

**COMPONENTES DE BASE DE DATOS PARA  
MOPROSOFT**

**TESIS**

Que para obtener el grado de  
**MAESTRO EN INGENIERÍA (COMPUTACIÓN)**

presenta

**Jonathan Rodríguez Hernández**

**Dr. Javier García García**

**Director de Tesis**

Universidad Nacional Autónoma de México

septiembre 2008



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## Agradecimientos

A mi esposa por su incondicional amor, paciencia y apoyo en el logro de este objetivo, además le agradezco por siempre estar conmigo en las buenas y en las malas.

A mis padres porque ellos me dieron la vida y me han guiado durante todos estos años.

A Daniela, por ser mi hermana y por su apoyo durante la realización de la maestría.

A Gabriela, por ser mi hermana y por su apoyo, pues me aliento a seguir adelante aunque hubiera adversidades.

A mi tutor, por exigirme y apoyarme durante el proceso de la maestría y de mi tesis, pues logre superar las expectativas deseadas.

A la Dra. Hanna, por considerar que este tema de tesis tiene un valor para MoPro-Soft.

Agradezco el apoyo recibido por parte del Macroproyecto: Tecnologías para la Universidad de la Información y la Computación.

A todos ellos les digo muchas gracias por estar conmigo y ayudarme a cumplir uno de mis objetivos, ahora dare mi máximo esfuerzo para ofrecer a mi país todo lo que he aprendido.

# Resumen

En el proceso de Desarrollo y Mantenimiento de Software del Modelo de Procesos para la Industria de Software (MoProSoft) se puede observar que actividades especializadas como son las referentes a la conceptualización, análisis, diseño, pruebas y carga de una base de datos, no reciben un tratamiento especial para realizar su ejecución y se consideran como un conjunto más de actividades. Consideramos que para estas actividades de alta especialización, las empresas pequeñas y medianas con niveles bajos de madurez requieren de apoyo adicional en el modelo con el objeto de mejorar los procesos donde este tipo de actividades intervienen. La presente tesis tuvo como antecedente un estudio profundo del modelo de procesos MoProSoft, el cual nos llevó a revisar otros modelos que tienen que ver con la calidad del proceso de desarrollo de software sobre todo desarrollos de software que tienen una componente importante de base de datos. A raíz de esta revisión se proponen tanto modificaciones al modelo, como una serie de prácticas y guías que apoyen a los usuarios de MoProSoft a realizar, en particular, las actividades que tienen que ver con las bases de datos durante el proceso de Desarrollo y Mantenimiento de Software.

Esta propuesta tiene como objetivo el que las empresas u organizaciones que utilicen tanto MoProSoft como las prácticas y guías propuestas, mejoren los procesos donde intervienen actividades relacionadas con las bases de datos.

Estas prácticas y guías de base de datos fueron probadas en una organización de la Universidad Nacional Autónoma de México, en la Dirección General de Servicios de Cómputo Académico en la Subdirección de Sistemas donde se llevan a cabo estos procesos durante el Desarrollo y Mantenimiento de Software.

## Abstract

In the Software Development and Maintenance process of the Process Model for the Software Industry (Modelo de Procesos para la Industria de Software-MoProSoft) specialized activities such as the conceptualization, analysis, design, test and load of a database, receive no special treatment despite of their complexity and are considered as a set of activities. We believe that these highly specialized, small and medium businesses with low levels of maturity in require additional support in the model to improve the processes where this type of activity involved. This thesis was a study of the background processes MoProSoft model, which led us to review other models that have to do with the quality of the software development process on software developments that have an important component of database. Following this review was therefore proposed change to the model, as a number of practices that support and guidelines users to MoProSoft, in particular, activities to do with databases in the process of Development and Maintenance Software. This proposal focuses on businesses and organizations that wish to improve the activities related to databases in the Software Development and Maintenance. These practices and guides database were tested in an organization of the Universidad Nacional Autónoma de México, the Dirección General de Computo Académico, Subdirección de Sistemas where these processes take place during the Software Development and Maintenance process.

# Contenido

Agradecimientos . . . . .	II
Resumen . . . . .	III
Abstract . . . . .	IV
Contenido . . . . .	IV
Lista de Figuras . . . . .	VII
1. Introducción . . . . .	1
1.1. Planteamiento del Problema . . . . .	1
1.2. Objetivos de la Tesis . . . . .	2
1.3. Trabajos relacionados . . . . .	3
1.4. Descripción de Capítulos . . . . .	7
2. Marco teórico . . . . .	9
2.1. La información . . . . .	9
2.2. Sistemas de información . . . . .	11
2.3. Componentes de un sistema de información . . . . .	13
2.4. Concepto de base de datos . . . . .	14
2.5. Tipos de usuarios propios de la base de datos . . . . .	15
2.6. Sistema manejador de base de datos . . . . .	17
2.7. Modelo de datos . . . . .	20
2.8. Una mirada a MoProSoft . . . . .	22
2.8.1. Modelo de procesos para la industria de software (MoProSoft) . . . . .	24
2.8.2. Descripción del proceso de Desarrollo y Mantenimiento de Software en MoProSoft . . . . .	25
3. Componentes de base de datos en MoProSoft . . . . .	28
3.1. Propuestas a MoProSoft para integrar las actividades para el tratamiento de las bases de datos . . . . .	31
3.2. Innovaciones y prácticas requeridas en MoProSoft . . . . .	34
3.3. Estudio de viabilidad de los componentes de base de datos para MoProSoft . . . . .	44
3.4. Guías, plantillas y listas de verificación en el proceso de Desarrollo y Mantenimiento de Software para el tratamiento de las bases de datos . . . . .	46

---

4. Conclusiones	101
4.1. Conclusiones generales . . . . .	101
4.2. Trabajos futuros . . . . .	102
Referencias	104

# Lista de Figuras

1.1.	Modelo de Calidad ISO-9126 . . . . .	4
1.2.	Dimensiones de calidad de base de datos . . . . .	8
2.1.	La información y sus cualidades . . . . .	10
2.2.	Componentes de un sistema de información . . . . .	13
2.3.	El SMBD como vínculo entre la base de datos y los niveles de gestión . . . . .	19
2.4.	Partes de un modelo de datos . . . . .	20
2.5.	Tipos de modelos de datos . . . . .	21
3.1.	Diagrama E/R de la base de datos <i>Hospital</i> . . . . .	54
3.2.	Diagrama UML de clases de la base de datos <i>Hospital</i> . . . . .	55
3.3.	Ejemplo de integridad referencial de la base de datos <i>Hospital</i> . . . . .	67
3.4.	Paso del ME/R al modelo relacional . . . . .	80
3.5.	Dominio . . . . .	80
3.6.	Entidad con atributos . . . . .	81
3.7.	Diagrama Entidad/Relación . . . . .	82
3.8.	Modelo Relacional . . . . .	82



# Capítulo 1

## Introducción

### 1.1. Planteamiento del Problema

El interés de ayudar a las pequeñas y medianas empresas durante el proceso de Desarrollo y Mantenimiento de Software quedó evidenciado en [Oktaba06] al descubrirse que el nivel de capacidad al adoptar MoProSoft era menor en la categoría de operación (OPE) donde se encuentra el proceso de desarrollo y mantenimiento de software.

En el presente trabajo se proponen diferentes componentes de base de datos que se deben realizar durante el proceso de desarrollo y mantenimiento de software utilizando como base el estudio realizado a MoProSoft. Se considera importante estudiar las prácticas relacionadas con las bases de datos y proponer guías a utilizarse en el proceso de Desarrollo y Mantenimiento de Software.

La importancia de lo anterior se fortalece por la razón de que una base de datos es un producto de software que se puede analizar, diseñar, construir y probar de manera independiente de cualquier aplicación que la utilice, además por ser un recurso muy importante para cualquier empresa u organización ya que contiene el conocimiento de la misma y sus datos [Piattini07].

Aunado a lo anterior, las actividades relacionadas con las bases de datos en el proceso de desarrollo y mantenimiento de software, necesitan de ciertos roles que tengan una especialización específica.

Por lo tanto, durante el desarrollo y mantenimiento de la base de datos deben

considerarse principios de calidad singulares que tienen que ver tanto con el proceso mismo de su construcción, como con los datos almacenados en ellas y con las aplicaciones que las explotan. Es por esto que el prestar especial atención en las actividades que intervienen en el proceso de desarrollo y mantenimiento de software relacionadas con las base de datos, consideramos que conducirá a la mejora de este proceso.

## 1.2. **Objetivos de la Tesis**

### **Objetivo general**

Integrar las prácticas y guías de base de datos dentro de la estructura de MoProSoft, definiendo la forma en cómo se deben concebir de manera correcta durante el desarrollo y mantenimiento de un sistema. Lo anterior se encamina a lograr que la base de datos cumpla con los estándares de calidad que se propongan.

### **Objetivos particulares**

- Realizar los ajustes necesarios en MoProSoft para las prácticas y guías de base de datos.
- Integrar las prácticas y guías de base de datos dentro del proceso de Desarrollo y Mantenimiento de Software siguiendo la estructura del modelo para facilitar su uso.
- Diseñar las guías y plantillas necesarias para el levantamiento de requisitos, el análisis y diseño, las pruebas de la base de datos y las pruebas de integración de la base de datos.
- Definir los pasos a seguir para el desarrollo de la base de datos.
- Presentar las diferentes prácticas y guías que se integran en el proceso de Desarrollo y Mantenimiento de Software propias del tratamiento de bases de datos.
- En MoProSoft se enfatizarán estas prácticas y guías de base de datos incluidas en el proceso de Desarrollo y Mantenimiento de Software.

### 1.3. Trabajos relacionados

Los componentes de los diferentes productos en los sistemas de información en relación con el tratamiento de bases de datos deben considerar diversas dimensiones de calidad de los datos. Diferentes estudios se han realizado para medir la calidad de la base de datos como un producto de software.

El modelo de calidad de software [ISO-912601] establece la capacidad de satisfacer al cliente definiendo seis características para calidad externa e interna de un producto de software, como son:

1. Funcionalidad: capacidad de un producto de software de manejar en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado.
2. Fiabilidad: capacidad del producto de software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.
3. Usabilidad: capacidad del producto de software de evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.
4. Eficiencia: Capacidad del producto de software de evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados.
5. Mantenibilidad: Capacidad del producto de software para medir el esfuerzo necesario al realizar modificaciones, ya sea por la corrección de errores o por el incremento de funcionalidad.
6. Portatilidad: Capacidad del producto de software de ser transferido de un ambiente a otro.

Con base en lo anterior, cada una de las características del modelo de calidad ISO-9126 cuenta con atributos que permiten medir la calidad del producto de software, la Figura 1.1 muestra la relación de estos atributos con cada una de las características del estándar.

Por tal motivo, el modelo de calidad ISO-9126 define una guía de evaluación de la calidad del producto de software. Asimismo, es necesario que la empresa u organización que



Figura 1.1: Modelo de Calidad ISO-9126

se dedica al desarrollo de software establezca su modelo de calidad para valorar el nivel de excelencia de sus productos.

Nuestro trabajo enriquece MoProSoft y además define también la calidad externa, interna y de uso de la base de datos lo que permitirá entender de una manera más clara el proceso de desarrollo y mantenimiento de la base de datos.

Otro estudio [Hoxmeier04] sugiere que deben existir métricas de calidad de datos bajo cuatro dimensiones:

- Calidad del proceso. Se refiere a considerar durante el proceso de desarrollo de sistemas, calidad en cuanto a conocimiento y modelado conceptual del dominio de la información de la empresa u organización durante el análisis. Calidad en cuanto al modelo lógico y físico de la base de datos, es decir, normalización, aislamiento de los datos, mejores tiempos de respuesta, etc. En cuanto a la instrumentación, aspectos relacionados con la cohesión de los datos, el desempeño, su complejidad. En cuanto a su mantenimiento, lo relacionado al incremento de funciones de la base de datos o corrección de errores.

- Calidad de los datos. Preservar la integridad de los datos es clave para obtener una base de datos de calidad. Para el Administrador de la Base de Datos (ABD), el identificar, interpretar, y aplicar las reglas del negocio, representa un reto difícil.
- Calidad del modelo de datos. El proceso de diseño de la base de datos es uno de los más difíciles de llevar a cabo pues básicamente se guía por los requerimientos y necesidades del consumidor de los datos. Aquí se identifican, se priorizan y visualizan los requerimientos semánticos.
- Calidad del comportamiento de la base de datos. Aspectos como la facilidad de uso son importantes en este rubro. El ABD tratará de construir un modelo de base de datos que se asemeje a las percepciones del usuario pero a su vez, deberá diseñar un producto que se instrumente, mantenga y modifique en una forma efectiva y eficiente.

Por otro lado, en [Lee02] los autores presentan un modelo de lo que para ellos significa calidad de información para los usuarios consumidores de la información y los correspondientes administradores. Este modelo se forma de cuatro cuadrantes, dependiendo de que si la información se considera un servicio o un producto y de que si las mejoras pueden validarse contra una especificación formal o contra la expectativa del cliente.

Aunado a lo anterior, existe un estudio [Wang96] en el que los autores determinan cuatro categorías de calidad de datos:

- En cuanto a sus propiedades intrínsecas. Los datos representan la información disponible sin errores.
- En cuanto a su contexto. La información es fiable para las operaciones en la base de datos y para el usuario.
- En cuanto a su representación. Provee de información concisa y consistente.
- En cuanto a su acceso. La facilidad de realizar operaciones adecuadas durante la estancia de las personas u organizaciones.

Los estudios anteriores [Hoxmeier04], [Lee02], [Wang96] basados en el aseguramiento de la calidad de los datos como producto de software han propuesto métricas de calidad de

datos, tomando en cuenta el número de defectos o los cambios realizados en la base de datos. En contraste en el presente trabajo se hace una reflexión considerando que las bases de datos deberán ser tratadas de tal forma que durante el proceso de Desarrollo y Mantenimiento de Software puedan ser validadas y verificadas antes de su integración con el sistema.

Por otro lado, se han propuesto criterios de calidad para bases de datos relacionales aparte de la normalización. En [Piattini01] los autores proponen el uso del modelo MPM (Meta- Pregunta-Métrica) para definir métricas para las bases de datos relacionales. MPM es un modelo de tres niveles, conceptual, operacional y cuantitativo propuesto para definir métricas. Por ejemplo, de una meta dada se deriva una pregunta que cuestiona la forma en que una característica de la base de datos influye en la meta y, finalmente, dada una respuesta a la pregunta, se definen métricas con el objeto de medir la característica de la base de datos.

Asimismo, se han definido métricas para validar la complejidad estructural y los indicadores de entendibilidad de los modelos E/R [Genero08]. En cuanto a los diagramas UML de clases, en [Genero07] los autores proponen métricas que miden la capacidad de recibir mantenimiento (mantenibilidad) que tienen los diagramas tal es el caso de métricas que miden el tiempo para entender y modificar un diagrama.

En este trabajo se incorporan nuevas prácticas y guías de base de datos en MoProSoft para evaluar la calidad de la base de datos durante el análisis, diseño, construcción, pruebas e integración de la base de datos.

En [Rumble03] se estudia la calidad del proceso de construcción de una base de datos como un producto. El objetivo de la construcción de un producto de base de datos es que pueda ser usado en forma intuitiva y que siga estándares normalmente aceptados en cuanto al diseño de su interfaz de tal forma que un científico del campo del dominio de la base de datos pueda usarla con un mínimo de ayuda o documentación. Podemos ver como en la construcción de este tipo de productos, el usuario consumidor de la información juega un papel importante en la concepción del producto.

En el presente trabajo se definen actividades que permitirán realizar un mejor análisis de los procesos de la base de datos dentro del sistema.

En cuanto a accesibilidad y seguridad de los datos tenemos lo siguiente. La accesi-

bilidad a la información puede definirse como la manera en que los usuarios podrán ingresar datos y la forma de acceder a ellos en la base de datos. El objetivo es establecer un control óptimo que mantenga la integridad de los datos, por lo que el contenido de la información almacenada tendrá la validez y admisibilidad [ISO-2700105] del dominio correspondiente en el que se pretenda construir la base de datos.

En cuanto a la seguridad de la información y el Desarrollo y Mantenimiento de Software en [ISO-2700105] se establece que deberá existir validación de los datos a introducir, con el objeto de evitar la corrupción de la información, se deberá velar por la autenticidad e integridad de mensajes entre aplicaciones, así como deberán validarse los datos de salida de las aplicaciones. Cabe mencionar que para los fines del modelo de procesos de software es primordial que la calidad de la base de datos a utilizar en las aplicaciones se produzca desde las etapas tempranas de análisis y diseño, por esta razón las modificaciones propuestas a MoProSoft, así como las prácticas y guías se concibieron contemplando apoyar al usuario del modelo en todas las fases de desarrollo y teniendo en cuenta las dimensiones de calidad estudiadas.

En base al estudio realizado, consideramos que en relación a las bases de datos pueden considerarse principalmente tres dimensiones de calidad. Por un lado, podemos encontrar criterios de calidad en relación al proceso de desarrollo y mantenimiento de la base de datos. Asimismo, existen criterios de calidad en relación a la base de datos misma, tomada esta como un producto de software y en relación al uso de la base de datos. En la Figura 1.2 mostramos las tres dimensiones de calidad mencionadas y los diferentes criterios a considerar [García-García08].

## 1.4. Descripción de Capítulos

El presente trabajo se estructura de la siguiente manera. El Capítulo I describe el motivo por el cual se realiza esta investigación, así como, los objetivos y los trabajos relacionados que se consideraron durante el desarrollo de las componentes de base de datos en el proceso de Desarrollo y Mantenimiento de Software. En el Capítulo II mencionaremos los conceptos necesarios para la descripción de las componentes de base de datos, se presentan los diferentes roles para el tratamiento de la base de datos, los modelos de datos, los sistemas

Calidad del proceso de desarrollo y mantenimiento de la base de datos	Calidad de la base de datos como producto	Calidad en cuanto al uso de la base de datos
Análisis de requerimientos Conocimiento del dominio Modelado conceptual Diseño Modelado lógico (Diagramas E/R y UML de clases) Normalización Modelado físico Implementación Cohesión de los datos Desempeño Complejidad Mantenimiento Modificabilidad Posibilidad de corregir errores	Interna Integridad conceptual Funcionalidad Fiabilidad Mantenibilidad Portabilidad Representación concisa Libre de error Externa Accesibilidad Seguridad Eficiencia Oportunidad	Completez Credibilidad Seguridad Facilidad de uso Cantidad apropiada Satisfacción Relevancia Efectividad

Figura 1.2: Dimensiones de calidad de base de datos

de información y las herramientas para definirlos, asimismo, la historia de MoProSoft y las actividades relacionadas con el proceso de Desarrollo y Mantenimiento de Software, el Capítulo III contiene las modificaciones propuestas a MoProSoft y las diferentes prácticas (actividades, roles, verificaciones y validaciones e infraestructura) y guías que se integran al proceso de Desarrollo y Mantenimiento de Software de MoProSoft para el buen desarrollo y mantenimiento de las bases de datos, asimismo, se explica un caso de estudio en el que se utilizaron los componentes de base de datos para MoProSoft. Este caso de estudio se realizó en la Dirección General de Servicios de Cómputo Académico (DGSCA), de la Universidad Nacional Autónoma de México en particular en la Subdirección de Sistemas. Por último, el Capítulo IV describe las conclusiones y alcances registrados durante el desarrollo de las componentes de base de datos para MoProSoft, además de los trabajos futuros ha realizar para establecer las componentes de base de datos en MoProSoft.



## Capítulo 2

# Marco teórico

El presente capítulo habla de los conceptos relacionados con la calidad de los datos, a continuación describiremos la importancia de éstos, así como sus características.

### 2.1. La información

En la actualidad la información es una necesidad, es decir, está estrechamente relacionada con el desarrollo económico y social. En [Piattini07], la investigación, la planificación y la toma de decisiones demandan que la información sea precisa, oportuna, completa, coherente y adaptada a las necesidades particulares de cada usuario (empresario, empleado), así como, de cada evento.

Hoy en día [Piattini07], es importante analizar el marco legal e institucional en el que se registra el derecho de la información, además de los condicionantes tecnológicos.

El desarrollo de la información nos ha llevado a tres enfoques distintos que se relacionan, puesto que cada uno reúne características de los anteriores al ir incrementando la generalidad de este término.

El primer enfoque es la investigación que entiende la información como instrumento indispensable y como elemento de entrada y salida. El segundo, el conocimiento transmisible, identifica a la información dentro de otro conjunto de actividades socioculturales y no solamente a los científicos y a los técnicos. Por último podemos considerar que la información refleja un recurso fundamental que puede ser utilizado en cualquier actividad humana.

En [Wietzenfeld05] la información se define como un modelo que sirve para puntualizar las estructuras fundamentales de una organización en función de los objetos que la constituyen. Además, la información como modelo permite identificar rápidamente que elementos de la organización deberán ser guardados físicamente en una base de datos.

Asimismo, la información es tomada como un eje de modelado independiente de los ejes de comportamiento y presentación, por lo que existe una mayor consistencia de la información en el desarrollo del sistema. Por tanto, el impacto se puede minimizar con la presentación y el comportamiento.

