

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA**



**MIGRACIÓN DE BASES DE DATOS RELACIONALES HACIA
BASES DE DATOS OBJETO RELACIONALES**

TESIS

**PRESENTADA A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR**

MARCO ANTONIO MEJÍA NAVAS

**AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, JULIO DE 1999

**Universidad de San Carlos de Guatemala
Facultad de Ingeniería**



NÓMINA DE LA JUNTA DIRECTIVA

**DECANO
VOCAL 1ro.
VOCAL 2do.
VOCAL 3ro.
VOCAL 4to.
VOCAL 5to.
SECRETARIA**

**Ing. Herbert René Miranda Barrios
Ing. José Francisco Gómez Rivera
Ing. Carlos Humberto Pérez Rodríguez
Ing. Jorge Benjamín Gutiérrez Quintana
Br. Oscar Stuardo Chinchilla Guzmán
Br. Mauricio Grajeda Mariscal
Inga. Gilda Marina Castellanos de Illescas**

**TRIBUNAL QUE PRACTICÓ EL EXAMEN
GENERAL PRIVADO**

**DECANO
EXAMINADOR
EXAMINADOR
EXAMINADOR
SECRETARIO**

**Ing. Herbert René Miranda Barrios
Ing. Byron Manuel Aguilar Lemus
Ing. Franklin Antonio Barrientos Quintana
Ing. Luis Alberto Vettorazzi España
Inga. Gilda Marina Castellanos de Illescas**

**Universidad de San Carlos de Guatemala
Facultad de Ingeniería**

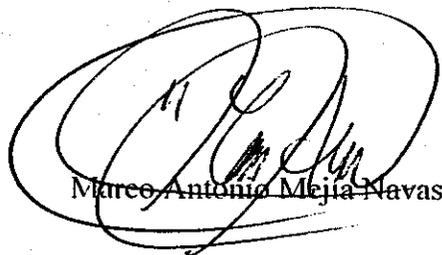


HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de tesis, titulado:

**Migración de bases de datos relacionales hacia
bases de datos objeto relacionales.**

Tema que nos fuere asignado por la coordinación de la carrera de Ingeniería en Ciencias y Sistemas, con fecha 30 de mayo de 1998.



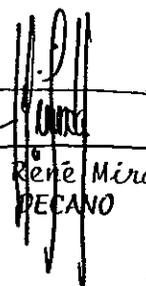
Marco Antonio Mejía Navas



FACULTAD DE INGENIERIA

El decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la autorización por parte del Coordinador de la Carrera de Ingeniería en Ciencias y Sistemas, al trabajo de tesis titulado MIGRACION DE BASES DE DATOS RELACIONALES HACIA BASES DE DATOS OBJETO RELACIONALES, presentado por el estudiante universitario MARCO ANTONIO MEJIA NAVAS, procede a la autorización para la impresión de la misma.

IMPRIMASE:


Ing. Herbert René Miranda Barrios
DECANO



Guatemala, julio de 1, 999



FACULTAD DE INGENIERIA

Guatemala,
21 de julio de 1,999

Ingeniero
Herbert René Miranda Barrios
Decano, Facultad de Ingeniería

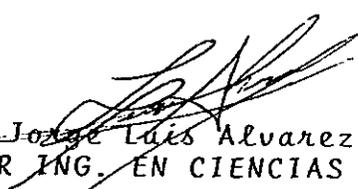
Señor Decano:

De manera atenta me dirijo a usted, para informarle que después de conocer el dictamen del asesor del trabajo de tesis titulado MIGRACION DE BASES DE DATOS RELACIONALES HACIA BASES DE DATOS OBJETO RELACIONALES, elaborado por el estudiante MARCO ANTONIO MEJIA NAVAS, procedo a la autorización del mismo.

Sin otro particular, me suscribo de usted,

Atentamente,

"ID Y ENSEÑAD A TODOS"


Ing. Jorge Luis Alvarez Mejía
COORDINADOR ING. EN CIENCIAS Y SISTEMAS



JLAM/ed,

c.c. Archivo

Guatemala 16 de Abril de 1999

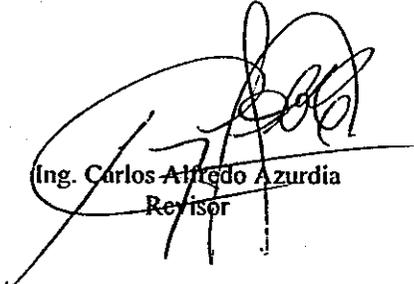
Ingeniero
Jorge Luis Alvarez Mejía
Coordinador
Carrera de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Estimado Ingeniero:

Por medio de la presente, me permito informarle que he procedido a revisar el trabajo de tesis titulado: "Migración de Bases de Datos Relacionales hacia Bases de Datos Objeto Relacionales", elaborado por la estudiante Marco Antonio Mejía Navas, asesorado por el Ing. Everest Medinilla, y a mi juicio, el mismo cumple con los objetivos propuestos para su desarrollo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme

Atentamente,



Ing. Carlos Alfredo Azurdia
Revisor

Guatemala, 14 de abril de 1999.

