla 46 dat_organoest

Nombre: dat_organoest		
Descripción: Órganos que aparecen por encima de las entidades.		
Atributo	Tipo	Descripción
idestructura	BIGINT	Identificador del órgano según la entidad dat_estructura
idorgano	SMALLINT	Tipo de órgano.
nropublico	CHAR(4)	Número público. Es igual al número público de la primera entidad.
denorgano	VARCHAR(50)	Denominación del órgano.

Tabla 47 dat_organostop

Nombre: dat_organostop		
Descripción: Órganos que aparecen por debajo de las entidades.		
Atributo	Tipo	Descripción
idestructuraop	BIGINT	Identificador del órgano según la entidad dat_estructuraop
idorgano	SMALLINT	Tipo de órgano.
denorgano	VARCHAR(50)	Denominación del órgano.
ctp	SMALLINT	Cantidad en tiempo de paz.
ctg	SMALLINT	Cantidad en tiempo de guerra.

Tabla 48 dat_tecnica

Nombre: dat_tecnica		
Descripción: Información relativa al material de guerra (armamento, carros...). Se especifica uno a uno. Uno a uno es cuando se define la plantilla por cada medio material, ejemplo- Carro de carga.		
Atributo	Tipo	Descripción
idcargo	BIGINT	Identificador del cargo según la entidad dat_cargo.
idtecnica	BIGINT	Identificador del material de guerra (armamento o técnica). Según nom_tecnica

Tabla 49 nom_agrupacion

Nombre: nom_agrupacion		
Descripción: Nomenclador de agrupaciones. (Unidades de aseguramiento, Órganos de dirección...)		
Atributo	Tipo	Descripción
idagrupacion	SMALLINT	Identificador del tipo de agrupación.
denagrupacion	VARCHAR(60)	Denominación del tipo de agrupación.
abrevagrupacion	VARCHAR(15)	Abreviatura de la agrupación.
orden	SMALLINT	Orden en que aparece la agrupación.
actual	BIT	Actual.

Tabla 50 nom_amtomaq

Nombre: nom_amtomaq		
Descripción:		
Atributo	Tipo	Descripción
idarmtomaq	SMALLINT	Identificador.
denamtomaq	VARCHAR(35)	Denominación.
orden	SMALLINT	Orden en que aparece.
actual	BIT	Actual.

Tabla 51 nom_armtosituac

Nombre: nom_armtosituac		
Descripción:		
Atributo	Tipo	Descripción
idarmtosituac	SMALLINT	Identificador.
denarmtosituac	VARCHAR(35)	Denominación.
orden	SMALLINT	Orden.
actual	BIT	Actual.

Tabla 52 nom_armtotipo

Nombre: nom_armtotipo		
Descripción: Nomenclador del tipo de armamento.		
Atributo	Tipo	Descripción
idarmtotipo	SMALLINT	Identificador del tipo de armamento.
denarmtotipo	VARCHAR(35)	Denominación del tipo de armamento.
orden	SMALLINT	Orden en que aparece el tipo de armamento.
actual	BIT	Actual.

Tabla 53 nom_cargocivil

Nombre: nom_cargocivil		
Descripción: Nomenclador de cargos civiles. (Técnico en electricidad, Especialista en informática, etc.)		
Atributo	Tipo	Descripción
idcargociv	SMALLINT	Identificador del cargo civil.
dencargociv	VARCHAR(60)	Denominación del cargo civil.
abrevcargocivil	VARCHAR(15)	Abreviatura del cargo civil.
idespecialidad	SMALLINT	Especialidad a la que pertenece el cargo civil.
idcategocup	SMALLINT	Identificador de la tabla idcategocup.
idcategocivil	SMALLINT	Identificador de la categoría civil, a la cual pertenece el cargo civil según nom_categcivil.
orden	SMALLINT	Orden del cargo civil.
actual	BIT	Actual.