Con base en lo anterior, la información es cada vez más importante para las personas físicas y jurídicas, ya que desempeña un papel importante dentro de las empresas u organizaciones.

La información posee diferentes cualidades que representan un recurso fundamental para las organizaciones y las personas, como se muestran en la Figura 2.1:



Figura 2.1: La información y sus cualidades

- Precisión: la información debe reflejar el evento al cual se refiere y su sistema de medición expresado con poca variabilidad.

- Oportunidad: es el tiempo que tarda el proceso de la información en estar disponible al usuario.
- Completitud: es la cualidad de la información que ha de ser completa para cumplir con las necesidades del usuario.
- Significativa: se refiere a que la información sea comprensible e interesante sin tener grandes cantidades de datos que no puedan ser entendidos, es decir, que exprese un máximo contenido semántico.
- Coherente: representa el sentido de las reglas semánticas utilizadas en el dominio del problema.
- Seguridad: establece que la información debe ser protegida contra accesos no autorizados, así como daños por causas físicas o lógicas.

Las cualidades de la información pueden ser incongruentes con otras; para obtener una precisión es necesario encontrar un equilibrio entre estas, dentro de objetivos concretos del sistema en cuestión contemplando el costo que éste pueda generar.

Asimismo, la información contenida en una base de datos debe ser suministrada de manera precisa, es decir, que el tratamiento de la información llevada de manera electrónica no mejora la calidad de los datos que son procesados, ya que la máquina sólo puede indicar algunos errores o incompatibilidades entre los datos y probablemente cambiar datos erróneos por otros que no lo sean.

## **2.2. Sistemas de información**

Para que una empresa u organización funcione es necesario transmitir información entre sus diferentes elementos, generalmente del interior al exterior del sistema. Así pues, [Piattini07] primeramente se debe realizar por medio de un sistema de información informal o lo que es lo mismo comunicación entre empleados, complementándolo con un sistema de información formal, nombrado organizacional.

Un sistema es un conjunto de elementos con un orden relacionados entre sí que contribuyen a un determinado objetivo. Asimismo, el sistema puede considerarse como un subsistema de otro sistema más amplio, por ejemplo, la base de datos puede ser un subsistema de un sistema de información y éste como un subsistema de la organización. Por lo tanto, el objetivo de un sistema de información es satisfacer las necesidades de la empresa u organización tomando los datos del dominio del problema y dando como resultado la información que la empresa u organización necesita para su administración y toma de decisiones. El sistema de información está compuesto por dos partes [Piattini07]: sistema objeto, que se refiere a la información que da la empresa u organización y los resultados de ésta y el sistema dinámico, que se refiere al control de las salidas en función de los objetivos de la empresa u organización y que son variables en el tiempo.

En [Sommerville04] un sistema de información es todo aquel que tiene comunicación con una base de datos, el cual controla los accesos a la base de información. Asimismo, un sistema de información contiene una capa superior que contiene los elementos necesarios para establecer una comunicación con los usuarios y una capa inferior que contiene la base de datos del sistema. Además de, las capas mencionadas existe la capa de comunicación que permite las entradas y salidas de la interfaz de usuario, y la capa de recuperación de la información que permite acceder y actualizar la información contenida en la base de datos.

Por otro lado, en [Shannon07] un sistema se define como un grupo de objetos unidos de alguna manera por alguna interacción o interdependencia, para realizar una función específica, además de, elementos y parámetros que constituyen el sistema de interés.

Con base en lo anterior, se puede definir el sistema de información como una unión de elementos, relacionados de manera ordenada acorde a las necesidades de la empresa u organización con reglas que contribuyen con la información propia de la organización determinando el sistema objeto definiendo los objetivos necesarios de la empresa u organización, coleccionando, elaborando y almacenando datos internos y externos de éste, haciendo más fácil su uso, procesamiento y presentación de los mismos.

### 2.3. Componentes de un sistema de información

El sistema de información [Piattini07] está formado por un conjunto de componentes como se muestra en la Figura 2.2.

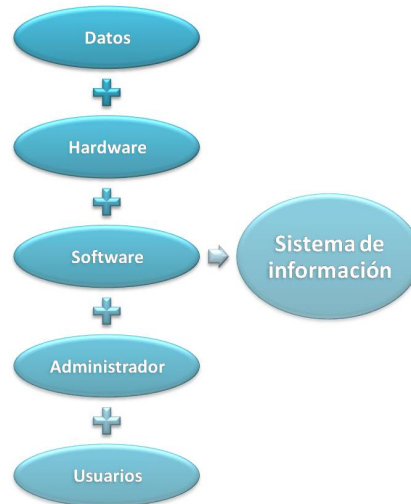


Figura 2.2: Componentes de un sistema de información

- Los datos forman parte del contenido del sistema de información, los cuales deben de acoplarse a los objetivos que se intentan lograr con dicho sistema. Se pueden distinguir dos tipos de datos:
  - Los datos referenciales: contienen referencias de los documentos en donde se encuentra la información mas no la información en sí misma
  - Los datos hechos: contienen la información requerida, los cuales se subdividen en:
    - \* Datos estructurados: son aquellos atributos o variables fuertemente tipados (int, float, string). Cada atributo en una relación está definido para todos los registros. Por ejemplo, registros, bases de datos relacionales.
    - \* Datos no estructurados: no poseen definiciones de tipos, no organizados de acuerdo a ningún patrón, no existe el concepto de variables o atributos. Por ejemplo documentos de texto sin estructura, correo electrónico, páginas html.

- \* Datos semi estructurados: son datos que están entre estructurados y no estructurados, contienen variables pobremente tipadas, un atributo de un registro puede ser otro registro. Por ejemplo los documentos XML.
- El hardware: se refiere a todo el equipo físico que soporta al sistema.
- El software: es el soporte lógico, el cual debe administrar los datos en base a un Sistema de Gestión de Base de Datos, asimismo se encarga de la comunicación y solución a los requerimientos específicos de la empresa u organización.
- El administrador: se refiere a la persona cuya función es definir reglas de autorización y el uso correcto y continuo de los datos.
- Los usuarios: se refiere a las personas que utilizan el sistema de información, pueden ser informáticos (analistas, administradores y diseñadores de base de datos y programadores) o usuarios finales.

## 2.4. Concepto de base de datos

Existen diferentes conceptos de base de datos [Piattini06] donde se puede observar que la mayoría de ellos concuerdan con ciertos elementos, pero en otros difieren o carecen de conceptos fundamentales, que son propios de las bases de datos y que las distinguen del concepto de archivos.

Primeramente, la base de datos es un conjunto de datos interrelacionados y estructurados con contenido semántico, almacenados en un dispositivo secundario. Las bases de datos procuran ayudar a la empresa u organización, como un recurso que se integra a los ya utilizados. Con base en lo anterior, “las bases de datos deberán dar servicio a usuarios y aplicaciones concurrentes a diferencia de los sistemas de archivos que sólo están diseñados para servir a un único usuario o aplicación”. [Piattini07]

Es importante que la base de datos almacene interrelaciones, ya que en el mundo real existen limitantes semánticas referentes al dominio del problema. “Para lograr estas limitantes semánticas, la base de datos se apoya en un modelo de datos”. [Piattini07]

Una característica fundamental de las bases de datos es que no exista redundancia de los datos, es decir, que se eliminen las duplicidades de los mismos, con el objeto de hacer más eficiente su tratamiento. De esta manera se dice que la base de datos debe estar exenta de redundancia lógica, lo que se denomina actualización de forma única por el usuario, aunque se admita redundancia física por razones de eficacia, también llamada redundancia controlada por el sistema.

“Otro aspecto que debe ser considerado en las bases de datos es la independencia de los datos y tratamientos, tanto físicos como lógicos”. [Piattini07]

“La estructura o esquema de la base de datos es la definición o descripción del conjunto de datos contenidos en la base de datos, es decir, los metadatos, índices y los mismos datos”. [Piattini07]

Por otro lado, la información que se proporcione a los usuarios deberá ser integra conforme las reglas definidas en el Sistema manejador de base de datos, además, los datos deben ser actualizados y recuperados para éste fin.

Con base a lo mencionado anteriormente y de acuerdo a las características descritas, se define la base de datos de la siguiente manera, conjunto de datos completos, almacenados en un dispositivo secundario sin redundancia lógica y con un mínimo de redundancia física. Los datos deben ser utilizados concurrentemente por usuarios y aplicaciones, además de ser independientes de ellos, el esquema de la base de datos deberá ser definido de manera única apoyándose con un modelo de datos que permita entender las interrelaciones y limitantes del dominio del problema. La actualización y recuperación de los datos deben ser procesos bien descritos y fáciles de entender, para facilitar la seguridad de los datos [Piattini07].

## 2.5. Tipos de usuarios propios de la base de datos

En [Elmasri04] los usuarios de una base de datos pueden clasificarse en:

- Administrador de la base de datos (ABD): es el responsable de autorizar el acceso a la base de datos, de coordinar y monitorear su uso, y de adquirir el software y hardware requeridos. El ABD debe verificar que los datos no sean violados, mantengan su confidencialidad, disponibilidad e integridad [Piattini07]. Para ello, el ABD no debe

permitir consultas o actualizaciones que ocasionen fallos lógicos o físicos. Por tanto, será el encargado de definir el control de accesos y uso de la base de datos.

- Diseñador de bases de datos (DBD): en la literatura la función del diseñador de bases de datos puede encontrarse como parte de las responsabilidades del ABD [Silbertchatz07], o separada [Elmasri04]. Los diseñadores de bases de datos son los responsables de identificar los datos que serán almacenados en la base de datos y de seleccionar las estructuras apropiadas para representar y almacenar los datos. Es su responsabilidad comunicarse con el Analista para garantizar que los requerimientos relativos a la información manejada se vean reflejados en el diseño de la base de datos. Consideramos que es tal la especialización del ABD y el DBD, que deberán considerarse como roles separados, diferenciados de los demás roles involucrados [García-García08].
- Usuarios: según la forma de interactuar con la base de datos, se han identificado los siguientes usuarios de una base de datos [Elmasri04]:
  - No sofisticados: son usuarios que realizan consultas y/o actualizaciones a la base de datos utilizando programas de aplicación creados previamente para interactuar con la base de datos. Como ejemplo tenemos un cajero bancario, un operador de un mostrador de una aerolínea, un cajero en un supermercado, etc.
  - Programadores de aplicaciones: son profesionales en computación que escriben programas que facilitan las tareas que no pueden ser llevadas a cabo por los usuarios habituales y esporádicos [De Miguel06] con diferentes herramientas de software para emitir formas y reportes.
  - Usuarios sofisticados: son usuarios parecidos a los usuarios no sofisticados ya que necesitan consultar y/o actualizar la base de datos, pero las tareas que llevan a cabo no son fácilmente formalizables [Piattini07], por tanto, deben interactuar con la base de datos por medio de lenguajes de consulta de bases de datos como SQL.
  - Usuarios especializados: son usuarios sofisticados que escriben aplicaciones especializadas no tradicionales como son sistemas expertos, sistemas de extracción de



conocimiento, sistemas que almacenan tipos de datos complejos (como gráficas, videos, audios, etc.).

- Analistas de sistemas: son usuarios que identifican los requerimientos de los usuarios no sofisticados y especifica los requisitos para los usuarios de transacciones.
- Usuarios de transacciones: son los encargados de realizar interfaces con las cuales los usuarios no sofisticados se comunican con la base de datos.

Los usuarios mencionados, de alguna forma u otra, deberán ser considerados durante el desarrollo y/o el mantenimiento de un producto de software [García-García08].

## 2.6. Sistema manejador de base de datos

Como se ha indicado anteriormente, toda empresa u organización esta compuesta por tres niveles de gestión como son: operacional, gerencial y de dirección, por lo que el sistema de información se encuentra distribuido en estos tres niveles.

Por lo tanto, la base de datos [Piattini07] tendrá la tarea de comunicar e integrar los subsistemas y aplicaciones generadas en cada uno de los tres niveles, asegurando el cumplimiento de las necesidades de los usuarios. El encargado de establecer una comunicación entre la base de datos y los tres niveles de gestión será el sistema manejador de base de datos (SMBD).

Por consiguiente, un sistema manejador de base de datos es un conjunto organizado de programas, procedimientos, lenguajes, etc. que proporciona los elementos necesarios a los diferentes usuarios para representar y tratar el conjunto de datos que se encuentran almacenados en la base de datos garantizando la integridad y la seguridad de los datos. El sistema manejador de base de datos debe estar acorde a las necesidades de la empresa u organización teniendo como base el tipo de estructura de base de datos que se vaya a utilizar, es decir, que los beneficios propios de la base de datos se vean reflejados en la realidad.

Asimismo, el SMBD debe proveer de operaciones que ayuden a los usuarios a describir, manipular y controlar los datos, estas pueden agruparse en dos categorías: la primera es que afecte la totalidad de los datos (creación, reestructuración y consulta total)

y la segunda se refiere a las operaciones sobre registros concretos (actualización y consulta selectiva) [Piattini06].

Aunado a lo anterior, el SMBD define tres funciones que gestionan las operaciones que pueden realizar, las cuales se describen a continuación:

- **Función de definición:** el SMBD proporciona un lenguaje para la definición de tablas, los atributos que la componen, las restricciones semánticas así como las características de tipo físico o almacenamiento. El usuario que define el esquema de la base de datos es el diseñador de la base de datos.
- **Función de manipulación:** el SMBD define instrucciones de actualización y consulta que permiten a los usuarios de la base de datos recuperar información o actualizarla por motivos de cambios en los datos.
- **Función de control:** el SMBD debe proporcionar funciones de servicio que permitan modificar el tamaño de los archivos, la obtención de algoritmos de ordenación y búsqueda, la carga de archivos, la seguridad física (copias de seguridad y restablecimiento) y de protección (control de acceso, confidencialidad e integridad).

De acuerdo a las funciones mencionadas el SMBD debe dar soporte a los distintos niveles de estructuración de una base de datos. Los SMBD proporcionan lenguajes y procedimientos que permiten la comunicación con la base de datos; unos están dirigidos a las funciones de definición o manipulación, mientras que otros se encaminan a los diferentes tipos de usuarios o de aplicaciones.

Así pues, si consideramos el tipo función, estaremos hablando de lenguajes de definición y lenguajes de manipulación; por otro lado si consideramos al tipo usuarios, el lenguaje que tendríamos podría estar definiendo a los usuarios informáticos o a los usuarios finales.

En ambos casos, el SMBD debe proporcionar un conjunto de procedimientos que faciliten las tareas para definir, administrar, extraer o manipular los datos de la base de datos. Con base en lo anterior, se definen los lenguajes que permiten interactuar con la base de datos [Piattini07]:

- Lenguaje de Definición de Datos (LDD): se utiliza para definir las estructuras que permitan describir los datos con facilidad y precisión, proporcionando medios para describir la estructura lógica global, la estructura interna y las estructuras externas.
- Lenguaje de Manipulación de Datos (LMD): se utiliza para añadir, suprimir, modificar y consultar datos en la base de datos, una vez que el esquema de la base de datos se ha definido. El LMD puede encontrarse embebido en un lenguaje anfitrión o directamente en el SMBD. Un LMD puede ser procedimental o no procedimental, es decir, más específicos para usuarios informáticos y menos formales para usuarios finales.

En conclusión, se puede acordar que el motivo fundamental del SMBD es establecer las interfaces apropiadas entre los tipos de usuarios y la base de datos que satisfagan las necesidades de la empresa u organización, la Figura 2.3 muestra estas interfaces.

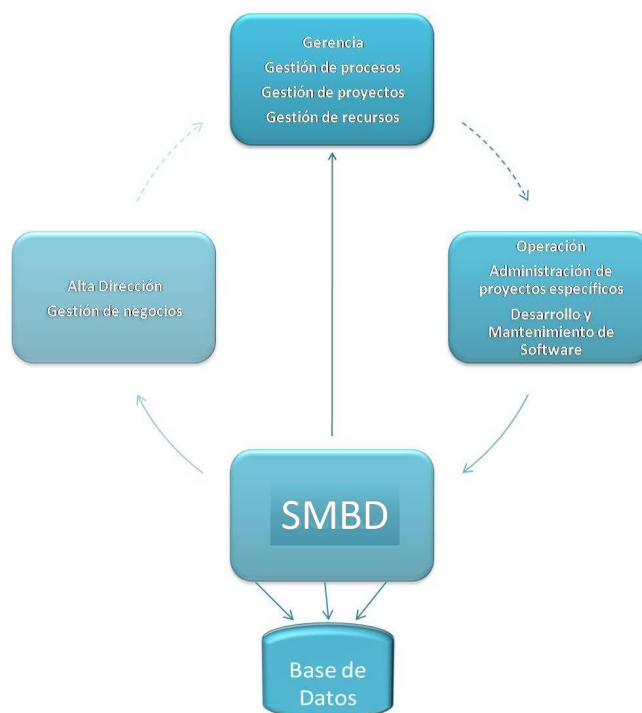


Figura 2.3: El SMBD como vínculo entre la base de datos y los niveles de gestión

## 2.7. Modelo de datos

Un modelo de datos es aquel que describe de una forma abstracta cómo se representan los datos, sea en una empresa, en un sistema de información o en un sistema de gestión de base de datos. Básicamente consiste en una descripción de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores.

Un modelo de datos Figura 2.4 consiste en:

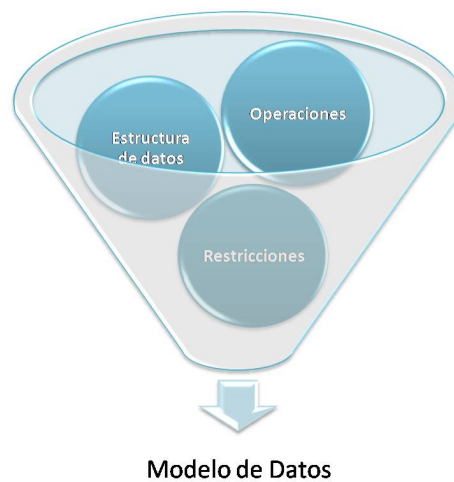


Figura 2.4: Partes de un modelo de datos

- Objetos: son las entidades que existen y se manipulan.
- Atributos: son las características básicas de los objetos.
- Interrelaciones: es la forma en que enlazan los distintos objetos entre si.

Un modelo de datos esta formado por tres componentes principales, los cuales se describen de la siguiente manera:

- Estructura de datos: es una forma de organizar un conjunto de datos elementales con el objetivo de facilitar su manipulación. Un dato elemental es la mínima información

que se tiene. La estructura de datos esta constituida por un conjunto de elementos que permiten la construcción del modelo.

- Operadores o reglas de inferencia: los operadores o reglas que pueden ser empleados en cualquier instancia de los tipos de datos.
- Reglas generales de integridad: las reglas de integridad son aquellas que permiten definir un estado de los datos de manera explícita o implícita, con el fin de verificar la semántica del universo a modelar y así comprobar los datos almacenados en la base de datos.

Los modelos de datos [Piattini07] facilitan la creación de categorías, lo que lleva a la clasificación de estos modelos considerando dos tipos de ellos Figura 2.5:

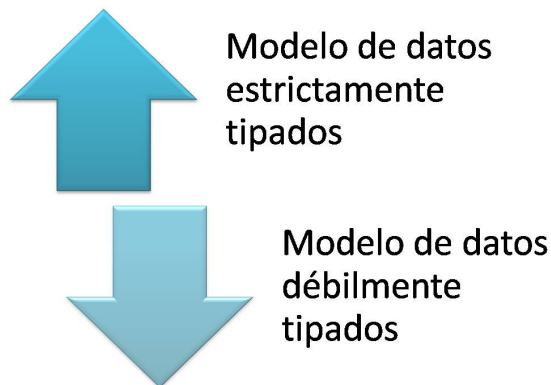


Figura 2.5: Tipos de modelos de datos

- Modelos de datos estrictamente tipados: se refiere a que los datos que son definidos deben pertenecer a una categoría del esquema, es decir, que los datos tienen un tipo de dato establecido.
- Modelos de datos débilmente tipados: en este modelo los datos no tienen que pertenecer a una categoría ya que estos pueden existir por si mismos.

Dentro de las bases de datos el modelo de datos más empleado es el modelo de datos estrictamente tipado por su capacidad para el manejo de grandes cantidades de datos, esto es posible debido al agrupamiento en categorías de los mismos.

Asimismo, un modelo de datos [De Miguel06] define sus reglas según las cuales los datos han de ser estructurados. La representación de la realidad mediante un modelo da lugar a un esquema, el cual describe las categorías existentes. Por lo tanto, se dice que la realidad no contempla sólo aspectos estáticos, como aquellos que representa el esquema, sino también propiedades dinámicas que han de ser especificadas en las operaciones que contenga la base de datos, principalmente las de consulta y actualización.

Por lo tanto, un modelo de datos puede ser utilizado de las siguientes maneras:

- Como una herramienta para especificar los tipos de datos y la organización de los mismos que son aceptados dentro de una base de datos.
- Como una plataforma para el desarrollo de una metodología general de diseño para las bases de datos.
- Como una base para el tratamiento de familias de lenguajes de alto nivel para el manejo de consultas y datos.
- Como la parte fundamental en el diseño de la arquitectura de un manejador de bases de datos.