Ing. Jorge Luis Alvarez Mejía
Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala
Presente

Ingeniero Alvarez:

Por este medio me permito informarle que he procedido a revisar el trabajo de tesis titulado "*Migración de bases de datos relacionales hacia bases de datos objeto relacionales*", elaborado por el estudiante Marco Antonio Mejía Navas, a mi criterio, el mismo cumple con los objetivos propuestos para su desarrollo.

Sin otro particular, me suscribo de usted,

Atentamente,


Ing. Everest Medina
Asesor

AGRADECIMIENTOS

- A DIOS :** Te agradezco padre por la ayuda que día a día y durante toda mi vida me has brindado, porque este triunfo es tuyo, porque yo he sido simplemente alguien que ha utilizado tu sabiduría para ser alguien mejor ante ti. Gracias padre.
- A MI MAMITA:** Gracias mama por toda esa ayuda y esfuerzo que hizo para darme esta educación, porque no fue nada fácil, solo el amor de una madre puede dar tanta ayuda y bondad. Te debo una vida de amor, mamita.
- A MIS HERMANOS:** Gracias Rony, Patty y Lourdes, por brindarme toda esa ayuda que en tantos momentos necesité, porque nunca me negaron, su apoyo y ayuda incondicional.
- A MI NOVIA:** Te agradezco mi querida Karolita, por todo ese cariño y apoyo que en estos difíciles años me has brindado, porque siempre me has comprendido y apoyado para realizar mis metas. Gracias mi querida Puky.
- A MI BANDA DE GUERRA:** Gracias mis amigos y compañeros de batalla, por estar allí conmigo y librar las más duras batallas a las que nos hallamos enfrentado. Gracias a Dios siempre las ganamos todas. Gracias, Heber (Hecman), José Manuel (Cogi de arropa canella.com.gt.etc,etc), Jorge Véliz (¡BUKY! como en física), Edgar Santos y Alvaro Navarro (Power Rangers).
- A MIS AMIGOS:** Por desearme siempre lo mejor y apoyarme en cada momento. Gracias mis amigos de infancia Wicho, Cheyo y George (desde niños ya éramos ingenieros). Gracias mi amigo Rodrigo Guinea (Pío) por ser tan buen cuate siempre (te debo mucho). Gracias a mis genios amigos Vladimiro y Fernando Méndez, por haber compartido tan ingeniosas ideas e inventos. Gracias a todos mis otros amigos; es una gran lista.
- A MIS CATEDRÁTICOS:** Les agradezco profundamente por todas y cada una de esas interesantes cátedras que cada uno de ustedes me ha impartido; además por todas esas desinteresadas fuentes de conocimiento. Gracias en especial a Don Javier Francisco Guevara, Luis Vettorazzi, Carlos Azurdia, Jorge Luis Alvarez, Franklin Barrientos, Geovanni Alvarado y Byron Aguilar.
- A ESTA MAGNA UNIVERSIDAD :** Gracias a la Universidad de San Carlos, que a través de la Facultad de Ingeniería me ha permitido alcanzar tan grande meta en mi vida. Me enorgullece ahora pertenecer a tan organizado, intelectual y especial grupo, como lo son los ingenieros en ciencias y sistemas de la Universidad de San Carlos de Guatemala.

DEDICATORIA

Dedico este trabajo en especial a cinco personas:

A ti Jesucristo:

Te dedico a ti, Jesús, este trabajo; por ser tú la fuente de todo mi esfuerzo, admiración y amor. Gracias a ti y solo a ti que me has ayudado ante Dios nuestro padre, y por quien he podido realizar y lograr este trabajo.

A ti mi querido papá, Marco Antonio Mejía Cabrera (QED):

Lo he logrado papa, y te dedico a ti este trabajo porque aunque no estés conmigo sé que te sientes muy orgulloso de tu pequeño hijo.