Tabla 54 nom_cargomilitar

Nombre: nom_cargomilitar		
Descripción: Nomenclador de cargos militares. (Jefe de compañía, Jefe de Batería, etc.)		
Atributo	Tipo	Descripción
idcargomilitar	SMALLINT	Identificado del cargo militar.
dencargomilitar	VARCHAR(60)	Denominación del cargo militar.
abrevcargomilitar	VARCHAR(35)	Abreviatura del cargo militar.
idespecialidad	SMALLINT	Especialidad a la que pertenece el cargo militar según nom_especialidad.
orden	SMALLINT	Orden del cargo militar.
actual	BIT	Actual.

Tabla 55 nom_categocup

Nombre: nom_categocup		
Descripción: Nomenclador de categoría ocupacional. (Administrativo, Obrero, etc.)		
Atributo	Tipo	Descripción
idcategocup	SMALLINT	Identificador de la categoría ocupacional.
dencatocup	VARCHAR(20)	Denominación de la categoría ocupacional.
orden	SMALLINT	Orden de la categoría.
actual	BIT	Actual.

Tabla 56 nom_categcivil

Nombre: nom_categcivil		
Descripción: Nomenclador que identifica las Categorías Funcional de los Civiles. (Directivos, Ejecutivos, etc.)		
Atributo	Tipo	Descripción
idcategcivil	SMALLINT	Identificador de la categoría funcional de un civil.
dencategcivil	VARCHAR(20)	Denominación de la categoría civil.
abrevcategcivil	VARCHAR(10)	Abreviatura de la categoría civil.
essueldo	BIT	Sueldo.
orden	SMALLINT	Orden de la categoría civil.
actual	BIT	Actual.

Tabla 57 nom_especialidad

Nombre: nom_especialidad		
Descripción: Nomenclador de especialidad. (Comunicaciones, armamento...)		
Atributo	Tipo	Descripción
idespecialidad	SMALLINT	Identificador de la especialidad.
denespecialidad	VARCHAR(60)	Denominación de la especialidad.
abrevespecialidad	VARCHAR(35)	Abreviatura de la especialidad.
codespecialidad	CHAR(3)	Código de la especialidad.
orden	SMALLINT	Orden de la especialidad.
actual	BIT	Actual.

Tabla 58 nom_ginstructor

Nombre: nom_ginstructor		
Descripción: Nomenclador de grado instructor.		
Atributo	Tipo	Descripción
idginstructor	SMALLINT	Identificador del grado del instructor.
denginstructor	VARCHAR(35)	Denominación del grado del instructor.
orden	SMALLINT	Orden de grado instructor.
actual	BIT	Actual.

Tabla 59 nom_gprofecional

Nombre: nom_gprofecional		
Descripción: Nomenclador de grado profesional.		
Atributo	Tipo	Descripción
idgprofecional	SMALLINT	Identificador del grado profesional.
dengprofecional	VARCHAR(35)	Denominación del grado profesional.
orden	SMALLINT	Orden de grado profesional.
actual	BIT	Actual.

Tabla 60 nom_gtecnico

Nombre: nom_gtecnico		
Descripción: Nomenclador de grado técnico		
Atributo	Tipo	Descripción
idgtecnico	SMALLINT	Identificador del grado técnico.
dengtecnico	VARCHAR(35)	Denominación del grado del técnico.
orden	SMALLINT	Orden del grado técnico.
actual	BIT	Actual.