## 2.8. Una mirada a MoProSoft

A continuación se da un panorama de MoProSoft, con el fin de enfatizar los procesos que se encuentran en el modelo, dado que en nuestra presentación nos referiremos a los componentes de base de datos que se integran en el mismo.

En 2002 la Secretaría de Economía (SE) empezó a organizar mesas de trabajo para definir las estrategias del programa para el desarrollo de la industria de software, hoy conocido bajo el nombre de PROSOFT. Alcanzar niveles internacionales en capacidad de procesos fue una de las estrategias del programa. Como presidente, en ese entonces, de la

Asociación Mexicana para la Calidad en Ingeniería de Software (AMCIS) la Dra. Hanna Oktaba [Hanna07] comenzó a coordinar los trabajos de un grupo interesado en este tema. Primero se analizaron las características básicas de la industria mexicana de software con respecto a su tamaño y las capacidades de sus procesos.

Con base en el estudio realizado, se confirmó que la gran mayoría de las empresas en México son MyPES (Micro y Pequeñas Empresas) y que el promedio de capacidades de procesos está a nivel 1 (en realidad 0.9),<sup>1</sup>. Se realizaron entrevistas en varias empresas con estas características cuestionando sobre sus necesidades con respecto a un modelo de procesos y su evaluación. Como respuesta se concluye que las empresas mexicanas necesitan un modelo que sea fácil de entender, práctico y barato. En ese momento la SE determina que se haga una norma mexicana para estandarizar los procesos de las empresas mexicanas.

Con esta especificación de requerimientos se comenzaron a revisar los modelos de procesos disponibles incluyendo ISO9000:2000, CMM-SW, ISO12207, ISO15504 y la versión inicial de CMMI, lo que llevó a concluir que ningún modelo cumplía con los requerimientos. Por tal motivo, se decide proponer a la SE un proyecto para desarrollar un modelo de procesos y un método de evaluación a la medida de la industria mexicana. Uno de los compromisos a cubrir era alcanzar las prácticas de CMM-SW nivel 3 e ISO9000:2000, en el caso de modelo de procesos, y cumplir con los lineamientos de ISO15504, con respecto al método de evaluación.

Entre septiembre y diciembre de 2002 se logró conjuntar a través de la AMCIS a un equipo de 11 personas, en su mayoría mujeres profesionistas, que aportaron su conocimiento y experiencia para generar MoProSoft (el Modelo de Procesos para la Industria de Software) como documento base para la norma mexicana. En junio de 2003 la SE lo hizo público a través del portal software.net.mx.

Para completar la norma se necesitaba definir el método de evaluación basado en MoProSoft como modelo de procesos. Para tal fin se conjuntó otro equipo de trabajo en 2003, en esta ocasión la mayoría fueron hombres, y como consecuencia de este trabajo se

---

<sup>1</sup>H. Oktaba and C. Alquicira and A. Su Ramos and A. Martínez and G. Quintanilla and M. Ruvalcaba and F. López and M. Rivera and M. Orozco and Y. Fernández and M. Flores Modelo de Procesos para la Industria del Software (nivel 1 Realizado, nivel 2 Gestionado, nivel 3 Establecido, nivel 4 Predecible y nivel 5 Optimizado).

definió EvalProSoft (el método de Evaluación de Procesos de Software), que fue revisado por los dos únicos Lead Assessors con los que se contaba en ese momento en México.

En el 2004 el modelo de procesos y el método de evaluación se formalizaron como norma en el subcomité de software del NYCE (Normalización y Certificación en Electrónica).

### 2.8.1. Modelo de procesos para la industria de software (MoProSoft)

El propósito del Modelo de Procesos para la Industria de Software (MoProSoft) en México es fomentar la estandarización de los procesos incorporando mejores prácticas en gestión e ingeniería de software. El modelo permite elevar la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.

El modelo esta basado en procesos considerando tres niveles básicos de una organización los cuales son [Oktaba05]:

- Alta Dirección: aborda las prácticas relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos.
  - Gestión de Negocios: establecer la razón de ser de la organización, sus objetivos y las condiciones para lograrlos. Adicionalmente habilita a la organización para responder a un ambiente de cambio y a sus miembros para trabajar en función de los objetivos establecidos.
- Gestión: aborda las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección, además de proporcionar los elementos para el funcionamiento de los procesos de la Categoría de Operación.
  - Gestión de Procesos: establecer los procesos de la organización, en función de los procesos requeridos identificados en el plan estratégico. Así como definir, planificar e implantar las actividades de mejora en los mismos.
  - Gestión de Proyectos: asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.



- Gestión de Recursos: conseguir y dotar a la organización de los recursos humanos, infraestructura, ambiente de trabajo y proveedores, así como crear y mantener la base de conocimiento de la organización.
- Operación: aborda las prácticas de los proyectos de Desarrollo y Mantenimiento de Software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y entrega a ésta la información y productos generados.
  - Administración de Proyectos Específicos: el propósito de este proceso es establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.
  - Desarrollo y Mantenimiento de Software: el propósito de este proceso es realizar sistemáticamente las actividades de análisis, diseño, construcción, integración y pruebas de productos de software, cumpliendo con los requerimientos especificados.

### 2.8.2. Descripción del proceso de Desarrollo y Mantenimiento de Software en MoProSoft

El proceso de Desarrollo y Mantenimiento de Software de MoProSoft [Oktaba05], contiene diferentes secciones que permiten definir las actividades y productos propios del tratamiento de software. Para identificarlos analizamos las diferentes fases de un ciclo de Desarrollo y Mantenimiento de Software en MoProSoft para la realización sistemática y de calidad de productos de software:

#### 1. Inicio y Requerimientos:

Es importante lograr un entendimiento y compromiso claro entre los miembros del *Equipo de Trabajo* y los empresarios y usuarios, definiendo las necesidades del cliente. Asimismo, se deben establecer los roles, prácticas y actividades que efectuaran durante el desarrollo y/o mantenimiento del software, además de, utilizar las guías establecidas

para cada una de las actividades. Se debe obtener el *Plan de Actividades* y el *Plan de Pruebas de Sistema*.

2. Análisis y Diseño:

Se deben considerar el estudio de los requerimientos obtenidos en la fase de requerimientos, que servirá como base para producir la estructura y arquitectura de los componentes de software. Es necesario, generar una buena descripción del problema para tomarlo como base para la construcción. Se debe obtener el *Plan de Pruebas de Integración*.

3. Construcción:

Se realizan los diferentes productos de software que se describieron en la fase de análisis y diseño, además de realizar las pruebas unitarias de cada producto que se genere durante el desarrollo de la construcción. Se obtiene como base para la siguiente fase los productos probados.

4. Integración y Pruebas:

Se debe integrar y probar los productos de software generados en la fase de construcción, con base en los planes de prueba de sistema y de integración, a fin de que el software cumpla las necesidades del cliente. Los productos generados en la etapa de integración y pruebas deben ser almacenados en los productos destinados a la *Configuración de Software*.

5. Cierre:

Se realiza la recopilación de todos los documentos y productos de software generados e integrados en la *Configuración de Software* con el fin de poder ser entregados. Asimismo, se deben anotar las lecciones aprendidas, reporte de mejoras y reporte de mediciones, para su análisis y ejecución en proyectos futuros.

MoProSoft proporciona un proceso de Desarrollo y Mantenimiento de Software que no particulariza en sistemas de información donde su principal componente sea la base de datos, por tal motivo, consideramos que los componentes de base de datos deben ser

integrados dentro del modelo para incrementar el nivel de operación de los usuarios de MoProSoft donde la base de datos sea el principal componente del sistema.

## Capítulo 3

# Componentes de base de datos en MoProSoft

Primeramente, se realizó un estudio del proceso de Desarrollo y Mantenimiento de Software de MoProSoft [Oktaba05], identificando diversas prácticas y productos propios del tratamiento de bases de datos. Para identificarlos se consideraron las siguientes fases de un ciclo de Desarrollo y Mantenimiento de Software:

- Inicio y Requerimientos: es necesario que en estas fases se logre un compromiso de colaboración de parte del usuario (cliente). Además de asegurar que los usuarios que conocen el dominio del problema y la información que se manipula tengan una participación total dado que las necesidades de todos los usuarios de la base de datos deberán reflejarse en los requerimientos y éstos deben ser comprendidos completamente por los analistas. Para lograr el entendimiento total del dominio del problema se deben llevar a cabo las siguientes actividades.
  - Revisión del plan de trabajo por los miembros del *Equipo de Trabajo* para lograr un entendimiento común y compromiso para su realización.
  - Lograr un entendimiento claro y preciso entre el cliente y el personal de desarrollo.
  - Incluir el desarrollo de estándares que especifiquen cómo realizar la recolección de datos, cómo especificar su formato, qué documentación será necesaria y cómo

se va a llevar a cabo el diseño y la implementación de la base de datos.

Así pues, los productos que se generan son: un bosquejo de los datos reales de la empresa u organización, las necesidades de los usuarios, las entidades y las interrelaciones existentes entre los datos, las funcionalidades de cada área, así como los posibles cambios a módulos de los sistemas ya implementados. Es decir, que se debe generar un primer modelado de conceptualización que logre indicar la realidad de los datos de la empresa, identificando las diferentes funciones de los datos que manejan cada uno de los usuarios. Asimismo, es necesario realizar una validación con el cliente con el motivo de utilizar cómo línea base los productos generados en ésta fase.

- **Análisis y Diseño:** la base de datos si bien debe ser considerada como un componente más de software, debe tratarse con especial cuidado puesto que de no hacerlo, podría causar en un futuro problemas graves dentro de la empresa u organización debido a que en muchos casos la base de datos es un componente de software estratégico para la organización.

En la fase de *Análisis* deberá obtenerse el esquema conceptual de la base de datos a partir del modelado de conceptualización, identificando entidades e interrelaciones existentes [Elmasri04]. Dicho lo anterior, el desarrollo del esquema conceptual podrá realizarse utilizando diferentes herramientas como son los diagramas E/R o UML de clases.

Posteriormente, y dependiendo del modelo de datos escogido, en la fase de *Diseño* se obtendrá el esquema lógico de la base de datos.

Ya definido el esquema lógico, se procede a realizar el esquema físico de la base de datos, definiéndose las estructuras físicas adecuadas para un manejo eficiente de los datos.

Con base en lo anterior, los productos generados en la fase de *Análisis y Diseño* deben ser presentados a los clientes para su aprobación y validación con el objeto de tomarlos como línea base para la fase de *Construcción*. Es importante considerar que esta validación se debe ser realizada por el cliente o si el contrato lo pide.

- **Construcción:** como una pieza de software más, la base de datos deberá estar contemplada en la construcción del producto de software y por tanto los productos asociados a ella deberán incluirse en la herramienta que administre la *Configuración de Software*. Un aspecto importante a considerar en esta fase es el hecho de que existen productos y código propios de la base de datos independientemente de los programas de aplicación. Tal es el caso de productos como los diagramas del diseño conceptual y lógico, el código de creación de la base de datos, o en general el código de definición de los datos utilizando un lenguaje de definición de datos (LDD), el código para implementar las restricciones de integridad, el código referente a la optimización de las consultas a la base de datos, como es la creación de índices, el código para llevar a cabo la migración y/o carga de los datos.

Todo el código mencionado deberá administrarse en la fase de construcción de igual forma como se administra todo el código del sistema. Este código deberá construirse y probarse adecuadamente, siguiendo los estándares definidos.

Es importante mencionar que los productos obtenidos en la fase de Construcción deberán ser validados por el Responsable de Pruebas Unitarias con la participación de los clientes a fin de establecer la línea base de la fase de *Integración y Pruebas* de la base de datos. Asimismo, se debe considerar que el cliente deberá tener el conocimiento para validar los productos generados.

- **Integración y Pruebas:** el *Plan de Pruebas de Integración* y el *Plan de Pruebas de Sistema* deberán contemplar pruebas propias de software que utiliza una base de datos, como son pruebas de volumen y desempeño, pruebas de integridad, pruebas de seguridad, pruebas de concurrencia y recuperación para el caso de sistemas transaccionales y las pruebas de migración. Nótese que los datos de prueba, deberán estar dentro de la administración de la *Configuración del Software*.

Cómo se señaló anteriormente, la validación de los productos generados en la fase de *Construcción* se deben realizar si el cliente puede hacerlo o si el contrato lo pide, de manera que el cliente apruebe los productos de base de datos integrados y probados con la interfaz de usuario.

- Cierre: los manuales deberán incluir lo correspondiente a la base de datos. El *Manual de Usuario* deberá contemplar, a manera de ejemplo, aspectos relativos a la ejecución de consultas no planeadas. Asimismo, el *Manual de Operación* deberá contemplar aspectos sobre la administración de la base de datos como puede ser lo referente a la eliminación física de registros dados de baja, así como respaldo, recuperación, arranque, terminación, reorganización, ampliación, etc., al volumen de los datos y a la ejecución optimizada de consultas.

Cabe mencionar que en el *Equipo de Trabajo* deberá participar el Administrador de la Base de Datos (ABD) al cual deberá consultársele todo lo relacionado con respaldo, recuperación, arranque, terminación, reorganización, ampliación, etc., a los datos, por la visión integral que tiene de los mismos.

Con base en lo anterior, se considera proponer componentes de base de datos que apoyen su desarrollo, utilizando prácticas y guías que permitan aumentar la calidad de la base de datos cumpliendo con las necesidades de la empresa u organización.

A continuación se muestran los componentes de base de datos que se proponen en MoProSoft.

### **3.1. Propuestas a MoProSoft para integrar las actividades para el tratamiento de las bases de datos**

Para realizar las modificaciones que se sugieren a MoProSoft, con el objeto de contemplar las actividades propias de bases de datos sin modificar la estructura del modelo, primeramente se estudiaron los elementos del proceso de Desarrollo y Mantenimiento de Software [Oktaba05].

El estudio realizado motivó la propuesta fundamentada en cambios al modelo mismos que se detallan más adelante. Dichos cambios incluyen enfatizar los aspectos propios de la base de datos que se deben considerar en las secciones de propósito, descripción general de actividades y salidas. Asimismo, se integran las prácticas en la sección de roles, actividades, verificaciones y validaciones. Por último se integran las guías en la sección de identificación

de guías. Los elementos de las prácticas y guías de base de datos perfeccionarán las actividades de especificación de requisitos, análisis y diseño, construcción, pruebas e integración, propias del tratamiento de una base de datos. Con base en lo mencionado anteriormente, estos componentes de base de datos no aumentan el nivel de dificultad del modelo, pero por otro lado, permitirán incrementar la precisión de las prácticas en el proceso de Desarrollo y Mantenimiento de Software de proyectos con bases de datos como componente central.

A continuación se describen las modificaciones propuestas al modelo con respecto a los componentes de base de datos para cada sección y subsección del proceso de Desarrollo y Mantenimiento de Software de MoProSoft:

- **Propósito.** Es necesario definir los requisitos, análisis y diseño, construcción, pruebas e integración propios del tratamiento de la base de datos durante el desarrollo del software.
- **Descripción general de actividades.** Se detallan las tareas necesarias para el tratamiento de la base de datos dentro de cada fase definida en el modelo.
- **Salidas.** Se considera importante definir e implementar en la especificación de requerimientos los elementos funcionales de la empresa u organización, el esquema conceptual, las reglas de integridad, los niveles de accesibilidad, seguridad y mantenimiento correspondientes a los esquemas de la base de datos y los datos mismos. Asimismo, se integra el modelo detallado en la fase de análisis y diseño que conste de esquema lógico estándar, esquema lógico específico y esquema físico. También se incorpora en el plan de pruebas de sistemas e integración los planes de prueba correspondientes a la base de datos.
- **Roles.** Se incorporan los roles de Administrador de la Base de Datos (ABD) y Diseñador de la Base de Datos (DBD), con el fin de aumentar la comprensión del dominio del problema de la empresa u organización y repercutirlo en la base de datos. Además se integran características propias del tratamiento de base de datos en los roles de Programador, Responsable de Pruebas, Revisor y Responsable de Manuales con el fin de lograr un entendimiento total con los roles de base de datos.



- Actividades
  - Fase de Requerimientos. Se integran prácticas propias de la base de datos como son: descripción y entendimiento del dominio del problema, la conceptualización de la base de datos y la transformación de la conceptualización de la base de datos al esquema conceptual.  
Además es necesario realizar el *Plan de Pruebas de Base de Datos*, considerando el volumen, el desempeño, la integridad, la accesibilidad, la seguridad, la concurrencia y en general la calidad de los datos.
  - Fase de Análisis y Diseño. Se incorporan prácticas de definición del modelo de datos, restricciones de integridad, diseño de los esquemas tanto lógicos, como físicos de la base de datos. Asimismo, es necesario definir el *Plan de Pruebas de Integración de la Base de Datos*, tomando en cuenta la carga o migración de los datos, conservación de la integridad, su desempeño, seguridad y en general la calidad de los datos.
  - Fase de Construcción. Se colocan las prácticas que son necesarias para realizar la creación de la base de datos, además de, la carga o la migración de los datos, asimismo, las pruebas unitarias correspondientes a la base de datos con la interfaz.
  - Fase de Integración y Pruebas. Se integran las prácticas correspondientes a la realización de las pruebas de base de datos y de integración, utilizando los planes definidos en las fases de *Especificación de Requerimientos y Análisis y Diseño*.
- Verificaciones y Validaciones. Se sugiere tomar en cuenta las prácticas de las fases de requerimientos, análisis, diseño y construcción propias del tratamiento de la base de datos. Asimismo, debemos incorporar prácticas de verificación del *Plan de Pruebas de Base de Datos* en el *Plan de Pruebas de Sistema*. Cabe mencionar, que se considera incorporar prácticas de verificación del *Plan de Pruebas de Integración de Base de Datos* en el *Plan de Pruebas de Integración*, esto con el objetivo de comprobar y corregir el desarrollo y/o mantenimiento de la base de datos.

En resumen las modificaciones sugeridas al proceso de Desarrollo y Mantenimiento

de Software de MoProSoft incluyen lo siguiente:

- 2 roles. ABD, DBD,
- 6 tareas de especificación de requerimientos para el tratamiento de las bases de datos,
- 7 tareas de *Análisis y Diseño*,
- 1 actividad y 5 tareas de *Construcción*,
- 1 actividad y 2 tareas de *Integración y Pruebas*,
- 1 práctica de apoyo y consultoría de base de datos,
- 1 guía para levantamiento de requisitos del conocimiento del dominio,
- 1 guía para el *Análisis y Diseño de la Base de Datos*,
- 1 guía para el *Plan de Pruebas de la Base de Datos*,
- 1 guía para el *Plan de Pruebas de Integración de la Base de Datos*,
- en la sección 5.4 del modelo se sugiere una modificación en producto de software referente al tratamiento de las base de datos,
- asimismo, se sugiere una modificación en el rol del Responsable de Desarrollo y Mantenimiento de Software.

### 3.2. Innovaciones y prácticas requeridas en MoProSoft

A la luz de lo expuesto a continuación se muestran las modificaciones propuestas a MoProSoft incluyendo las prácticas y guías de base de datos. Para lo anterior, se muestra el sitio donde deberá ir cada uno de los elementos propuestos que corresponden al tratamiento de bases de datos en el proceso de Desarrollo y Mantenimiento de Software. Cabe aclarar que se procuró realizar las modificaciones mínimas, considerando que las responsabilidades, actividades y tareas toman en cuenta los conceptos arriba expuestos.

Primeramente, se muestran los cambios necesarios en la definición general del proceso. Lo cambios sugeridos en MoProSoft para la integración de las prácticas y guías de base de datos se presentan como *Agregar*<sup>1</sup> y/o *Modificar*<sup>2</sup>.

#### Definición general del proceso

<b>Proceso</b>	OPE.2 Desarrollo y Mantenimiento de Software
<b>Categoría</b>	Operación (OPE)
<b>Propósito</b>	<i>Modificar</i> : El Propósito de Desarrollo y Mantenimiento de Software es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requerimientos especificados. <b>Debe considerar también el análisis y diseño, construcción e integración de productos de base de datos.</b>
<b>Descripción</b>	<p>- <i>Modificar</i>: <b>Requerimientos</b>: Conjunto de actividades cuya finalidad es obtener la documentación de la <i>Especificación de Requerimientos del sistema así como de la base de datos (Esquema Descriptivo, Esquema Percibido, Esquema Conceptual, Niveles de Seguridad, Reglas de Integridad), Plan de Pruebas de Sistema y Plan de Pruebas de Base de Datos</i>, para conseguir un entendimiento común entre el cliente y el proyecto.</p> <p>- <i>Modificar</i>: <b>Análisis y Diseño</b>: Conjunto de actividades en la cuales se analizan los requerimientos especificados para producir una descripción de la estructura de los</p>

<sup>1</sup>Se utiliza cuando el texto es nuevo en las definiciones de MoProSoft

<sup>2</sup>Se utiliza cuando dentro de las definiciones ya establecidas en MoProSoft se coloca texto para integrar las componentes de bases de datos

componentes de software, la cual servirá de base para la construcción. Como resultado se obtiene la documentación del *Análisis y Diseño* del sistema, ***Modelo de Base de Datos: Esquema Lógico Estándar, Esquema Lógico Específico, Esquema Físico y Diccionario de datos*** y el *Plan de Pruebas de Integración de Sistema* **incluyendo el *Plan de Pruebas de Integración de Base de Datos***.

- *Modificar: Construcción*: Conjunto de actividades para producir *Componente(s)* de software **incluyendo, si es el caso, la *Estructura Física de la Base de Datos*** que correspondan al *Análisis y Diseño* **basándose en el *Modelo de Base de Datos***, así como la realización de las pruebas unitarias al sistema **y a la carga o migración de la base de datos**. Como resultado se obtienen el (los) *Componente(s)* de software probados.

- *Modificar: Integración y Pruebas*. Conjunto de actividades para integrar y probar los componentes de software, basadas en los *Planes de Pruebas de Integración y de Sistema* **incluyendo los *Planes de Prueba de Integración de Base de Datos y Planes de Prueba de Base de Datos***, con la finalidad de obtener **un sistema** que satisfaga los requerimientos especificados. Se genera la versión final del *Manual de Usuario, Manual de Operación y Manual de Mantenimiento*. Como resultado se obtiene el producto de *Software* probado y documentado.

La siguiente modificación corresponde a las salidas que se deben obtener a partir de las entradas, es importante mencionar que las salidas deseadas de la base de datos se encuentran dentro de las ya definidas en MoProSoft, esto con el fin de facilitar el entendimiento del modelo. La siguiente tabla muestra las salidas propias del tratamiento de la base de datos.

#### Salidas

Nombre	Descripción	Destino
<i>Especificación de requerimientos</i>	<b>Agregar: Debe redactarse una descripción general tanto de la base de datos</b>	

Nombre	Descripción	Destino
	<p><b>como de su uso dentro de la empresa u organización.</b></p> <p>Descripción de requerimientos:</p> <p><i>Agregar: - Esquema Descriptivo:</i></p> <p><b>Necesidades que permitan ver claramente la información con todo su contenido semántico, a partir de las reglas de la empresa u organización.</b></p> <p><i>Agregar: - Esquema Percibido:</i></p> <p><b>Delimitación de qué entidades, atributos, interrelaciones y restricciones semánticas deben ser considerados para entender el dominio del problema, éstos deberán ser expresados en un lenguaje natural.</b></p> <p><i>Agregar - Esquema conceptual:</i></p> <p><b>Descripción rigurosa de los distintos contenidos de información que describen a la organización y que son necesarios para su funcionamiento, utilizando para ello el modelo E/R o diagramas UML de clases.</b></p> <p><i>Agregar - Niveles de seguridad:</i></p> <p><b>Especificación de los recursos establecidos para cada usuario dependiendo de su función dentro de la organización, de acuerdo con reglas de confidencialidad, accesibilidad e integridad.</b></p> <p><i>Agregar - Reglas Integridad:</i></p> <p><b>Especificación rigurosa de los datos con respecto a los atributos, interrelaciones y restricciones semánticas, asegurando la integridad referencial, de entidad, de dominio y definida por el usuario.</b></p>	<p>Admón. de proyectos específicos</p>

Nombre	Descripción	Destino
	<p><i>Modificar: - Mantenimiento:</i></p> <p>Descripción de los elementos que facilitarán la comprensión y la realización de las modificaciones futuras del software <b>incluyendo los de la estructura de la base de datos cómo los datos mismos.</b></p>	
<i>Análisis y Diseño</i>	<p><i>Modificar</i> Este documento contiene la descripción textual y gráfica de la estructura de los componentes de software. <b>Incluye, si es el caso, el Modelo de Base de Datos.</b> El cual consta de las siguientes partes:</p> <p><i>Agregar</i></p> <p>- <b>Modelo de Base de Datos:</b></p> <p><b>Descripción de las reglas y mecanismos para la representación del dominio del problema de los datos que se pretenden sean persistentes. Para esto se definen los siguientes elementos:</b></p> <p>- <b>Esquema Lógico Estándar:</b> Especificación del modelo de datos soportado por el Sistema Manejador de Base de Datos (SMBD), sin restricciones propias del SMBD.</p> <p>- <b>Esquema Lógico Específico:</b> Descripción del modelo de datos en términos del Lenguaje de Definición de Datos propio del SMBD.</p> <p>- <b>Esquema Físico:</b> Especificación de los registros físicos de la base de datos.</p>	Admón. de proyectos específicos

Nombre	Descripción	Destino
	- <b>Diccionario de Datos: Descripción de la información sobre los datos almacenados en la estructura de la base de datos.</b>	
<i>Plan de Pruebas de Sistema</i>	Agregar - <b>Plan de Pruebas de Base de Datos: Identificación de las pruebas que se deben realizar de acuerdo al Esquema Conceptual y Niveles de Seguridad definidos en la Especificación de Requerimientos.</b>	Admón. de proyectos específicos
<i>Plan de Pruebas de Integración</i>	Agregar - <b>Plan de Pruebas de Integración de Base de Datos: La integración de la base de datos debe tomar en cuenta el Esquema Físico del Modelo de Base de Datos.</b>	Admón. de proyectos específicos

### Prácticas

La próxima modificación describe los roles, actividades y mediciones definidos para el desarrollo y/o mantenimiento de software propios del tratamiento de la base de datos. Dicho lo anterior, consideramos especificar claramente las actividades y mediciones propias de la base de datos con el fin de facilitar su ejecución.

Primeramente mostraremos los roles involucrados en el desarrollo de la base de datos.

### Roles

Rol	Abreviatura	Capacitación
<i>Agregar:</i> <b>Administrador de Base de Datos</b>	<b>ABD</b>	<b>Conocimiento y experiencia en administración de base de datos.</b>

Rol	Abreviatura	Capacitación
<i>Agregar:</i> Diseñador de Base de Datos	<b>DBD</b>	<b>Conocimiento y experiencia en diseño de base de datos.</b>
Programador	PR	<i>Agregar:</i> Conocimiento y/o experiencia en la programación, integración y pruebas unitarias <b>a nivel aplicación y base de datos.</b>
Responsable de Pruebas	RPU	<i>Agregar:</i> Conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistema. <b>Deberán contemplarse conocimiento y experiencia de pruebas de integración a nivel base de datos.</b>
Revisor	RE	<i>Agregar:</i> Conocimiento en las técnicas de revisión y experiencia en el Desarrollo y Mantenimiento de Software <b>y de base de datos.</b>
Responsable de Manuales	RM	<i>Agregar:</i> Conocimiento en las técnicas de redacción y experiencia en el Desarrollo y Mantenimiento de Software <b>y de base de datos.</b>

### Actividades

La siguiente tabla muestra las actividades y tareas propias del tratamiento de las bases de datos.

Rol	Descripción
	<b>A2. Realización de la fase de Requerimientos (O1, O3).</b>



Rol	Descripción
<p><i>AN CL US</i> <i>DU DBD</i></p>	<p>A2.2. Documentar o modificar la <i>Especificación de Requerimientos</i>.</p> <ul style="list-style-type: none"> <li>- Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos.</li> <li>- Analizar los requerimientos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto.</li> <li>- Elaborar o modificar el prototipo de la interfaz con el usuario.</li> <li>- <i>Agregar: Analizar los datos y procesos identificando y describiendo las reglas y el detalle de los datos de una empresa.</i></li> <li>- <i>Agregar: Realizar una representación ordenada de los procesos y los datos apoyada en un modelo de datos que cumpla con ciertas propiedades como: coherencia, plenitud, no redundancia, simplicidad y fidelidad. Como resultado se obtiene el Esquema Descriptivo.</i></li> <li>- <i>Agregar: Elaborar el Esquema Percibido del dominio del problema descrito en lenguaje natural a partir del Esquema Descriptivo.</i></li> <li>- <i>Agregar: Elaborar o modificar el Esquema Conceptual interpretando las frases del lenguaje natural en el que está descrito el Esquema Percibido utilizando el modelo E/R o diagrama UML de clases para convertir las frases en entidades, atributos e interrelaciones. Es necesario ir refinando y ordenando sucesivamente el Esquema Conceptual en el modelo E/R extendido o UML de clases.</i></li> <li>- <i>Agregar: Elaborar o modificar los Niveles de Seguridad considerando las reglas y procesos de la empresa u organización.</i></li> </ul>

Rol	Descripción
	<ul style="list-style-type: none"> <li>- <i>Agregar: Definir las Reglas de Integridad a partir del Esquema Conceptual considerando las restricciones del dominio del problema.</i></li> <li>- Generar o actualizar la <i>Especificación de Requerimientos</i>.</li> </ul>
<i>RPU AN ABD</i>	A2.7. <i>Modificar:</i> Elaborar o modificar el <i>Plan de Pruebas de Sistema</i> <b>incluyendo el Plan de Pruebas de Base de Datos.</b>
	<b>A3. Realización de la fase de Análisis y Diseño (O1, O3).</b>
<i>AN DI DU DBD</i>	<p>A3.2. Documentar o Modificar el <i>Análisis y Diseño:</i></p> <ul style="list-style-type: none"> <li>- Analizar la <i>Especificación de Requerimientos</i> para generar la descripción de la estructura interna del sistema y su descomposición en subsistemas, y éstos a su vez en componentes, definiendo las interfaces entre ellos.</li> <li>- Describir el detalle de la apariencia y el comportamiento de la interfaz con base en la <i>Especificación de Requerimientos</i> de forma que se puedan prever los recursos para su implementación.</li> <li>- Describir el detalle de los componentes que permita su construcción de manera evidente.</li> <li>- <i>Agregar: Generar el Esquema Lógico Estándar a partir del Esquema Conceptual, de manera que se obtenga un modelo de datos muy parecido al SMBD que se va a utilizar, pero sin las restricciones de éste, usando técnicas de normalización para definir este esquema.</i></li> <li>- <i>Agregar: Generar el Esquema Lógico Específico a partir del Esquema Lógico Estándar, éste deberá ser descrito en el lenguaje de definición de datos.</i></li> <li>- <i>Agregar: Generar el Esquema Físico a partir del Esquema Lógico Específico teniendo en cuenta los procesos, características del SMBD, del sistema operativo, y del hardware, lo que nos llevará a cumplir los objetivos del sistema.</i></li> </ul>

Rol	Descripción
	<ul style="list-style-type: none"> <li>- Agregar <b>Generar el <i>Diccionario de Datos</i> a partir del <i>Esquema Lógico Específico</i>, con el fin de reunir la información de los datos almacenados con base en la estructura de los datos, sus interrelaciones y restricciones.</b></li> <li>- Generar o actualizar el <i>Análisis y Diseño</i>.</li> <li>- Generar o modificar el <i>Registro de Rastreo</i>.</li> </ul>
<i>RPU ABD</i>	<i>Modificar:</i> A3.7. Elaborar o modificar <i>Plan de Pruebas de Integración</i> <b>incluyendo el <i>Plan de Pruebas de Integración de Base de Datos</i>.</b>
	<b>A4. Realización de la fase de Construcción (O1,O3).</b>
<i>PR</i>	<p><i>Agregar</i> A4.2.Construir o modificar el(los) <i>Componente(s)</i> de software <b>incluyendo la <i>Estructura Física de la Base de Datos</i>:</b></p> <p><i>Agregar:</i> - <b>Construir la <i>Estructura Física de la Base de Datos</i> con base en los entregables generados durante la fase(s) de <i>Análisis y Diseño</i>.</b></p> <p><i>Agregar:</i> - <b>Implementar las reglas definidas en la fase de <i>Especificación de Requerimientos</i> específicamente en el <i>Esquema Conceptual</i> de la base de datos.</b></p> <p><i>Agregar:</i> - <b>Especificar y realizar pruebas unitarias a la base de datos con el fin de identificar que el desempeño, accesibilidad, seguridad e integridad de los datos cumplan con lo definido en el <i>Análisis y Diseño</i>.</b></p> <p><i>Agregar:</i> - <b>Realizar correcciones encontradas durante la ejecución de las pruebas unitarias a las base de datos.</b></p> <p><i>Modificar:</i> - Actualizar el <i>Registro de Rastreo</i>.</p>
<i>PR RPU</i>	<p>A5.2. Realizar integración y pruebas:</p> <p><i>Modificar:</i> - Integrar los componentes en subsistemas o en el sistema del <i>Software</i> y aplicar las pruebas siguiendo el <i>Plan de Pruebas de Integración</i> <b>incluyendo el <i>Plan de Pruebas de Integración de Base de Datos</i></b>, documentando los resultados en un <i>Reporte de Pruebas de Integración</i>.</p>

Rol	Descripción
	<b>A5. Realización de la fase de Integración y Pruebas (O1,O3).</b>
<i>RPU</i>	<i>Agregar</i> A5.6. Realizar las pruebas de sistema siguiendo el <i>Plan de Pruebas de Sistema</i> <b>incluyendo el <i>Plan de Pruebas de Base de Datos</i></b> , documentando los resultados en un <i>Reporte de Pruebas de Sistema</i> .

***Agregar: Apoyo y Consultoría en Base de Datos***

La siguiente tabla muestra una nueva actividad que servirá para tener un mejor control y apoyo con respecto al tratamiento de la base de datos.

---

*Agregar* Los roles PR, RPU, RE y RM deberán solicitar apoyo al ABD en aspectos relacionados con la base de datos.

---

### 3.3. Estudio de viabilidad de los componentes de base de datos para MoProSoft

Ahora se presenta el estudio obtenido de la implementación de los componentes de base de datos para MoProSoft. Es importante mencionar que éstos componentes de base de datos fueron diseñados para el desarrollo de sistemas donde su principal interés es la base de datos.

Primeramente se realizó una presentación en la Dirección General de Servicios de Cómputo Académico (DGSCA). La DGSCA es una organización de desarrollo de software de la Universidad Nacional Autónoma de México con nivel 3 de capacidad evaluada bajo el Modelo de Procesos para la Industria de Software, basado en la Norma Mexicana para la Industria del Software en México. Se llevó a cabo en la Subdirección de Sistemas donde se contó con la participación de la subdirectora de sistemas y su equipo de trabajo, esta presentación tuvo como objetivo dar a conocer los componentes de base de datos que requería MoProSoft en el proceso de desarrollo y mantenimiento de software. Durante la presentación se surgieron entre otros aspectos el especificar lo siguiente: los requerimientos funcionales de la base de datos, las actividades de análisis y diseño de la base de datos, los roles encargados

de dichas actividades, además de incorporar plantillas que permitieran documentar cada uno de los productos que se obtendrán durante la instrumentación de éstos componentes y así poderlos integrar fácilmente en el reporte de *Configuración de Software*.

Así pues, y con base en las observaciones se procedió a definir las prácticas, guías, plantillas y listas de verificación requeridas para los componentes de base de datos para MoProSoft. Durante la definición de éstos, y a raíz de los comentarios de la encargada del desarrollo de bases de datos de la DGSCA pudimos afinar las componentes de base de datos en el proceso de *Desarrollo y Mantenimiento de Software*.

Cabe aclarar que los componentes de base de datos son aplicables a empresas que no tienen ningún tipo de proceso establecido e incluso para empresas u organizaciones que si cuentan con este tipo de procesos, pero requieren de apoyo para llevar a cabo el desarrollo y/o mantenimiento de base de datos.

Ya definidos los componentes de base de datos para MoProSoft se realizaron diversos ejercicios, donde se observó que las prácticas concordaban con prácticas informales realizadas por los especialistas en base de datos para mejorar la componente de base de datos durante el *Desarrollo y Mantenimiento de Software*.

Asimismo, Se observó que las guías y prácticas propuestas formalizaban las prácticas utilizadas en forma intuitiva por lo que el haberlas documentado y sistematizado representó un progreso para la mejora de los procesos de la categoría de operación.

Podemos concluir que el resultado obtenido del uso de los componentes de base de datos es un progreso para la mejora de los procesos de operación en empresas u organizaciones que carecen de éstos, y en aquellas organizaciones que si tienen procesos establecidos como la DGSCA, permite a los equipos de desarrollo estandarizar tareas repetitivas relacionadas con el tratamiento de las bases de datos y en general hacen que sean debidamente documentadas.

Cabe aclarar que si bien en esta organización las prácticas y guías fueron bien recibidas en un futuro deberán realizarse ejercicios similares en otras organizaciones que desarrollen sistemas con un componente importante de base de datos.

Como un resultado adicional podemos comentar que este tipo de ejercicios debe ser motivado y promovido por la alta dirección dado que en principio el uso de las prácticas y guías de base de datos puede percibirse como trabajo adicional en vez de como una

herramienta que redundará en la calidad de las bases de datos de los sistemas de información. Se recomienda que en futuros casos de estudio en los que se requiera utilizar y evaluar los trabajos presentados se dedique un tiempo en motivar la introducción y el uso de las herramientas propuestas.

### 3.4. Guías, plantillas y listas de verificación en el proceso de Desarrollo y Mantenimiento de Software para el tratamiento de las bases de datos

Las guías para el tratamiento de la base de datos serán instrumentos que ayuden a los usuarios MoProSoft, para identificar las técnicas y herramientas que se deben utilizar para realizar las actividades correspondientes a las prácticas de bases de datos. Dentro de ellas podremos encontrar plantillas, listas de verificación y ejemplos, que mejoren la percepción de los productos que se deben obtener para cada una de las actividades. Las cuales se definen en base a su implementación durante el proceso de Desarrollo y Mantenimiento de Software de la siguiente manera: la primera guía de *Levantamiento de Requisitos de la Base de Datos* sirve para obtener el conocimiento del dominio de la empresa u organización, la segunda de *Análisis y Diseño*, sirve para pasar del esquema conceptual a los esquemas lógico y físico de la base de datos, además de generar el diccionario de datos correspondiente, la tercera guía define el *Plan de Pruebas de la Base de Datos*, por último la guía de *Plan de Pruebas de Integración de Base de Datos*, sirve para integrar la base de datos con el sistema.

A continuación se presentan las guías, plantillas, listas de verificación y ejemplos de las componentes de base de datos para MoProSoft.

La guía del *Especificación de Requisitos extensión Base de Datos* permite realizar y documentar la conceptualización de la base de datos, así como el modelo E/R para lograr un concepto más apropiado de la base de datos. Dentro de la guía se podrá encontrar una plantilla que permite al usuario guiarse en los elementos que debe contener la documentación que se va generando, también encontrará una lista de verificación que permitirá conocer la calidad del modelo E/R, además incluye dos ejemplos que permiten ver como se puede

realizar la concepción de la base de datos y posteriormente el modelo conceptual de ésta. Veamos ahora las componentes de la especificación de requisitos de la base de datos.

**Guía de Implantación para OPE.2 - A2. Realización de la fase de Requerimientos**

Proceso	OPE.2 Desarrollo y Mantenimiento de Software. - A2. Realización de la fase de Requerimientos
Actividad (es)	A2.2. Documentar o modificar la <i>Especificación de Requerimientos</i>
Tarea(s) Detalle	<ul style="list-style-type: none"> <li>- Analizar los datos y procesos identificando y describiendo las reglas y el detalle de los datos de una empresa.</li> <li>- Realizar una representación ordenada de los procesos y los datos apoyada en un modelo de datos que cumpla con ciertas propiedades cómo: coherencia, plenitud, no redundancia, simplicidad y fidelidad. Cómo resultado se obtiene el <i>Esquema Descriptivo</i>.</li> <li>- Elaborar el <i>Esquema Percibido</i> del dominio del problema descrito en lenguaje natural a partir del <i>Esquema Descriptivo</i>.</li> <li>- Elaborar o modificar el <i>Esquema Conceptual</i> interpretando las frases del lenguaje natural en el que está descrito el <i>Esquema Percibido</i> utilizando el modelo E/R o diagrama UML de clases para convertir las frases en entidades, atributos e interrelaciones. Es necesario ir refinando y ordenando sucesivamente el <i>Esquema Conceptual</i> en el modelo E/R extendido o UML de clases.</li> <li>- Elaborar o modificar los <i>Niveles de Seguridad</i> considerando las reglas y procesos de la empresa u organización.</li> <li>- Definir las <i>Reglas de Integridad</i> a partir del <i>Esquema Conceptual</i> considerando las restricciones del dominio del problema.</li> </ul>

Técnicas disponibles	<p>Algunas de las técnicas de trabajo en grupo utilizadas para la <i>Especificación de Requisitos</i> de la base de datos son: entrevistas, encuestas, sesiones JAD, grabaciones, video conferencias.</p> <p>Algunas de la técnicas utilizadas para modelar y analizar los requisitos de la base de datos son: modelo E/R extendido y modelado conceptual.</p>
Producto específico generado	<i>Esquema conceptual</i>
Herramientas de apoyo	<p>Herramienta CASE con UML:</p> <ul style="list-style-type: none"> <li>- Una herramienta que permita realizar un modelado de datos orientado a objetos,</li> <li>- que pueda generar el diseño lógico de la base de datos.</li> </ul> <p>Herramientas para el modelo E/R:</p> <ul style="list-style-type: none"> <li>- Una herramienta que permita realizar el diagrama E/R,</li> <li>- que optimice el tiempo en realización del diagrama.</li> </ul>
Criterio de selección	<ul style="list-style-type: none"> <li>- Modelo E/R extendido: Es técnica muy útil para relacionar los conceptos e interrelaciones de la realidad de los datos con el esquema conceptual.</li> <li>- Modelado conceptual: Es una técnica para obtener una mejor conceptualización de los datos de la empresa u organización, sin tener relación con los usuarios o aplicaciones que los utilizan además de ser independiente de las características del hardware.</li> </ul>
Soportes	<ul style="list-style-type: none"> <li>- Plantilla de Especificación de Requerimientos extensión Base de Datos</li> <li>- Lista de verificación del modelo E/R</li> <li>- Concepción de la base de datos</li> <li>- Modelado conceptual</li> </ul>



Bibliografía	- Piattini, Mario. Marcos, Esperanza. Calero, Coral. Vela, Belén. "Tecnología y diseño de base de datos", Alfaomega, Ra - ma. 2007.
--------------	---

***Plantilla de Especificación de Requerimientos extensión base de datos***

En la plantilla de *Especificación de Requerimientos* extensión base de datos, se definirán los elementos que permiten elaborar o modificar el *Esquema Conceptual* de la base de datos, para obtener la concepción del dominio del problema de la empresa u organización.