Tabla 61 nom_gradomilit

Nombre: nom_gradomilit		
Descripción: Nomenclador de grado militar. (Soldado, teniente...)		
Atributo	Tipo	Descripción
idgradomilit	SMALLINT	Identificador del grado militar.
dengradomilit	VARCHAR(60)	Denominación del grado militar.
abrevgradomilit	VARCHAR(10)	Abreviatura del grado militar.
abrevlgradomilit	VARCHAR(15)	
idgsubcateg	SMALLINT	Identificador de la sub-categoría del grado militar de la tabla nom_gsubcateg.
esmarina	BIT	Si el grado es de la marina.
homologoterr	SMALLINT	El grado homologa a este en las tropas terrestres.
anterior	SMALLINT	Grado anterior a este.
sucesor	SMALLINT	Grado superior a este.
idgprofecional	SMALLINT	Identificador del grado profesional al que pertenece el grado militar según nom_gprofecional.
idgtecnico	SMALLINT	Identificador del grado técnico al que pertenece el grado militar según nom_gtecnico.
idginstructor	SMALLINT	Identificador del grado instructor al que pertenece el grado militar según nom_ginstructor.
orden	SMALLINT	Orden del grado militar.
actual	BIT	Actual.

Tabla 62 nom_gsubcateg

Nombre: nom_gsubcateg		
Descripción: Nomenclador de sub categoría del grado militar. (Sub-oficiales, oficiales, primeros oficiales...)		
Atributo	Tipo	Descripción
idgsubcateg	SMALLINT	Identificador de la sub categoría del grado militar.
densubcateg	VARCHAR(35)	Descripción de la sub-categoría del grado militar.
orden	SMALLINT	Orden de la sub categoría.
actual	BIT	Actual.

Tabla 63 nom_modulo

Nombre: nom_modulo		
Tipo de Clase: Nomenclador de módulos.		
Atributo	Tipo	Descripción
idmodulo	SMALLINT	Identificador del módulo.
denmodulo	VARCHAR(35)	Denominación del módulo.
orden	CHAR(10)	Orden del módulo.
actual	BIT	Actual.

Tabla 64 nom_nivelestr

Nombre: nom_nivelestr		
Descripción: Nomenclador de nivel estructural. (Escuadra, Pelotón, Compañía...)		
Atributo	Tipo	Descripción
idnivelestr	SMALLINT	Identificador del nivel estructural.
dennivelestr	VARCHAR(35)	Denominación del nivel estructural.
orden	SMALLINT	Orden de nivel estructural.
actual	BIT	Actual.

Tabla 65 nom_nivelprep

Nombre: nom_nivelprep		
Descripción: Nomenclador de nivel de preparación.		
Atributo	Tipo	Descripción
idnivelprep	SMALLINT	Identificador del nivel de preparación.
dennivelprep	VARCHAR(35)	Denominación del nivel de preparación.
orden	SMALLINT	Orden del nivel de preparación.
actual	BIT	Actual.

Tabla 66 nom_organo

Nombre: nom_organo		
Descripción: Nomenclador de órganos. (Brigada de tanques, Taller Integral...)		
Atributo	Tipo	Descripción
idorgano	SMALLINT	Identificador del Órgano.
denorgano	VARCHAR(60)	Denominación del órgano.
abrevorgano	VARCHAR(15)	Abreviatura del órgano.
idnivelestr	SMALLINT	Identificador del nivel estructural según nom_nivelestr.
idespecialidad	SMALLINT	Identificador de la especialidad, a la que pertenece el órgano según nom_especialidad.
idagrupacion	SMALLINT	Identificador de la agrupación, a la que pertenece el órgano según nom_agrupacion.
orden	SMALLINT	Orden del órgano.
actual	BIT	Actual.

Tabla 67 nom_prefijo

Nombre: nom_prefijo		
Descripción: Nomenclador de prefijos. Lugares donde se generan las plantillas.		
Atributo	Tipo	Descripción
idprefijo	SMALLINT	Identificador del prefijo.
prefijo	CHAR(7)	Nombre del prefijo.
desclugar	VARCHAR(30)	Descripción del lugar.

Tabla 68 nom_prepmilitar

Nombre: nom_prepmilitar		
Descripción: Nomenclador de nivel de preparación militar.		
Atributo	Tipo	Descripción
idprepmilitar	SMALLINT	Identificador del nivel de preparación militar.
codprepmilitar	CHAR(2)	Código del nivel de preparación militar.
denprepmilitar	VARCHAR(35)	Denominación del nivel de preparación militar.
abrevprepmilitar	VARCHAR(10)	Abreviatura
orden	SMALLINT	Orden del nivel de preparación militar.
actual	BIT	Actual.