COMPETISOFT v 0.2 <Nombre de la Empresa>  
<Nombre del Proyecto>

---

<Nombre del proyecto>

Documento de Especificación de Requerimientos extensión base  
de datos

Historia de revisiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<Detalles>	<Nombre>

Documento de Especificación de Requerimientos extensión base de datos	Página 1 de 4
---	---------------

---

COMPETISOFT v 0.2 <Nombre de la Empresa>  
<Nombre del Proyecto>

---

TABLA DE CONTENIDO

1. DESCRIPCIÓN DEL CONTENIDO
2. OBJETIVOS DE LA BASE DE DATOS
3. REQUISITOS DE LA BASE DE DATOS

3.1 Esquema Descriptivo

3.2 Esquema Percibido

#### 4. DEFINICIÓN DEL ESQUEMA CONCEPTUAL

4.1 Modelo Entidad - Interrelación

4.2 Diagrama UML de clases

#### 5. GLOSARIO DE TÉRMINOS

Documento de Especificación de Requerimientos extensión base de datos
---

Página 2 de 4
---------------

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

### 1. DESCRIPCIÓN DEL CONTENIDO

*[Describir los argumentos que se utilizará para la solución. Los argumentos deben resaltar la manera en cómo se manejan los datos de la empresa actualmente, detallar los problemas del manejo de los datos que existen y definir la forma en cómo se modificarán los procesos del dominio del problema de la empresa con el nuevo producto de base de datos. La descripción de los procesos del dominio del problema de la empresa deben realizarse en un lenguaje natural. Veamos un ejemplo para una base de datos de Hospital.]*

Se desea construir una base de datos para un *Hospital* que permita resolver los problemas de la repetición de información. Por ejemplo cada vez que se registra un paciente se repite la información sobre su ingreso. Sucede lo mismo cada vez que se asigna un médico al paciente que ingresa ya que siempre se deben registrar los datos del médico que atenderá al paciente. Aunque el problema de espacio no sea grave si lo es el mantener los datos consistentes. Es decir, cada vez que se introduce una tupla habrá que comprobar que toda la información redundante sea consistente. Por ejemplo cada vez que se introduce un ingreso habrá que comprobar que coincide la procedencia con los datos previamente guardados en la tabla paciente, lo mismo con paciente y médico. Es decir se deben habilitar mecanismos para realizar todas estas comprobaciones en el momento de realizar inserciones o modificaciones sobre la tabla, ya sea mediante triggers, reglas, restricciones o procedimientos.

Con base en lo anterior, se debe diseñar una base de datos que de seguimiento a los pacientes que ingresan al *Hospital*, además de, llevar un control de los médicos que son asignados a los pacientes que han ingresado. Para realizar la construcción de la base de datos del *Hospital* se deben realizar los esquemas conceptual, lógico y físico correspondientes al diseño y construcción de la base de datos, asimismo, es necesario definir el diccionario de datos para visualizar los detalles de la estructura física de la base de datos.

## 2. OBJETIVOS DEL PRODUCTO DE BASE DE DATOS

*[En esta sección se debe describir el alcance del producto de base de datos. Deben relacionarse con los objetivos del plan de proyecto. Veamos un ejemplo con el sistema del Hospital.]*

Diseñar e integrar una base de datos para un *Hospital* que administre los ingresos de los pacientes, además de, controlar el médico que es asignado para atender al paciente.

## 3. REQUISITOS DE LA BASE DE DATOS

*[En esta sección se debe definir en un **Esquema Descriptivo** el dominio del problema de la empresa, identificando y describiendo las reglas y el detalle de los datos de una empresa. El dominio del problema en esta primera etapa debe especificar los procesos y los datos de la empresa apoyándose en un modelo de datos que cumpla con ciertas propiedades fijando para ello objetivos sobre el dominio del problema que se está estudiando. Asimismo, el Esquema Descriptivo se expresa en lenguaje natural con la intención de comenzar a comprender el dominio del problema para el que se requiere una base de datos. Teniendo definido el **Esquema Descriptivo** se debe describir el **Esquema Percibido** del dominio del problema con el fin de depurar los procesos y los datos identificados. Este esquema describe lo que se desea almacenar, por tanto, es preciso interpretar las frases del lenguaje natural en el que está descrito el **Esquema Percibido**, convirtiéndolas en elementos del modelo E/R, cómo son: entidades, atributos e interrelaciones. Una vez definido el **Esquema Percibido** podremos obtener más fácilmente el **Esquema Conceptual**.]*

*Veamos un ejemplo del diseño de una base de datos de un Hospital para la definición de los **Esquemas Descriptivo y Percibido** que servirán como base para la elaboración del **Esquema Conceptual**.*

### 3.1 Esquema Descriptivo:

- Problema del dominio: Se desea realizar un sistema que de seguimiento a los pacientes que ingresan en un *Hospital*, asimismo, debe llevar el control de los pacientes que llegan al servicio de urgencias y que posteriormente son ingresados a la planta correspondiente (traumatología, cuidados intensivos, cardiología, etc.) bajo la supervisión de un médico responsable.
- Identificación de las reglas y detalle de los datos:
  - Los pacientes están identificados por el número de seguro social. El *Hospital* almacena nombre del paciente, apellidos del paciente, domicilio, población, provincia, código postal, número de teléfono, número de historial clínico y descripción. El paciente puede ingresar al *Hospital* por el servicio de urgencias y posteriormente al área específica.
  - El *Hospital* almacena la fecha de ingreso, la procedencia, número de planta, número de cama y especificaciones del paciente.
  - Para identificar a los médicos se asigna un código de identificación del médico. El *Hospital* almacena el nombre, apellidos, especialidad, cargo y descripción.

### 3.2 Esquema Percibido:

- Identificación de entidades: La entidad que surge inmediatamente es Paciente. Otras entidades posibles son Médico e Ingreso. La primera se refiere a los médicos que son responsables de los pacientes y la segunda al ingreso en el *Hospital*.
- Identificación de atributos:
  - Conjunto de entidades Paciente: número de historial clínico, número de seguro social, nombre del paciente, apellidos del paciente, domicilio, población, provincia, código postal, número de teléfono y descripción.
  - Conjunto de entidades Ingreso: procedencia, fecha de ingreso, número de planta, número de cama y especificaciones del paciente.

- Conjunto de entidades Médico: código de identificación del médico, nombre, apellidos, especialidad, cargo, descripción.
- Identificación de interrelaciones: Por una parte tenemos a los pacientes que realizan ingresos y, por otra, a los médicos que atienden a los pacientes. Según esto aparecen dos interrelaciones: Realiza: Paciente x Ingreso y Atiende: Ingreso x Médico. Ninguna de ellas tiene atributos asociados.
- Identificación de restricciones: (de llave primaria para las entidades)
  - Conjunto de entidades Paciente: número de seguro social.
  - Conjunto de entidades Médico: código de identificación del médico.
  - Conjunto de entidades Ingreso: ID (puesto que no tiene bien definido un atributo que lo pueda identificar de manera única).
- Identificación de restricciones: (de cardinalidad)
  - Relación Realiza:
    - \* En caso de los ingresos sólo pueden corresponder a un paciente.
    - \* Para el caso de que el paciente ingresa varias veces al *Hospital*.
  - Relación Atiende:
    - \* En el caso de los ingresos estos pueden ser atendidos por un sólo médico.
    - \* Para el caso de los médicos estos pueden tener varios ingresos.

Una vez definidos los *Esquemas Descriptivo y Percibido* podremos definir el *Esquema Conceptual* del problema del dominio al que queremos dar solución.

#### 4. DEFINICIÓN DEL ESQUEMA CONCEPTUAL

[En esta sección se debe definir el **Esquema Conceptual** a partir del *Esquema Percibido* con la intención de tener una mejor concepción del dominio del problema. Usando técnicas para su estructuración como el modelo E/R o los diagramas UML de clases.

Veamos cómo se debe elaborar el **Esquema Conceptual** a partir del *Esquema Percibido* definido anteriormente. Para esto, utilizaremos las técnicas del modelo E/R y/o los diagramas UML de clases.]

##### 4.1 Modelo Entidad - Interrelación:

Modelo E/R: Es una técnica que se utiliza para elaborar el *Esquema Conceptual* ya que permite identificar claramente las entidades, los atributos, las interrelaciones y aunque no bien definidas las restricciones de los datos. Veamos ahora la obtención del diagrama E/R del *Esquema Percibido* de la base de datos del *Hospital*, Figura 3.1.

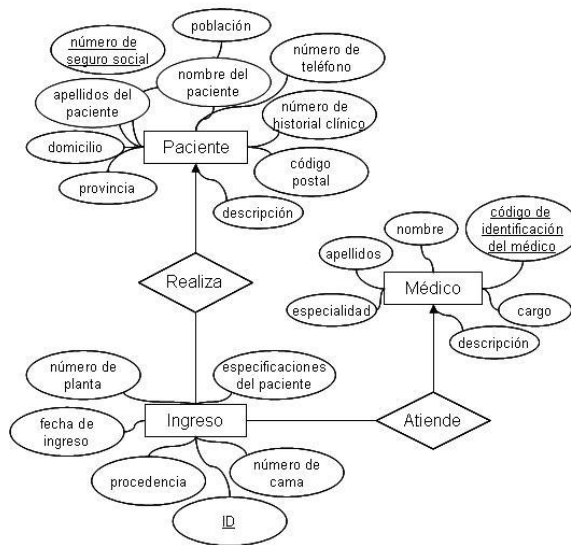


Figura 3.1: Diagrama E/R de la base de datos *Hospital*

##### 4.2 Diagrama UML de clases:

Diagrama UML de clases: Es una técnica que se utiliza para definir el *Esquema*

*Conceptual* puesto que permite identificar las clases de entidad, sus atributos, las interrelaciones, además se puede tener en una primera instancia los métodos que efectuarán las clases de entidad dentro del sistema. Veamos ahora la obtención de los diagramas UML de clases del *Esquema Percibido* de la base de datos del *Hospital*, Figura 3.2.

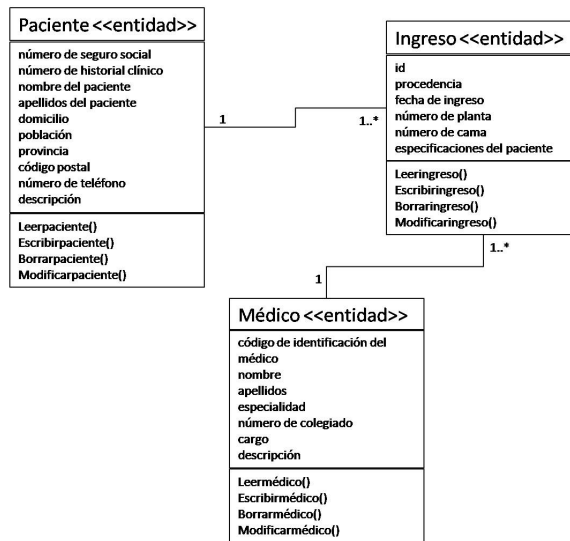


Figura 3.2: Diagrama UML de clases de la base de datos *Hospital*

## 5. GLOSARIO DE TÉRMINOS

*[En esta sección se relacionan los términos propios del contexto y su significado. Este glosario sirve como vocabulario de referencia de todos los involucrados en el proyecto. Veamos un ejemplo para la base de datos Hospital.]*

Término	Descripción
Atributo	Son las características propias de una entidad o de una relación muchos a muchos.
Ejemplar	Son los datos que se encuentran almacenados en el esquema, por lo tanto, contienen valores que en un determinado momento toman un cierto dominio.

Entidad	Es cualquier objeto real o abstracto del cual deseamos tener información persistente.
Esquema	Es la descripción de la estructura de la base de datos. Lo que nos lleva a definirlo cómo una percepción de una determinada realidad interpretada de acuerdo con un cierto modelo.
Esquema Conceptual	También denominados de alto nivel, facilitan la descripción global del conjunto de información de la empresa al nivel más próximo al usuario, por lo que sus conceptos son cercanos al mundo real (entidades, atributos, interrelaciones, etc.).
Esquema Descriptivo	Permite recoger una primera información del problema de dominio en lenguaje natural.
Esquema Percibido	Define en un lenguaje natural las entidades, atributos, interrelaciones y restricciones semánticas del problema de dominio.
Interrelación	Se define cómo una asociación entre entidades.

Documento de Especificación de Requerimientos extensión base de datos	Página 4 de 4
---	---------------

A continuación se muestra la lista de verificación del modelo E/R, esta lista nos confirma que la construcción del modelo E/R sea la correcta y que en realidad contenga lo necesario para la descripción del dominio del problema.

COMPETISOFT v 0.2	<Nombre de la Empresa>
<Nombre del Proyecto>	

Tipo de Artefacto: Modelo Entidad - Interrelación
Nombre del Sistema A Revisar:
Nombre del Artefacto a Revisar:
EVAL = <b>Cumple</b> ; <b>No cumple</b>



CÓDIGO	DESCRIPCIÓN	EVAL C/N
REQ-ER-01	El <i>Esquema Descriptivo</i> para la base de datos <i>Hospital</i> define el problema del dominio claramente en un lenguaje natural.	
REQ-ER-02	El <i>Esquema Percibido</i> para la base de datos <i>Hospital</i> identifica las entidades, interrelaciones y restricciones.	
REQ-ER-03	El Esquema Conceptual contiene el modelo E/R o el diagrama UML de clases.	
REQ-ER-04	El modelo E/R define correctamente las entidades, interrelaciones (asociaciones), cardinalidad (multiplicidad) y llaves.	
REQ-ER-05	Las entidades identificadas para la base de datos <i>Hospital</i> están escritas en singular y con la primer letra mayúscula.	
REQ-ER-06	Los atributos identificados para la base de datos <i>Hospital</i> definen los valores que tendrán cada una de las entidades.	
REQ-ER-07	Para la base de datos <i>Hospital</i> se identifica correctamente el atributo que describe completamente cada una de las entidades.	
REQ-ER-08	La cardinalidad del conjunto de entidades Paciente e Ingreso y Médico e Ingreso, define claramente el máximo y mínimo de ejemplares que pueden estar interrelacionados.	
REQ-ER-09	En caso de que se haya realizado el diagrama UML de clases se define correctamente el nombre de la clase, los atributos, las interrelaciones y su tipo de asociación.	

CÓDIGO	DESCRIPCIÓN	EVAL C/N
REQ-ER-10	Los nombres de clases para la base de datos <i>Hospital</i> están escritos en singular y con la primer letra en mayúscula.	
REQ-ER-11	Los atributos describen lo que debe contener la clase.	
REQ-ER-12	Las interrelaciones cumplen con las restricciones definidas en el <i>Esquema Percibido</i> .	
REQ-ER-13	La cardinalidad de las clases de Paciente e Ingreso y Médico e Ingreso, define claramente el máximo y mínimo de ejemplares que pueden estar relacionados.	
REQ-ER-14	Cumple la 1FN si todos sus atributos son atómicos (indivisibles).	
REQ-ER-15	Se encuentra en 2FN si está en 1FN y cada atributo que no es llave es completamente dependiente de la llave.	
REQ-ER-16	Cumple la 3FN donde la dependencia funcional $X \rightarrow A$ , si X es una super llave y A es un miembro de la llave candidata de la relación.	
REQ-ER-17	Se encuentra en FNBC si cada atributo que determina completamente a otro, es clave candidata.	
REQ-ER-18	Cumple la 4FN si cada una de sus dependencias múltiples no funcionales $X \twoheadrightarrow Y$ , X es una clave candidata o un conjunto de claves primarias.	
REQ-ER-19	Se encuentra en 5FN, si la tabla esta en 4FN, además de, que cada relación de dependencia se encuentra definida por las claves candidatas.	

Observaciones
---------------

Fecha	Elaborado por	Revisado por

Otro componente que se integra en MoProSoft es el *Plan de Pruebas de Base de Datos*, esta guía permite realizar el *Plan de Pruebas de Base de Datos* lo que permitirá realizar pruebas de la base de datos de forma unitaria, además contiene una plantilla que nos indica que productos se deben generar para incorporarlos en el reporte de configuración de software, por último se incluye una lista de verificación que nos indicará si se cumple o no con los estándares de calidad establecidos en MoProSoft. A continuación se muestran las componentes de pruebas de base de datos que se integran a MoProSoft.

**Guía de Implantación para OPE.2 - A2 Realización de la fase de Requerimientos**

Proceso	OPE.2 Desarrollo y Mantenimiento de Software. A2 Realización de la fase de Requerimientos
Actividad (es) / Tarea(s)	A2.7. Elaborar o modificar el <i>Plan de Pruebas de Sistemas</i> incluyendo el <i>Plan de Pruebas de Base de Datos</i> .
Producto específico generado	<i>Plan de Pruebas de Base de Datos</i>
Técnicas disponibles	Una de las técnicas que podemos utilizar para elaborar el <i>Plan de Pruebas de Base de Datos</i> es el Diagrama de Gantt, para establecer la agenda para la realización de las pruebas.
Herramientas de apoyo	Procesador de Texto

Criterio de selección	<p>Para definir el Plan de Pruebas de Base de Datos se debe establecer quién, cuándo, dónde y cómo se realizarán las actividades para probar la base de datos. El definir un <i>Plan de Pruebas de Base de Datos</i> brindará el conocimiento al equipo de trabajo para establecer modelos de datos bien definidos de manera integra y segura al cliente. Los puntos principales del Plan de Pruebas de Base de Datos son:</p> <ol style="list-style-type: none"> <li>1. Introducción</li> <li>2. Agenda Propuesta</li> <li>3. Tipos de pruebas a la base de datos</li> <li>4. Elaboración del Plan de Pruebas de Base de Datos</li> <li>5. Condiciones</li> <li>6. Requerimientos del entorno</li> <li>7. Ejecución de las pruebas a la base de datos</li> <li>8. Productos de salida</li> <li>10. Aprobadores</li> </ol>
Soportes	<ul style="list-style-type: none"> <li>- Plantilla de Plan de Pruebas de Base de Datos.</li> <li>- Lista de verificación para el Plan de Pruebas de Base de Datos.</li> </ul>
Bibliografía	

***Planilla de Plan de Pruebas de Base de Datos***

En la plantilla de *Plan de Pruebas de Base de Datos*, se definirán las pruebas que se deben realizar a la base de datos con la intención de cumplir con las necesidades establecidas en el dominio del problema de la empresa u organización.

---

COMPETISOFT v 0.2
<Nombre de la Empresa>

<Nombre del Proyecto>

---

<Nombre del proyecto>

Historia de revisiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<Detalles>	<Nombre>

Documento de Pruebas de base de datos	Página 1 de 6
---------------------------------------	---------------

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

TABLA DE CONTENIDO

1. INTRODUCCIÓN

1.1 Resumen

1.2 Definiciones

1.3 Referencias

2. AGENDA PROPUESTA

3. TIPOS DE PRUEBAS A LA BASE DE DATOS

4. ELABORACIÓN DEL PLAN DE PRUEBAS DE BASE DE DATOS

4.1 Casos de prueba

5. CONDICIONES

5.1 Condiciones de Entrada

5.2 Condiciones de Salida

6. REQUERIMIENTOS DEL ENTORNO

6.1 Hardware Base

6.2 Software Base

6.3 Herramientas para las pruebas

7. EJECUCIÓN DE LAS PRUEBAS A LA BASE DE DATOS

7.1 Personas y roles

7.2 Requerimientos de Capacitación

7.3 Ciclos

8. PRODUCTOS DE SALIDA

9. APROBADORES

Documento de Pruebas de Base de Datos	Página 2 de 6
---------------------------------------	---------------

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

1. INTRODUCCIÓN

*[Identifica qué va a ser probado, resume el enfoque general usado para las pruebas, los objetivos y metodologías. Veamos un ejemplo para la base de datos Hospital.]*

Este documento detalla las pruebas de la base de datos *Hospital* para garantizar la confiabilidad, transparencia e integridad de los datos. Para llevar a cabo las pruebas de la base de datos es necesario desarrollar un *Plan de Pruebas de Base de Datos*, que nos permita identificar que elementos de la base de datos deberán ser probados durante la ejecución de las pruebas.

1.1 Resumen

*[Describir a manera de resumen el objetivo del **Plan de Pruebas de Base de Datos**, de modo que se entienda lo que se quiere probar, es decir, conocer los datos de la base de datos que van a ser probados y definir las metodologías que se utilizarán para probar los datos. Veamos un ejemplo para la base de datos Hospital.]*

El objetivo de la prueba de la base de datos *Hospital* es obtener la certeza de que la base de datos obedece a las reglas del negocio y a las restricciones de integridad que se van a implementar en el entorno de producción. La certificación implica una comprobación completa de todos los elementos de la base de datos (tablas, atributos, interrelaciones, integridad, operaciones de consulta, así como restricciones semánticas) según los parámetros de prueba predefinidos.

1.2 Definiciones

*[Realizar un glosario de términos que se usen en este **Plan de Pruebas de Base de Datos** para que se tenga un entendimiento total por parte de todos los involucrados.]*

*Veamos un ejemplo para la base de datos Hospital.]*

Término	Descripción
Confiabilidad	Considera que la base de datos funcione correctamente cómo se estableció en un inicio, además de, proporcionar los datos a los usuarios que acceden la base de datos. Las funciones encargadas de proveer la confiabilidad deben incluir tanto el control físico cómo el técnico, para poder implementar una protección a los datos.
Control de concurrencia	Considera que las transacciones a los datos se puedan realizar sin afectar otras transacciones, aun cuando estas se efectúen al mismo tiempo.
Integridad	Se refiere a que las funciones que administran los datos garanticen un tratamiento sin errores. Es decir, que la base de datos debe conservar los datos fiables mientras se están realizando diferentes operaciones (transfiriendo o almacenando) en ella, con la seguridad de que los datos no podrán ser eliminados por los usuarios sin previa autorización.
Plan de Pruebas de Base de Datos	Permite especificar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas.

### 1.3 Referencias

*[Realizar un listado de todos los documentos a los cuales se hizo referencia para la aplicación de este plan, además de colocar su versión.]*

### 2. AGENDA PROPUESTA

*[Realizar un diagrama de actividades para el **Plan de Pruebas de Base de Datos** con el objeto de establecer la fecha de inicio y fin de las pruebas a los componentes de la base de datos. A fin de establecer dependencias entre actividades y análisis de riesgos en caso de*

retraso. Es importante mencionar que la agenda debe estar descrita de acuerdo al plan de actividades definido en la **Administración de Proyectos Específicos**.

Para realizar la agenda del **Plan de Pruebas de Base de Datos** se puede utilizar el Diagrama de Gantt, el cual nos permite visualizar las actividades y las fechas en que se deben realizar y terminar cada una de las pruebas. Veamos un ejemplo para diseñar la agenda de actividades para la realización de las pruebas de la base de datos Hospital.]

Base de Datos <i>Hospital</i>	Septiembre					Octubre	
	1sem	1sem	1sem	1sem	1sem	1sem	1sem
Definir los requerimientos de prueba de la base de datos	*						
Identificar los elementos de la base de datos a ser probados		*	*				
Planificación de las <i>Pruebas de la Base de Datos</i>			*	*			
Establecer los requisitos del laboratorio para cada tipo de prueba				*			
Planificación de los riesgos					*		



Base de Datos <i>Hospital</i>	Septiembre					Octubre	
Actividad/Tiempo	1sem	1sem	1sem	1sem	1sem	1sem	1sem
Identificación de las herramientas que se utilizarán para las pruebas de la base de datos					*	*	
Reporte de Validación de las <i>Pruebas de Base de datos</i>							*

### 3. TIPOS DE PRUEBAS A LA BASE DE DATOS

*[Describir la clase de pruebas que se realizarán a los componentes de base de datos.*

*Los tipos de pruebas pueden ser de desempeño, funcionalidad, integridad y de concurrencia.*

*Veamos un ejemplo para la base de datos Hospital.]*

- *Prueba de desempeño.* Verifica que las transacciones (accesos, operaciones de consulta) a la base de datos se realizan de manera eficiente. Como resultado de la prueba se obtiene el rendimiento de la base de datos en casos extremos de procesos. Veamos un ejemplo aplicando pruebas de desempeño a la base de datos *Hospital* evaluando las siguientes propiedades: atomicidad, consistencia, aislamiento, durabilidad.
  - Atomicidad: para esta prueba es necesario garantizar que la(s) operación(es) se realiza(n) o no se realiza(n), para el caso de la base de datos *Hospital* se probó realizar una transacción con dos operaciones la primera debía consultar los datos de los médicos que tenían pacientes en la planta 4, la segunda debía borrar los ingresos de los pacientes dados de alta en la planta 4. Como resultado de estas pruebas se observa que existe un fallo en la operación de borrar por lo que no se

efectúa se la consulta de los datos de los médicos que están atendiendo pacientes en la planta 4, es decir, que aunque la operación de consulta haya terminado su ejecución al fallar la operación de borrar toda la transacción se cancela.

- Consistencia: este tipo de prueba asegura que la base de datos permanece en un estado consistente antes y después de la transacción sea exitosa o no, para el caso de la base de datos *Hospital* se probó una transacción que eliminara los datos de los ingresos que se dieron entre las fechas 12/01/1999 y 31/12/1999. Como resultado la transacción eliminó los ingresos en las fechas mencionadas sin alterar los datos de los pacientes.
  - Aislamiento: se debe tomar en cuenta el tipo de aislamiento que se va a utilizar, en el caso de la base de datos *Hospital* se probó la seriabilidad con dos transacciones una debía consultar dos veces los datos de los pacientes que habían ingresado al *Hospital* en un rango de fecha de 01/09/2008 al 01/10/2008, la otra transacción debía insertar los pacientes que habían ingresado en la fecha 28/09/2008. Como resultado de estas pruebas se observa que la transacción 1 y la transacción 2 se ejecutan en forma concurrente, asimismo la transacción 1 regresa el mismo resultado, ya que la consulta se realiza antes o después de la inserción de los datos.
  - Durabilidad: esta prueba garantiza que una vez emitida la notificación de éxito la transacción persiste y no se desecha, en el caso de la base de datos *Hospital* se probó una transacción que debía insertar los datos de los pacientes al finalizar la transacción se debe apagar el equipo. Como resultado la transacción insertó los datos de los pacientes en la tabla Paciente aun cuando se apago el equipo ya que la transacción finalizó la consulta.
- *Prueba de funcionalidad.* Esta prueba permite garantizar que la base de datos realice las funciones para lo que fue creada, en el caso de la base de datos *Hospital* se probó insertar los datos de 7000000 de pacientes que se tenían guardados en un archivo. Como resultado se realizó la inserción de los datos de los pacientes en un tiempo de 15542ms, de esta manera podemos observar que la base de datos *Hospital* cumple con

la función de inserción de los datos.

- *Prueba de integridad.* Esta prueba verifica la integridad de los datos. Las pruebas deben verificar las reglas de integridad referencial, de entidad, de dominio, definida por el usuario. Veamos cada una de ellas utilizando la base de datos *Hospital*.
  - Integridad referencial: es posible imponer restricciones de integridad referencial, observando que los atributos añadidos a la tabla Ingreso provienen de las tablas Paciente y Médico, y sabemos que debemos imponerlas para tales atributos. En concreto, el valor del campo número de seguro social lo debemos encontrar en Paciente, así cómo el valor del campo Código de identificación del médico lo debemos encontrar en Médico y colocarlos en la tabla Ingreso. La siguiente Figura 3.3 muestra esta situación.

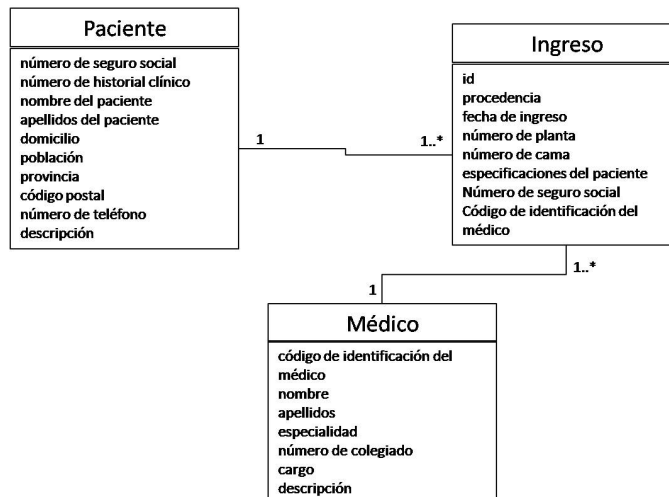


Figura 3.3: Ejemplo de integridad referencial de la base de datos *Hospital*.

En ella también se pueden observar las restricciones de cardinalidad una a varias que se han detectado entre esas entidades. (Con 1 se indica la parte una de la restricción y con 1..\* se indica la parte varias).

- Integridad de entidad: es posible imponer restricciones de integridad, observando que los atributos que formen parte de la llave primaria no puede contener valores

nulos. Para las tablas de Paciente, Ingreso y Médico, se realizaron las pruebas a los atributos de Número de historial clínico, Código de identificación del médico e ID respectivamente, el resultado obtenido fue que no permitían valores nulos.

- Integridad de dominio: los atributos pueden tener restricciones de dominio, colocando restricciones que impidan almacenar datos que violen la integridad de dominio. Por ejemplo, la regla de dominio del atributo fecha de ingreso puede ser dd/mm/aaaa (día/mes/año) y el texto "La fecha de ingreso del paciente debe ser en formato español (México)". La prueba que se realizó fue en el campo fecha de ingreso del paciente de la tabla Ingreso es introducir los datos de diferentes formas, lo que se obtuvo es que no introducía ningún valor hasta que se inserto el formato correcto.
- Integridad definida por el usuario: existe la posibilidad de que el usuario defina restricciones propias, cómo transacciones donde se definan bloqueos ó marcas de tiempo. Para el caso de la base de datos *Hospital* no se definieron ningún tipo de transacción con estas restricciones.
- *Prueba de concurrencia.* se debe tener el control de las transacciones que se ejecutan de manera simultánea, en el caso de la base de datos *Hospital* se probaron dos transacciones, la primera transacción debía consultar los datos de los pacientes que son atendidos por el médico 'Alfonso Zapata' y la segunda transacción debía actualizar los datos de los médicos que empiezan con la letra 'A'. Como resultado la transacción 1 devolvió los datos de los pacientes del médico 'Alfonso Zapata' ya con la actualización de los datos de los médicos que empiezan con la letra 'A' realizado por la transacción 2.

*Para la realización de todas las pruebas mencionadas anteriormente, se pueden realizar scripts de carga de datos que permitan generar los datos prueba diseñados para satisfacer los casos de prueba.*

#### 4. ELABORACIÓN DEL PLAN DE PRUEBAS DE BASE DE DATOS

##### 4.1 Casos de prueba

La siguiente tabla define algunos casos de prueba, los cuales se detallarán utilizando el siguiente formato, lo que permitirá tener un control de las pruebas que se desean realizar.

Caso de prueba	Entrada(s)	Resultado
CSPBD 1.1	Insertar datos en la tabla Paciente.	El SGBD deberá insertar los datos del Paciente, verificando cada uno de los datos introducidos.
CSPBD 2.1	Realizar varias transacciones en un mismo tiempo.	El SGBD deberá ejecutar las transacciones sin ningún problema y en caso de que alguna transacción no se cumpla cancelar la operación.
CSPBD 3.1	Realizar un borrado de datos en una tabla con restricciones de borrado de cascada.	El SGBD deberá evaluar el impacto que tiene el borrado de los datos y autorizar o denegar la consulta.

#### 5. CONDICIONES

*[Aquí debemos describir de manera general las condiciones esenciales de entrada y salida, que deben cumplirse para poder cubrir cada ciclo de pruebas.]*

Documento de Pruebas de Base de Datos	Página 3 de 6
---------------------------------------	---------------

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

##### 5.1 Condiciones de Entrada

*[Definir las condiciones necesarias que deben ser cubiertas para poder realizar las pruebas de los componentes de la base de datos. Veamos un ejemplo para la base de datos Hospital.]*

- La documentación de la base de datos *Hospital*. De manera específica, identificación de las herramientas para las pruebas.

#### 5.2 Condiciones de Salida

*[Se listan las condiciones que deben cumplirse para terminar las pruebas de los componentes de la base de datos. Veamos un ejemplo para la base de datos Hospital.]*

- Se generó un documento de registro de las actividades realizadas durante las pruebas de la base de datos *Hospital*.

- Se registraron de las pruebas realizadas según plan de pruebas de la base de datos *Hospital*.

### 6. REQUERIMIENTOS DEL ENTORNO

*[Se debe realizar en este punto un relación de los recursos no humanos que se requieren para las pruebas.]*

#### 6.1 Hardware Base

*[Presenta los requerimientos de Hardware necesarios para probar la base de datos. Se debe indicar versión. Veamos un ejemplo para la base de datos Hospital.]*

Para realizar las pruebas a la base de datos *Hospital* se debe tener cómo mínimo un hardware con la siguientes características un procesador a 800Mhz, 128 RAM, disco duro de 20Gb. En nuestro ejemplo se utilizó una laptop con procesador INTEL CORE2 a 1.66Ghz, 1Gb de RAM, disco duro de 65Gb.

#### 6.2 Software Base

*[Presenta los requerimientos de software necesarios para probar la base de datos. Se debe indicar versión. Veamos un ejemplo para la base de datos Hospital.]*

Para la realización de las pruebas se debe tener un sistema operativo Windows XP o Linux, un SGBD Postgresql 8 o superior. El software donde se encuentra la base de datos *Hospital* es un sistema operativo WINDOWS XP, SGBD Postgresql 8.2.

#### 6.3 Herramientas para las pruebas

*[Presenta las herramientas necesarias para probar la base de datos. Veamos un ejemplo para la base de datos Hospital.]*

Se utilizó una plantilla en formato .doc con nombre Lista de verificación de la base de datos.doc, para efectuar las pruebas correspondientes a la base de datos *Hospital*. Se utilizó el pgAdminIII para crear scripts de consultas para insertar, borrar, modificar y consultar los datos de la base de datos Hospital. Se crearon transacciones que permitieran efectuar dos operaciones al mismo tiempo sobre la tabla Paciente.

## 7. EJECUCIÓN DE LAS PRUEBAS A LA BASE DE DATOS

*[Se describen las actividades de pruebas de la base de datos que serán realizadas según el **Plan de Pruebas de Base de datos**. Es importante definir la forma en que serán manejadas las pruebas de los datos, las personas y los roles que participarán en la realización de las pruebas, la capacitación, el seguimiento de defectos y los ciclos que se necesitan para el desarrollo de las pruebas.]*

### 7.1 Personas y roles

*[Se describen los roles que existirán para la realización de las actividades del punto 6. Para cada rol, se identifica qué persona va a cumplir el rol y cuales son las responsabilidades, junto con la información de contacto. Veamos un ejemplo para la base de datos Hospital.]*

- Los roles encargados de realizar el *Plan de Pruebas de Base de Datos* del *Hospital* son: *Responsable de Pruebas* y el *Administrador de la Base de Datos*.

- El rol encargado de realizar las pruebas a la base de datos *Hospital* es el *Responsable de Pruebas*

### 7.2 Requerimientos de Capacitación

*[Definir los requerimientos de capacitación mínimos para realizar las pruebas de la base de datos. Veamos un ejemplo para la base de datos Hospital.]*

El *Responsable de Pruebas* revisó la documentación de la base de datos *Hospital*, además de realizar pruebas unitarias a cada elemento que se estableció para ser probado.

### 7.3 Ciclos

*[Definir cuantos ciclos de prueba se realizarán, fechas de comienzo y fin de cada ciclo, nombre de la base de datos que se probará en ese ciclo, etc. Veamos un ejemplo para*

*la base de datos Hospital.]*

Para la base de datos *Hospital* se definió sólo un ciclo debido a que es una base de datos pequeña puesto que se centra en el control de ingresos de los pacientes y el médico que se encarga de atenderlos. La fecha de inicio que se estableció fue el 01 de septiembre de 2008 y la fecha de fin fue el 11 de octubre de 2008.

Documento de Pruebas de Base de Datos	Página 5 de 6
---------------------------------------	---------------

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

## 8. PRODUCTOS DE SALIDA

*[Se debe realizar una recopilación de todos los productos generados durante la definición del **Plan de Pruebas de Base de Datos** y su desarrollo. Por ejemplo:]*

*Plan de Pruebas de Base de Datos*

*Reporte de las pruebas realizadas, con fecha de inicio y fin de cada una*

*Registro de defectos encontrados en la realización de las pruebas*

*Registro de la capacitación para la realización de las pruebas a la base de datos*

*Reporte de Verificación*

*Reporte de Validación*

## 9. APROBADORES

*[Se identifican las personas que aprobarán este documento. Veamos un ejemplo para la base de datos Hospital.]*

Documento de Pruebas de Base de Datos	Página 6 de 6
---------------------------------------	---------------

La lista de verificación de la estructura de la base de datos nos confirma que el esquema de conceptualización de la base de datos corresponda con el dominio del problema de la empresa u organización, lo que permitirá establecer un nivel de calidad en los componentes de base de datos por medio de un plan de pruebas de base de datos. A continuación se muestra la correspondiente lista de verificación de la base de datos.

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---



Tipo de Artefacto: Base de Datos Nombre del Sistema A Revisar: Nombre del Artefacto a Revisar: <p style="text-align: center;">EVAL = Cumple; No cumple</p>
---

CÓDIGO	DESCRIPCIÓN	EVAL C/N
PRU-BD-01	En la explicación del documento se definen claramente los elementos que se deben tomar en cuenta para la realización de las pruebas.	
PRU-BD-02	En el propósito, se realizó una selección de estados de prueba y tipos de prueba a ejecutar acorde con los requisitos de la base de datos. Se define que pruebas no se van a ejecutar.	
PRU-BD-03	Se especifican las personas a quienes está dirigido el documento.	
PRU-BD-04	Las metas descritas están relacionadas con los criterios de aceptación que el cliente presentó al inicio del proyecto.	
PRU-BD-05	Se definen las pruebas que se van a ejecutar dependiendo de las circunstancias.	
PRU-BD-06	Se define la metodología que se debe utilizar para las pruebas establecidas, dentro de esta se definen los objetivos y propósitos acorde con los requisitos de la base de datos.	
PRU-BD-07	Se describe el hardware, sistemas operativos y SGBD requerido para la ejecución de las pruebas y son consistentes con los recursos del ambiente de producción descritos en el plan del proyecto.	

CÓDIGO	DESCRIPCIÓN	EVAL C/N
PRU-BD-08	Se describe el recurso humano necesario y a cada uno de estos se le asigna responsabilidades y compromisos de acuerdo al rol.	
PRU-BD-09	Se tiene detallado un cronograma de actividades con fecha de realización y responsable para cada una de las actividades involucradas en la planeación, diseño, implementación, ejecución y evaluación del proceso de pruebas.	
Observaciones		
Fecha	Elaborado por	Revisado por

El siguiente componente de base de datos se integra en las fases de *Análisis y Diseño*. En estas fases se integra una guía y una plantilla que permitirán incorporar los productos de *Esquema Lógico Estándar*, *Esquema Lógico Específico*, *Esquema Físico* de la base de datos, además del *Diccionario de Datos* que permitirá al programador conocer los datos, las interrelaciones y restricciones que existen entre ellos. Asimismo, se genera una lista de verificación que permite conocer si la base de datos cumple o no con las necesidades de la empresa u organización.

A continuación se muestra la guía referente al *Análisis y Diseño* de la base de datos.