Tabla 69 nom_tecnica

Nombre: nom_tecnica		
Descripción: Nomenclador del material de guerra. (Carros de carga>5t, Careta antigás...)		
Atributo	Tipo	Descripción
idtecnica	BIGINT	Identificador del material de guerra (armamento o técnica). Según nom_tecnica
codtecnica	CHAR(6)	Código del material de guerra.
dentecnica	VARCHAR(35)	Denominación del material de guerra.
abrevtecnica	VARCHAR(15)	Abreviatura del material de guerra.
idarmtosituac	SMALLINT	
idarmtotipo	SMALLINT	Identificador del tipo del armamento según nom_armtotipo.
idarmtomaq	SMALLINT	
vaplantilla	BIT	Si va a la platilla o no.
orden	BIGINT	Orden del material de guerra.
actual	BIT	Actual.

Tabla 70 nom_tipocifra

Nombre: nom_tipocifra		
Descripción: Nomenclador de tipo de cifra. (Envergadura, provisional...)		
Atributo	Tipo	Descripción
idtipocifra	SMALLINT	Identificador del tipo de cifra.
dentipocifra	VARCHAR(35)	Denominación del tipo de cifra.
escifracargo	BIT	Si es una cifra del cargo.
esdescargable	BIT	Si es descargable.
actual	BIT	Actual.

3 Anexo 3. Requisitos no funcionales.

5. Soporte.

El Gestor de Base de Datos instalado debe soporte grandes volúmenes de datos y velocidad de procesamiento.

6. Portabilidad.

La BD debe ser compatible con los sistemas operativos Linux y Windows. Además accesible principalmente con el navegador Mozilla.

7. Hardware.

Para el servidor de Base de Datos se requiere tarjeta de red, que tenga al menos 256MB de RAM y 1 GB de disco duro; y que el procesador tenga 1.2 GHz de velocidad como mínimo.

8. Software.

Los servidores deben tener instalados el sistema operativo Windows 2000 o superior, o Linux preferentemente. La Base de Datos será implementada en el Gestor de Bases de datos PostgreSQL.

9. Seguridad.

La comunicación entre la BD y el sistema el cual la utiliza debe ser seguro al igual que la forma en que se almacenan los datos y estos viajan a través de la red.

10. Integridad.

11. La información almacenada será objeto de cuidadosa protección contra la corrupción.

GLOSARIO

Microsoft (acrónimo de Microcomputer Software)

Es una empresa de Estados Unidos, fundada por Bill Gates y Paul Allen. Dueña y productora de los sistemas operativos: Microsoft DOS y Microsoft Windows, que se utilizan en la mayoría de las computadoras del planeta.

Sistema operativo (SO)

Es un conjunto de programas destinados a permitir la comunicación del usuario con un computador y gestionar sus recursos de una forma eficaz.

Lenguaje de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente.

SQL (Structured Query Language)

Es un lenguaje de acceso a las Bases de Datos, permite especificar todas las operaciones sobre la base de datos como por ejemplo: Inserción, Borrado, Actualización. Utiliza características de álgebra y cálculo relacional permitiendo de esta forma realizar consultas a la base de datos de forma sencilla.

Disparadores (trigger)

Un disparador es una regla en la que se especifica una acción que el SGBD debe ejecutar como respuesta a la ocurrencia de un evento, siempre que se cumpla la condición especificada.

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

Las Herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Multiplataforma

Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Una plataforma es una combinación de hardware y software usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de ambos.

Tablespaces

Un tablespace es una unidad lógica de almacenamiento de datos representada físicamente por uno o más archivos de datos.

Servidores

Un servidor, en informática o computación, es el ordenador en el que se ejecuta un programa que realiza alguna tarea en beneficio de otras aplicaciones llamadas clientes.