**Guía de Implantación para OPE.3 - A3. Realización de la fase de Análisis y Diseño**

Proceso	OPE.2 Desarrollo y Mantenimiento de Software. - A2. Realización de la fase de Análisis y Diseño
Actividad (es)	A3.2. Documentar o Modificar el <i>Análisis y Diseño</i> :

Tarea(s) Detalle	<ul style="list-style-type: none"> <li>- Generar el <i>Esquema Lógico Estándar</i> a partir del <i>Esquema Conceptual</i>, de manera que se obtenga un modelo de datos muy parecido al SMBD que se va a utilizar, pero sin las restricciones de éste, usando técnicas de normalización para definir este esquema.</li> <li>- Generar el <i>Esquema Lógico Específico</i> a partir del <i>Esquema Lógico Estándar</i>, éste deberá ser descrito en el lenguaje de definición de datos.</li> <li>- Generar el <i>Esquema Físico</i> a partir del <i>Esquema Lógico Específico</i> teniendo en cuenta los procesos, características del SMBD, del sistema operativo, y del hardware, lo que nos llevará a cumplir los objetivos del sistema y conseguir optimizar los tiempos de respuesta.</li> <li>- Generar el <i>Diccionario de Datos</i> a partir del <i>Esquema Lógico Específico</i>, con el fin de reunir la información de los datos almacenados con base en la estructura de los datos, sus interrelaciones y restricciones.</li> </ul>
Técnicas disponibles	<p>Algunas de las técnicas para el <i>Análisis y Diseño de la Base de Datos</i> son: simplificación del esquema, reglas de integridad, dependencias funcionales, normalización, definición de campos, definición de interrelaciones, operaciones de tablas.</p>
Productos específicos generados	<ul style="list-style-type: none"> <li>- <i>Modelo de Base de Datos</i></li> <li>- <i>Esquema Lógico Estándar</i></li> <li>- <i>Esquema Lógico Específico</i></li> <li>- <i>Esquema Físico</i></li> <li>- <i>Diccionario de Datos</i></li> </ul>
Herramientas de apoyo	<p>* Modelo relacional:</p> <ul style="list-style-type: none"> <li>- Permite representar el esquema relacional de una manera sencilla y completa,</li> <li>- esta compuesto por un conjunto de nodos multiparticionados, donde cada nodo representa un esquema de relación,</li> <li>- indica el nombre, sus atributos, la llave primaria y sus llave foránea.</li> </ul>

	<p>* Dependencias funcionales:</p> <ul style="list-style-type: none"> <li>- Permita definir las propiedades del contenido semántico de los datos,</li> <li>- debe simplificar el esquema del dominio del problema, sin perder la generalidad.</li> </ul> <p>* Normalización:</p> <ul style="list-style-type: none"> <li>- Debe ser una herramienta que consiga la formalización del diseño lógico de la base de datos,</li> <li>- permita la utilización de métodos algorítmicos para auxiliar al diseño de la base de datos,</li> <li>- incluir el desarrollo de programas que agilicen el diseño de la base de datos.</li> </ul>
Criterio de selección	<ul style="list-style-type: none"> <li>- Modelo Relacional: Es una técnica que se basa en la teoría de las relaciones, la cual proporciona que los datos tengan una estructura natural, independiente del SMBD y de la máquina que se utilicen. Así pues, el modelo tiene independencia física, lógica, además de, ser flexible, uniforme y sencillo.</li> <li>- Modelo Relacional Orientado a Objetos: Es una extensión de la técnica utilizada en el modelo relacional, que incluye además la teoría del modelo de objetos. La unión de estas dos técnicas permite representar sistemas de objetos que tienen una gran cantidad de interrelaciones complejas para obtener una mejor estructura de los datos de la empresa u organización, manteniendo la independencia del SGBD y de la máquina. Asimismo, adiciona las siguientes características; extensibilidad, poder de expresión, reusabilidad e integración.</li> </ul>
Soportes	<ul style="list-style-type: none"> <li>- Plantilla de Análisis y Diseño extensión Base de Datos</li> <li>- Lista de verificación del modelo relacional</li> </ul>

Bibliografía	- Piattini, Mario. Marcos, Esperanza. Calero, Coral. Vela, Belén. "Tecnología y diseño de base de datos", Alfaomega, Ra - ma. 2007.
--------------	---

***Plantilla de Análisis y Diseño extensión Base de Datos***

En la plantilla de *Análisis y Diseño extensión Base de Datos*, se definirán los esquemas lógico y físico de la base de datos que permitan definir la estructura de la base de datos utilizando cómo base el *Esquema Conceptual* definido en la fase de *Especificación de Requerimientos*. Con base en lo anterior, veamos la plantilla definida para realizar el *Análisis y Diseño* de la base de datos.

---

COMPETISOFT v 0.2	<Nombre de la Empresa>
<Nombre del Proyecto>	

---

<Nombre del proyecto>

Análisis y Diseño extensión Base de Datos

Historia de revisiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<Detalles>	<Nombre>

Documento de Análisis y Diseño extensión Base de Datos	Página 1 de 7
--	---------------

---

COMPETISOFT v 0.2	<Nombre de la Empresa>
<Nombre del Proyecto>	

---

TABLA DE CONTENIDO

1. DESCRIPCIÓN DEL CONTENIDO
2. OBJETIVOS DEL MODELO DE LA BASE DE DATOS
3. DESARROLLO DEL MODELO DE LA BASE DE DATOS
  - 3.1 Esquema Lógico Estándar

3.2 Esquema Lógico Específico

3.3 Esquema Físico

3.4 Diccionario de Datos

4. GLOSARIO DE TÉRMINOS

5. PRODUCTOS DE SALIDA

6. APROBADORES

Documento de Análisis y Diseño extensión	Página 2 de 7
Base de Datos	

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

## 1. DESCRIPCIÓN DEL CONTENIDO

*[Describir el **Modelo de la Base de Datos** que se utilizará para la definición de los esquemas propios del **Análisis y Diseño** de la base de datos. El modelo debe utilizar cómo base el **Esquema Conceptual** definido en la fase de **Especificación de Requerimientos** para detallar el manejo de los datos. Asimismo, se deben definir cada uno de los esquemas (**Lógico Estándar, Lógico Específico y Físico**), que permitirán el diseño de la estructura de la base de datos, es necesario incluir la descripción del **Diccionario de Datos** que servirá para mostrar las características de los datos y sus restricciones.]*

## 2. OBJETIVOS DEL MODELO DE LA BASE DE DATOS

*[Definir lo que se debe generar al terminar el **Análisis y Diseño** de la base de datos para que sirva cómo base para la fase de **Construcción** de la base de datos. Veamos un ejemplo de que es lo que debería llevar este punto:*

- *Diseñar el **Esquema Lógico Estándar** para ir minimizando redundancia, lograr la máxima simplicidad y evitar cargas no necesarias.*
- *Diseñar el **Esquema Lógico Específico** a partir del **Esquema Lógico Estándar**, utilizando el lenguaje de definición de datos (LDD) cómo instrumento para su descripción.*

- Definir el Sistema Gestor de Base de Datos (SGBD) que servirá para la definición de la estructura de la base de datos.
- Diseñar **Esquema Físico** con base en el **Esquema Lógico Específico**, para la construcción de la **Estructura Física de la Base de Datos**.
- Diseñar el Diccionario de Datos, para identificar las características de los atributos, sus interrelaciones y sus restricciones semánticas, en base al SGBD definido para la carga de la base de datos.
- Definir los ciclos de **Análisis y Diseño** de la base de datos que se necesitarán para concluir correctamente la fase.]

### 3. DESARROLLO DEL MODELO DE LA BASE DE DATOS

[Diseñar los esquemas del **Modelo de la Base de Datos** que servirán como base para la fase de **Construcción**.]

3.1 Esquema Lógico Estándar [En este paso debemos convertir el Esquema Conceptual al Esquema Lógico Estándar. Asimismo, para poder realizar esta conversión debemos tener en cuenta las siguientes reglas:

- La entidad definida en el modelo E/R debe ser transformada en una relación.
- Las interrelaciones que tengan cardinalidad muchos a muchos deberá transformarse en una relación.
- Las interrelaciones que tengan cardinalidad uno a muchos deberán utilizar la propagación de claves o crear una nueva relación.]

Veamos el ejemplo que hemos estado trabajando para transformar el modelo E/R al modelo relacional, utilizando las reglas definidas anteriormente, Figura 3.4 <sup>1</sup>.

Documento de Análisis y Diseño extensión	Página 3 de 7
Base de Datos	

---

<sup>1</sup>El diagrama esta inspirado en el diagrama grafos de Piattini, M. (2007) Tecnologías y diseño de Bases de Datos

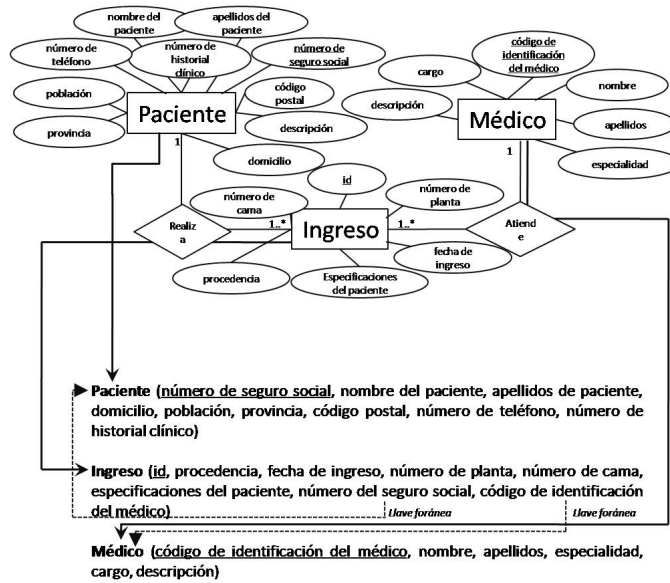


Figura 3.4: Paso del ME/R al modelo relacional

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

Veamos detalladamente esta transformación del ME/R al modelo relacional (MR), describiendo por separado cada una de las transformaciones:

- Transformación de dominios: el dominio Figura 3.5 es un objeto que debe ser definido en el LDD propio del SGBD que estemos utilizando, por ejemplo:

ME/R



Figura 3.5: Dominio

MR

DOMINIO CÓDIGO-POSTAL CHAR(5)



- Transformación de entidades: el conjunto de entidades descrito en el ME/R se debe convertir en una relación en el MR. Por ejemplo:

**El conjunto de entidades PACIENTE se debe transformar en una relación (tabla), con el mismo nombre.**

- Transformación de atributos de entidades: Los atributos definidos en el ME/R Figura 3.6 de cada conjunto de entidad, se transforman en columnas de la relación a la que ha dado lugar la entidad. Es necesario tener en cuenta que los atributos pueden ser identificadores principales, identificadores alternativos y atributos no identificadores. Por ejemplo:

**ME/R**



Figura 3.6: Entidad con atributos

**MR**

**PACIENTE (Número de Seguridad Social, Nombre del paciente, Apellidos del paciente, Domicilio, Población, Provincia, Código postal, Número de teléfono, Número de historial clínico, Descripción)**

- Transformación de interrelaciones: Esta transformación se da dependiendo del tipo de la cardinalidad de la interrelación Figura 3.7. Para el ejemplo de la base de datos *Hospital*, tenemos interrelaciones 1:N, veamos cómo se realiza la transformación Figura 3.8.

**ME/R**

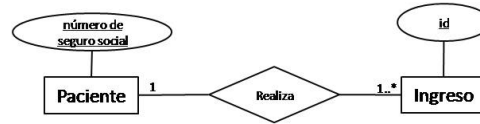


Figura 3.7: Diagrama Entidad/Relación

**MR**

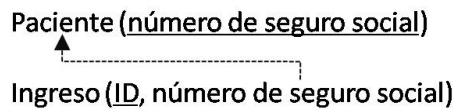


Figura 3.8: Modelo Relacional

- Transformación de atributos de interrelaciones: Esta transformación se realiza sólo cuando la interrelación se convierta en relación. Si es el caso los atributos de la interrelación pasan a ser columnas de la relación obtenida. **Para el ejemplo del *Hospital*, no existen interrelaciones que se deban transformar en tablas.**
- Transformación de restricciones: Para las restricciones de usuario existen cláusulas que permiten restringir rangos de valores, determinar los valores que se pueden desplegar, incluso se pueden verificar ciertas condiciones que deben cumplir un conjunto de atributos de la relación.

Documento de Análisis y Diseño extensión	Página 4 de 7
Base de Datos	

### 3.2 Esquema Lógico Específico

*[En este paso debemos convertir el **Esquema Lógico Estándar** al **Esquema Lógico Específico**, el cual será descrito en el LDD del SGBD que se va a utilizar.]*

### 3.3 Esquema Físico

*[En este paso debemos diseñar el **Esquema Físico** de la base de datos. Para llevar a cabo este proceso debemos utilizar el **Esquema Lógico Específico** que servirá como base para elaborar el **Esquema Físico** de la base de datos. Para diseñar el Esquema Físico de la base de datos existen varios elementos que debemos considerar, veamos la utilización de cada uno de ellos:]*

- Diseño de bloques: Este diseño sólo se puede realizar si el SGBD lo permite. Si es el caso, podemos definir el espacio libre de cada bloque, el porcentaje de utilización de cada bloque, el número de bloques asignados a cada tabla. Por ejemplo:

```
CREATE TABLE Paciente
numero-historial-clínico NUMBER (4) NOT NULL,
(numero-seguro-social NUMBER (4),
nombre-paciente      VARCHAR(25) NOT NULL,
apellidos-paciente   VARCHAR(25) NOT NULL,
domicilio            VARCHAR(50) NOT NULL,
poblacion            VARCHAR(25) NOT NULL,
provincia            VARCHAR(25) NOT NULL,
código-postal        VARCHAR(5) NOT NULL,
numero-telefono      VARCHAR(8),
descripción          VARCHAR(100),
PRIMARY KEY (Numero-seguro-social),
TABLESPACE Datos1
(INITIAL 100K
NEXT 200K ;
```

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

- **Diseño de la organización:** Este sirve para definir cómo realizar la organización de los archivos para cada tabla, de la base de datos. Asimismo, nos indica la forma de acceder a los datos de la base de datos. Algunos de los criterios utilizados para la organización de ficheros son: Árbol B+ y Hash.
- **Utilización de índices:** El utilizar índices permite optimizar los tiempos de respuesta de las consultas a los atributos indizados, pero disminuye el rendimiento de la base de datos, debido a la actualización que se deba hacer al índice. Veamos el ejemplo que podemos utilizar para definir índices en nuestra base de datos *Hospital*.

```
CREATE INDEX Ind-Paciente ON PACIENTE (número-seguro-social)
```

Así pues, es bueno crear índices de las tablas cuando las consultas a ellas son frecuentes. Por otro lado existen numerosas estrategias de optimización de consultas o procesamientos, cómo son: candados, nivel de segmento, nivel de tabla, nivel de página, etc., éstos permiten incrementar el nivel de desempeño de las tablas indizadas.

- **Agrupamiento de tablas (Clusters):** Esta técnica permite agrupar tablas donde sus filas comparten un grupo de atributos. El uso de esta técnica permite agilizar el tiempo de respuesta de la combinación de tablas, pero se complica el procesamiento de consultas y actualizaciones a las tablas por separado. Para poder implementar esta técnica es necesario que el SGBD la soporte de lo contrario no se podrá realizar, por tal motivo, no se maneja ningún ejemplo, debido a que el Esquema Físico debe tratarse de forma que se pueda utilizar en cualquier circunstancia.
- **Redundancia de datos:** Se debe minimizar la redundancia de datos, evitando posibilidades de inconsistencia de los datos. El uso de disparadores que controlen la inconsistencia derivada de la redundancia.

### 3.4 Diccionario de Datos

*Sirve para establecer y tener un esquema de la **Estructura Física de la Base de Datos**, es importante diseñar en este paso el **Diccionario de Datos** para evitar conflictos de operaciones con los datos, además de mantener un conocimiento de la manera en cómo se encuentra el esquema de la base de datos. A continuación se presenta una plantilla de cómo se puede construir un **Diccionario de Datos**.*

Nombre	Tipo de dato	Tamaño	Tipo de llave	Descripción
Se debe escribir el nombre del campo.	Especificar el tipo de dato que almacenará el campo.	Es necesario especificar la cantidad de caracteres que almacenará el campo.	Si el campo es llave es necesario establecer de que tipo es (primaria, foránea).	Se da una breve explicación de la función del campo y lo que almacena.

Documento de Análisis y Diseño extensión Base de Datos	Página 6 de 7
---	---------------

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

*Continuación del esquema del **Diccionario de Datos**.*

Autoincremento	Constraints	Campos con los que se relaciona	Indexado
Se puede o no establecer que el campo se incremente de manera automática, sólo puede ser utilizado en campos de tipo número.	Si los campos tienen interrelaciones específicas en esta sección se deberá establecer ese tipo de relación y la operación que realizará por ejemplo, (unique, not null, check u otro).	Es importante conocer los campos que forman parte de la relación.	Se debe colocar que campos formarán el índice de la tabla para establecer su almacenamiento.

*Veamos un ejemplo del esquema del **Diccionario de Datos** para la tabla *Paciente* de la base de datos *Hospital*.*

Nombre	Tipo de dato	Tamaño	Tipo de llave	Descripción
número-historial-clínico	number	4	PRIMARY KEY	Identifica a la tabla Paciente. Es la llave primaria.
número-seguro-social	number	4		Almacena el número de seguro social del paciente.
nombre-paciente	varchar	25		Almacena el nombre(s) del paciente.
apellidos-paciente	varchar	25		Almacena los apellidos del paciente.
domicilio	varchar	50		Almacena el lugar donde vive el paciente.
población	varchar	25		Almacena la colonia donde vive el paciente.
provincia	varchar	25		Almacena el estado o provincia.
código-postal	varchar	5		Almacena el código postal.
número-teléfono	varchar	8		Almacena el teléfono.
descripción	varchar	100		Almacena las observaciones que puedan tener del paciente.
ID	number	4	FOREING KEY	Permite relacionar la tabla Ingreso con la tabla Paciente.

*Continuación del esquema del **Diccionario de Datos** de la tabla Paciente.*

Autoincremento	Constraints	Campos con los que se relaciona	Indexado
SI	NOT NULL		NO
NO	NOT NULL		SI
NO	NOT NULL		NO
NO	NOT NULL		NO
NO	NOT NULL		NO
NO	NOT NULL		NO

Autoincremento	Constraints	Campos con los que se relaciona	Indexado
NO	NOT NULL		NO
NO	NOT NULL		NO
NO	NULL		NO
NO	NULL		NO
NO	NOT NULL	ID de la tabla Ingreso	NO

#### 4. GLOSARIO DE TÉRMINOS

*[En esta sección se relacionan los términos propios del contexto y su significado. Este glosario sirve como vocabulario de referencia de todos los involucrados en el proyecto.]*

#### 5. PRODUCTOS DE SALIDA

*En esta sección se deben colocar todos los productos generados durante la fase de **Análisis y Diseño** de la base de datos con el fin de tener el control de los documentos, para que puedan ser integrados en la **Configuración de Software**. Los documentos generados son:*

- *Esquema Lógico Estándar*
- *Esquema Lógico Específico*
- *Esquema Físico*
- *Diccionario de Datos*
- *Reporte de Verificación y Validación de los productos generados*

#### 6. APROBADORES

*[Se identifican las personas que aprobarán este documento]*

Documento de Análisis y Diseño extensión	Página 7 de 7
Base de Datos	

La lista de verificación correspondiente al Análisis y Diseño de la base de datos, nos confirma que el desarrollo de los esquemas propios del Análisis y Diseño de la base de datos han sido realizados de manera correcta. A manera de ejemplo se presenta la lista de verificación para el Modelo Relacional.

<p>Tipo de Artefacto: Modelo Relacional</p> <p>Nombre del Sistema A Revisar:</p> <p>Nombre del Artefacto a Revisar:</p> <p style="text-align: center;">EVAL = <b>C</b>umple; <b>N</b>o cumple</p>
---

CÓDIGO	DESCRIPCIÓN	EVAL C/N
AD-MR-01	El diagrama Entidad/Interrelación se transformo al modelo relacional conteniendo relaciones (tablas), atributos y dominios.	
AD-MR-02	Se define un esquema de relación que se compone de un nombre de relación por ejemplo Paciente, de un conjunto de n atributos número de seguro social y de un conjunto de n dominios número de seguro social int, donde cada atributo debe ser definido por su dominio.	
AD-MR-03	La relación se representa utilizando una tabla donde exista grado (columnas de la tabla) y cardinalidad (filas de la tabla).	
AD-MR-04	El grado corresponde al número de atributos de la relación por ejemplo Paciente.	
AD-MR-05	Se establece la cardinalidad de la relación por ejemplo, para Paciente 1 a N	
AD-MR-06	Un ejemplo de dominio por extensión podría ser Materias donde el conjunto de valores podría ser: "Base de Datos", "Sistemas Operativos", "Lenguajes, Ingeniería de Software", etc.	
AD-MR-07	En ejemplo de dominio por intensión podría ser código postal, cómo cadena de 5 caracteres.	



CÓDIGO	DESCRIPCIÓN	EVAL C/N
AD-MR-08	Se establece la integridad de entidad la cual dice que todas las tuplas deben ser distintas. La cual establece que ningún atributo que forme parte de la clave primaria puede tomar valores nulos.	
AD-MR-09	Se describen las restricciones con precisión y sencillez.	
AD-MR-10	Se indican las acciones a seguir ante una posible violación de una restricción.	
AD-MR-11	Se verifica la coherencia de las restricciones entre sí mismas.	

Observaciones
---------------

Fecha	Elaborado por	Revisado por

Por último se integra un componente para realizar las pruebas de integración de la base de datos, en ella se describe una guía para realizar el plan de pruebas de integración de la base de datos en base a un plantilla que nos proporciona la base de conocimientos para documentar y realizar el plan de pruebas de integración de la base de datos. Asimismo, se tiene una lista de verificación para el modelo relacional orientado a objetos que permitirá establecer el cumplimiento de la integración de la base de datos con la interfaz de usuario. A continuación se muestran estos componentes.

**Guía de Implantación para OPE.3 - A3 Realización de la fase de Análisis y Diseño**

Proceso	OPE.2 Desarrollo y Mantenimiento de Software. A3 Realización de la Fase de Análisis y Diseño
---------	--

Actividad (es) / Tarea(s)	A3.7. Elaborar o modificar el <i>Plan de Pruebas de Integración</i> incluyendo el <i>Plan de Pruebas de Integración de Base de Datos</i> .
Técnicas disponibles	Definir <i>Plan de Pruebas de Integración de Base de Datos</i> con base en los componentes del sistema y de la base de datos generados.
Producto específico generado	<i>Plan de Pruebas de Integración de Base de Datos</i>
Herramientas de apoyo	Procesador de texto, base de datos
Criterio de selección	Para definir el <i>Plan de Pruebas de Integración de Base de Datos</i> se debe establecer quién, cuándo, dónde y cómo se realizarán las actividades para integrar la base de datos con la interfaz de usuario. El definir un <i>Plan de Pruebas de Integración de Base de Datos</i> permitirá al equipo de trabajo establecer mecanismos bien definidos para realizar una integración transparente y así otorgar un sistema funcional y

	<p>claro para el cliente. Los puntos principales del <i>Plan de Pruebas de Integración de Base de Datos</i> son:</p> <ol style="list-style-type: none"> <li>1. Introducción</li> <li>2. Planificación de las Pruebas de Integración de Base de Datos</li> <li>3. Diseño de las Pruebas de Integración de Base de Datos</li> <li>4. Exigencias del Ambiente</li> <li>5. Pruebas no funcionales para la Integración de la Base de Datos</li> <li>6. Casos de Prueba de Integración de Base de Datos</li> <li>7. Personas y roles</li> <li>8. Productos de Base de Datos</li> <li>9. Aprobadores</li> </ol>
Soportes	- Plantilla del Plan de Pruebas de Integración de Base de Datos.
Bibliografía	-

***Plantilla de Plan de pruebas de integración de la base de datos***

La plantilla de *Plan de Pruebas de Integración de Base de Datos*, nos permite definir las diferentes pruebas de integración que se deben realizar con la base de datos, para lograr que se conserve la consistencia de los datos, además de agilizar la integración de los componentes de software con la base de datos, sin perder las restricciones establecidas en la base de datos. Esto con la intención de cumplir con las necesidades establecidas en el dominio del problema de la empresa u organización.

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

<Nombre del proyecto>

Plan de Pruebas de Integración de Base de Datos

Historia de revisiones

Fecha	Versión	Descripción	Autor
<dd/mmm/yy>	<x.x>	<Detalles>	<Nombre>

Documento de Plan de Pruebas de Integración de Base de Datos

Página 1 de 7

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

#### TABLA DE CONTENIDO

##### 1. INTRODUCCIÓN

###### 1.1 Objetivo

##### 2. PLANIFICACIÓN DE LAS PRUEBAS DE INTEGRACIÓN DE BASE DE DATOS

##### 3. DISEÑO DE LAS PRUEBAS DE INTEGRACIÓN DE BASE DE DATOS

###### 3.1 Base de datos

###### 3.2 Visualizador de la base de datos

###### 3.3 Sincronización de datos

###### 3.4 Integración funcional

##### 4. EXIGENCIAS DEL AMBIENTE

###### 4.1 Hardware

###### 4.2 Software

###### 4.3 Sistema Operativo

###### 4.4 Sistema Gestor de Base de Datos

##### 5. PRUEBAS NO FUNCIONALES PARA LA INTEGRACIÓN DE LA BASE DE DATOS

###### 5.1 Pruebas de desempeño

###### 5.2 Pruebas de seguridad

5.3 Pruebas de concurrencia

6. CASOS DE PRUEBAS DE INTEGRACIÓN DE BASE DE DATOS

7. PERSONAS Y ROLES

8. PRODUCTOS DE BASE DE DATOS

9. APROBADORES

Documento de Plan de Pruebas de Integración de Base de Datos
--

Página 2 de 7
---------------

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

## 1. INTRODUCCIÓN

*[Describir el interés de realizar el **Plan de Pruebas de Integración de Base de Datos**, así cómo, su objetivo. Veamos un ejemplo para la base de datos Hospital.]*

El principal interés de realizar las *Pruebas de Integración de Base de Datos* se da con la intención de validar la estructura física de la base de datos *Hospital* con el fin de garantizar que no exista conflicto con los tipos de datos y sus longitudes. Asimismo, es necesario realizar la validación de las operaciones de consultas sobre los datos. Otro punto importante es la validación de la sincronización correcta de los datos.

Por lo anterior este documento detalla los elementos que debemos considerar para poder realizar la integración correcta de la base de datos con la aplicación.

### 1.1 Objetivo

*[Definir el objetivo del **Plan de Pruebas de Integración de Base de Datos**. Veamos un ejemplo para la base de datos Hospital.]*

El objetivo es garantizar la estabilidad de los elementos de la base de datos *Hospital* y el cumplimiento de los requerimientos propuestos en la fase de *Análisis y Diseño*.

## 2. PLANIFICACIÓN DE LAS PRUEBAS DE INTEGRACIÓN DE BASE DE DATOS

*[Se debe generar un calendario de actividades que contenga cómo puntos principales los objetivos de las pruebas de integración de base de datos, así cómo, las fechas de inicio y*

de fin de cada una de las actividades que se realizaron. Veamos un ejemplo para realizar la planificación de las **Pruebas de Integración de Base de Datos**.

Base de Datos <i>Hospital</i>	Octubre					Noviembre	
	1sem	1sem	1sem	1sem	1sem	1sem	1sem
Identificar los requerimientos de integración de la base de datos <i>Hospital</i>	*						
Definir los elementos de la base de datos <i>Hospital</i> que se integrarán		*	*				
Realizar el <i>Plan de Pruebas de Integración de Base de Datos</i> para el <i>Hospital</i>			*	*			
Definir el equipo de pruebas para la integración de la base de datos <i>Hospital</i>				*			
Establecer la plataforma de pruebas					*		
Generar el reporte de hallazgos					*	*	

Base de Datos <i>Hospital</i>	Octubre					Noviembre	
Actividad/Tiempo	1sem	1sem	1sem	1sem	1sem	1sem	1sem
Reporte de <i>Verificación y Validación</i> de las pruebas de integración							*
Documento de Plan de Pruebas de Integración de Base de Datos					Página 3 de 7		

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

### 3. DISEÑO DE LAS PRUEBAS DE INTEGRACIÓN DE BASE DE DATOS

#### 3.1 Base de datos

*[Se realiza validación de la estructura de la base de datos para garantizar que no exista conflicto con los tipos de datos y longitudes de los mismos. La base de datos debe soportar la estructura de los datos fuentes y destinos para garantizar su correcta operación. Además, debe considerar que los destinos pueden ser múltiples aplicaciones. Veamos un ejemplo para la base de datos Hospital.]*

En el caso de la base de datos *Hospital*, podemos considerar que los datos son de tipo numérico y caracter, con longitudes no mayores a 100, por tanto cumple con los requerimientos de cualquier interfaz de usuario, pues los datos que se recibirán de la interfaz hacia la base de datos son enteros y cadenas los cuales cubren con lo estipulado en la estructura de la base de datos. Aun cuando se pueda establecer los tipos de datos, éstos pueden variar debido a que los SGBD soportan diferentes set de caracteres y varían dependiendo del lenguaje (ingles, japonés, unicode, etc.).

#### 3.2 Visualizador de la base de datos

*[La base de datos varía de proyecto en proyecto, por lo cual se requiere validar que se puedan realizar consultas y reportes sobre éste. Veamos un ejemplo para la base de datos*

*Hospital.]*

Si deseamos consultar el nombre del paciente que está siendo atendido por el médico con Código de identificación del médico 005 desde la interfaz de usuario, el SGBD deberá realizar la transacción de consulta regresando el nombre del paciente. Es importante mencionar que se debe tener conocimiento de la estructura física de la base de datos para realizar estas pruebas.

### 3.3 Sincronización de datos

*[Las pruebas van orientadas a realizar validación de la sincronización de datos, es decir que la interacción de los datos básicos comunes entre aplicaciones se cumpla adecuadamente. Los mecanismos empleados son muy variados y van desde APIs hasta mediadores que permiten configuración flexible de la información. Veamos un ejemplo para la base de datos Hospital.]*

Para la base de datos *Hospital* se definió el tipo de conexión y su tipo de acceso. Por tanto, cuando se realizó la prueba de inserción de datos desde la interface a la tabla Paciente se tuvo un resultado favorable, ya que la conexión que se estableció fue la correcta.

### 3.4 Integración funcional

*[Se debe validar la sincronización de los datos a nivel de procesos, es decir la interacción entre los productos a partir de la ejecución de procesos y la transferencia de los datos a otros procesos. Debemos identificar los procesos que deben realizarse cuando se ejecute una transacción y el mecanismo de intercambio de los datos. Veamos un ejemplo para la base de datos Hospital.]*

Con la intención de validar el buen funcionamiento de las transacciones a la base de datos *Hospital* se realizó un procedimiento para borrar los datos de un ingreso el cual resulto favorable puesto que se validó la integridad de las tablas dado que si no existen los datos del Paciente y del Médico no se podía eliminar el ingreso de ningún paciente.



#### 4. EXIGENCIAS DEL AMBIENTE

*[Define los recursos no humanos que se requieren para realizar las pruebas de base de datos.]*

##### 4.1 Hardware

*[Se debe indicar que hardware se utilizará para la realización de las pruebas. Veamos un ejemplo para la base de datos Hospital.]*

Para realizar las pruebas de integración de la base de datos *Hospital* se utilizó una laptop con procesador INTEL CORE2 a 1.66Ghz, 1Gb de RAM, disco duro de 65Gb.

##### 4.2 Software

*[Se debe indicar que software se utilizará para la ejecución de la base de datos con la interfaz de usuario. Veamos un ejemplo para la base de datos Hospital.]*

El software para realizar las pruebas de integración de la base de datos *Hospital* es, pgAdminIII.

##### 4.3 Sistema Operativo

*[Es necesario indicar que sistema operativo se utilizará para la ejecución de las pruebas de base de datos. Veamos un ejemplo para la base de datos Hospital.]*

Sistema operativo WINDOWS XP

##### 4.4 Sistema Gestor de Base de Datos

*[Define el SGBD en el cual esta cargada la base de datos para su correcta manipulación. Veamos un ejemplo para la base de datos Hospital.]*

SGBD Postgresql 8.2.

Documento de Plan de Pruebas de Integración de Base de Datos
--

Página 5 de 7

---

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

#### 5. PRUEBAS NO FUNCIONALES PARA LA INTEGRACIÓN DE LA BASE DE DATOS

*Describir las pruebas no funcionales que se realizarán durante la integración de la base de datos con la interfaz de usuario. Es importante mencionar que dependiendo de las*

características de la solución se pueden requerir la ejecución de pruebas No Funcionales tales como:

#### 5.1 Prueba de desempeño

*Esta prueba verifica que las transacciones a la base de datos se realizan de manera correcta. Como resultado de la prueba se obtiene el rendimiento de la base de datos en casos extremos de procesos.*

#### 5.2 Pruebas de recuperación

*Esta prueba verifica que la base de datos es capaz de recuperarse durante y después de haber detectado un fallo ya sea físico, lógico e incluso humano. En este caso se debe realizar una prueba de integración para medir la recuperación que ofrece la base de datos.*

#### 5.5 Pruebas de concurrencia

*Define que diferentes usuarios puedan acceder a la base de datos en un mismo tiempo y espacio utilizando la interfaz de usuario.*

### 6. CASOS DE PRUEBAS DE INTEGRACIÓN DE BASE DE DATOS

*Diseñar una tabla que defina los casos de **Prueba de Integración de Base de Datos**, lo que permitirá tener un control de las pruebas que se desean realizar. Veamos un ejemplo utilizando la base de datos Hospital.*

Caso de prueba de integración	Componentes a integrar	Caso de Prueba de Sistema	Componente integrador	Resultado
CSPIBD 1.1	BD <i>Hospital</i>	CSPBD1.1 Ingresar datos del paciente.	Tabla Pa- ciente	Los datos insertados son válidos.

Caso de prueba de integración	Componentes a integrar	Caso de Prueba de Sistema	Componente integrador	Resultado
CSPIBD 1.2	BD <i>Hospital</i>	CSPBD2.1 Consultar datos del paciente desde diferentes terminales.	Tabla Paciente	La consulta se realizó de manera correcta para todos los pacientes que se encuentran en la base de datos.
CSPIBD 1.3	BD <i>Hospital</i>	CSPBD3.1 Borrar datos de Ingreso sin tener datos de paciente.	Tabla Ingreso	La transacción no se ejecuta debido a que no se cumple con las restricciones de integridad referencial en la operación de borrado.

## 7. PERSONAS Y ROLES

*Se describen los roles que existirán para la realización de las actividades del punto 6. Para cada rol, se identifica qué persona va a cumplir el rol y cuales son las responsabilidades, junto con la información de contacto.*

Documento de Plan de Pruebas de Integración de Base de Datos

Página 6 de 7

COMPETISOFT v 0.2

<Nombre de la Empresa>

<Nombre del Proyecto>

---

## 8. PRODUCTOS DE BASE DE DATOS

*Se definen todos los productos de base de datos que se generaron después de la realización de las pruebas de integración de base de datos. Estos entregables están definidos en las **Pruebas de Integración de Base de Datos** dentro de la actividad de Pruebas de Integración. Entre ellos se encuentran:*

- *Planificación de las Pruebas de Integración de Base de Datos.*
- *Diseño de las Pruebas de Integración de Base de Datos.*
- *Pruebas no funcionales para la Integración de la Base de Datos.*
- *Casos de Pruebas de Integración de Base de Datos.*
- *Reporte Pruebas de Integración de Base de Datos.*
- *Reporte de evaluación de las Pruebas de Integración de Base de Datos.*
- *Reporte de Verificación y Validación.*

## 9. APROBADORES

*Se definen las personas que realizarán la aprobación del documento.*

Documento de Plan de Pruebas de Integración de Base de Datos

Página 7 de 7

## Capítulo 4

# Conclusiones

### 4.1. Conclusiones generales

Se realizó un análisis de las actividades y productos que se definen en el proceso de Desarrollo y Mantenimiento de Software de MoProSoft identificando los componentes propios del tratamiento de bases de datos.

Asimismo, se llevó a cabo el análisis de las dimensiones y criterios de calidad que se consideraron para el tratamiento de las bases de datos. Con base en el estudio realizado, se establecieron de manera general las siguientes dimensiones y criterios de calidad para las bases de datos: calidad en relación al proceso de desarrollo y mantenimiento de las bases de datos, calidad de las bases de datos como producto de software, por último la calidad en relación al uso de la base de datos y de los datos mismos.

Posteriormente se revisaron los roles propios del tratamiento de la base de datos, determinado cuales debían integrarse al modelo, para lograr un mejor desempeño en la realización de las actividades relacionadas con el manejo de las bases de datos. Los roles que se definieron para integrarse al modelo fueron, el Administrador de Base de Datos y el Diseñador de Base de Datos, esto, por su nivel de conocimiento con respecto al tratamiento de la base de datos y de los datos mismos.

Con base en lo anterior, se presentaron nuestras contribuciones consistentes en modificaciones, prácticas y guías sugeridas para el proceso de Desarrollo y Mantenimiento de

Software de MoProSoft. Las modificaciones que se sugieren contemplan actividades propias del tratamiento de base de datos sin modificar la estructura del modelo. Dentro de las prácticas se definen los roles, actividades, verificaciones y validaciones sugeridas para el desarrollo y/o mantenimiento de la base de datos. Asimismo, se presentan las guías sugeridas para la elaboración del proceso de Desarrollo y Mantenimiento de Software donde su principal explotación es la base de datos.

Específicamente, sobre estas últimas, propusimos una guías para realizar el levantamiento de requerimientos de la base de datos, para elaborar el análisis y diseño de la base de datos, para la elaboración del plan de pruebas de base de datos y por último, para instrumentar la elaboración del plan de pruebas de integración de la base de datos.

Además se realizaron las pruebas de las prácticas y guías de base de datos para MoProSoft, presentadas en la DGSCA, las cuales fueron consideradas como una aportación para la mejora del proceso de Desarrollo y Mantenimiento de Software. Se observó que las organizaciones que no tienen un proceso establecido tienen un progreso en cuanto a las mejoras del proceso de operación, por su parte empresas que si tienen un proceso establecido como DGSCA reflejan un beneficio en el proceso de Desarrollo y Mantenimiento de Software, y minimizan sus costos de Desarrollo y Mantenimiento de Software, debido a que las prácticas y guías de base de datos permiten estandarizar tareas repetitivas relacionadas con el tratamiento de bases de datos.

Por último, durante el desarrollo de la investigación se logró publicar el artículo de Prácticas y Guías de base de datos para MoProSoft, donde se definen las prácticas y guías de base de datos que se necesitan para el tratamiento de éstas. La contribución propone incrementar el nivel de capacidad de la categoría de operación de MoProSoft.

## **4.2. Trabajos futuros**

Se planea en un futuro realizar más pruebas de las prácticas y guías propuestas en empresas u organizaciones que desarrollen sistemas que tengan una componente de base de datos y que usen MoProSoft ó que no tengan ningún tipo de proceso definido con el objeto de continuar midiendo los beneficios que nuestra propuesta otorgue a los usuarios del

modelo.

Se planea realizar una contribución al modelo que permita incrementar el nivel de capacidad en las tres categorías (Dirección, Gerencia y Operación) de MoProSoft para base de datos distribuidas.

Siguiendo estas ideas deberán revisarse otros dominios en los que se requiera prácticas y guías adicionales para MoProSoft, cómo por ejemplo, sistemas con una componente importante de web, sistemas distribuidos, manejo de imágenes, recuperación de información (information retrieval), comunicación móvil ó referentes a problemas específicos como por ejemplo, astronómicos, médicos, juegos, etc.

# Referencias

- [De Miguel06] De Miguel, A., Piattini, M., y Marcos, E. *Diseño de Base de Datos Relacionales*. Alfaomega RA-MA, 4ª ed<sup>ón</sup>., 2006.
- [Elmasri04] Elmasri, R. y Navathe, S. B. *Fundamentals of Database Systems, Fifth Edition*. Addison Wesley, 2004. ISBN 0-8053-1748-1.
- [García-García08] García-García, J., Rodríguez-Hernández, J., Oktaba, H., Piattini, M., y Ordonez, C. Prácticas y Guías de base de datos para MoProSoft. *SIS'08*, 2008.
- [Genero07] Genero, M., Manso, M. E., Visaggio, C. A., Canfora, G., y Piattini, M. Building measure-based prediction models for uml class diagram maintainability. *Empirical Software Engineering*, 12(5):517–549, 2007.
- [Genero08] Genero, M., Poels, G., y Piattini, M. Defining and validating metrics for assessing the understandability of entity-relationship diagrams. *Data Knowl. Eng.*, 64(3):534–557, 2008.
- [Hanna07] Hanna, O. Moprosoft o historia de una norma. *Software Guru*, 03(1), 2007.
- [Hoxmeier04] Hoxmeier, J. Typology of database quality factors. *Software Quality Journal*, 7(3-4):179–193, October 2004.
- [ISO-2700105] ISO-27001, I. S. O. Tecnología de la información - técnicas de seguridad - sistemas de gestión de la seguridad - requerimientos. *En ISO/IEC 27001*. 2005.



- [ISO-912601] ISO-9126, I. S. O. Software engineering - product quality - part 1: Quality model. iso/iec standard 9126. *En Geneva*. 2001.
- [Lee02] Lee, Y., Strong, D., Kahn, B., y Wang, R. AIMQ: A methodology for information quality assessment. *Inf. Manage.*, 40(2):133–146, 2002. ISSN 0378-7206. doi:[http://dx.doi.org/10.1016/S0378-7206\(02\)00043-5](http://dx.doi.org/10.1016/S0378-7206(02)00043-5).
- [Oktaba05] Oktaba, H., Alquicira, C., Ramos, A. S., Martínez, A., Quintanilla, G., Ruvalcaba, M., López, F., Rivera, M., Orozco, M., Fernández, Y., y Flores, M. Modelo de procesos para la industria del software (MoProSoft). 1(1.3):135, 2005.
- [Oktaba06] Oktaba, H. Moprosoft: A software process model for small enterprises. *En Proc. 1st Int'l Research Workshop for Process Improvement in Small Settings, special report CMU/SEI-2006-SR-001; Software Eng. Institute*, págs. 93–101. [www.sei.cmu.edu/pub/documents/06.reports/pdf06sr001.pdf](http://www.sei.cmu.edu/pub/documents/06.reports/pdf06sr001.pdf), 2006.
- [Piattini01] Piattini, M., Calero, C., y Genero, M. Table oriented metrics for relational databases. *Software Quality Control*, 9(2):79–97, 2001. ISSN 0963-9314. doi:<http://dx.doi.org/10.1023/A:1016670717863>.
- [Piattini06] Piattini, M. y De Miguel, A. *Fundamentos de Base de Datos*. Alfaomega RA-MA, 5<sup>a</sup> ed<sup>ón</sup>., 2006.
- [Piattini07] Piattini, M., Marcos, E., Calero, C., y Vela, B. *Tecnología y diseño de bases de datos*. Alfaomega RA-MA, México, 1<sup>a</sup> ed<sup>ón</sup>., 2007. ISBN 9789701512685.
- [Rumble03] Rumble, J. y Blakeslee, D. The essentials of a database quality process. *Data Science Journal*, 12:35–46, February 2003.
- [Shannon07] Shannon, R. *Systems Simulation: the art and science*. Prentice Hall, 2007. ISBN 9682426731.

- 
- [Silbertchatz07] Silbertchatz. *Fundamentos de Base de Datos*. Mc Graw Hill, 5<sup>a</sup> ed<sup>ón</sup>., 2007.
- [Sommerville04] Sommerville, I. *Software Engineering*. Addison-Wesley, 7<sup>a</sup> ed<sup>ón</sup>., 2004.
- [Wang96] Wang, R. y Strong, D. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–34, 1996.
- [Wietzenfeld05] Wietzenfeld, A. *Ingeniería de Software Orientada a Objetos con UML, Java e Internet*. Thomson, México, 2005.