A mi asesor Ing. Everest Medinilla:

Le dedico además este trabajo a usted Ing. Medinilla, por tantas ideas, esfuerzo y apoyo para impulsar el logro del mismo. Un millón de gracias ingeniero Everest.

A mi primer catedrático en esta Magna Facultad, Ing. Luis Córdova:

Le dedico también este trabajo a usted ingeniero Córdova, por el interés y buen consejo que me brindó a mí y al grupo del curso Matemática Básica 1, en la primer cátedra, en enero de 1994: "usted no puede darse el lujo de desperdiciar la oportunidad que le brindan sus papás al enviarlo a estudiar acá; usted debe de aprovecharlo". Porque la vida de un estudiante es estudiar. (Seguí su buen consejo al pie de la letra).

Y, finalmente, dedico este trabajo a mi amigo Luis Quiñones

También te dedico este trabajo a vos mi buen amigo Luisito, por ser tan buena onda, tan INTELIGENTE, idealista, y sobre todo, por ser la persona que más he admirado en la carrera de ingeniería en sistemas. Seguiré tratando siempre de ser tan inteligente como vos y poder algún día tener el honor de llamarte colega (continúa viendo la vida tan simple y a la vez tan compleja como lo has hecho hasta ahora, porque vas a encontrar respuestas, muchas respuestas).

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	VII
INTRODUCCIÓN	XI
1. ASPECTOS GENERALES DE LOS SISTEMAS DE BASES DE DATOS	
OBJETO RELACIONALES	1
1.1 Introducción	1
1.2 Estructura y base sobre el modelo relacional	3
1.3 Extensibilidad de complejidad de datos	5
1.4 Extensión de la metodología orientada a objetos	7
1.5 Aspectos básicos de la base relacional objetos	8
1.6 Características migratorias y soporte.....	11
1.6.1 Soporte de objetos	12
1.6.2 Referencias y relaciones del modelo relacional	13
1.6.3 Accesibilidad externa	14
2. REPRESENTACIÓN RELACIONAL EN LAS BASES DE DATOS OBJETO	
RELACIONALES	16
2.1 Introducción	16
2.2 Integración objeto – relacional	17
2.3 Diseño de tablas	19
2.4 Representación de objetos como tablas	21
2.5 Representación de relaciones de objetos como tablas	23
2.6 Representación de jerarquías en la base de datos relacional	25
2.7 Representación de colecciones en la base de datos relacional.....	27

2.8 Representación individual de objetos en la base de datos relacional	29
2.9 Representación de llaves extranjeras	30
3. ANÁLISIS COSTO/BENEFICIO DE MIGRACIÓN DE UNA BASE DE DATOS RELACIONAL HACIA UNA BASE DE DATOS OBJETO RELACIONAL	32
3.1 Introducción	32
3.2 Simplicidad de uso	36
3.3 Simplicidad de desarrollo	37
3.4 Extensibilidad y contenido	38
3.5 Relaciones y representaciones complejas de datos	39
3.6 Desempeño contra interoperabilidad	41
3.7 Distribución, replicación y soporte de datos	42
3.8 Expiración del sistema	43
3.9 Utilización y universalidad de SQL	44
3.10 Soporte de software	45
3.11 Soporte técnico	46
3.12 Administración de la base de datos	47
3.13 Precio transaccional	49
4. CASO PRÁCTICO PARA UNA MIGRACIÓN DE UNA BASE DE DATOS RELACIONAL HACIA UNA BASE DE DATOS OBJETO RELACIONAL	52
4.1 Introducción	52
4.2 Situación inicial de la empresa	53
4.2.1 Actividades iniciales de la organización	54
4.2.2 Estructura de datos relacional de la empresa	56
4.2.3 Nuevas necesidades de la empresa	58
4.2.4 Limitantes del sistema actual	58
4.3 Solución a las necesidades aplicando la tecnología objeto relacional	59

4.4 Evaluación costo/beneficio de la migración objeto relacional	62
4.5 Estructura de datos objeto relacional de la empresa	66
4.6 Solución y cobertura de las nuevas y futuras necesidades de la empresa	68
CONCLUSIONES	69
RECOMENDACIONES	70
REFERENCIAS BIBLIOGRÁFICAS	71
BIBLIOGRAFÍA	72
APÉNDICES	
A. CONCEPTOS DE LOS SISTEMAS DE BASES DE DATOS	
RELACIONALES	73
A.1 Introducción	73
A.2 Fundamentos del modelo relacional	74
A.2.1 Dominios	75
A.2.2 Relaciones	76
A.2.3 Llaves primarias	79
A.2.4 Llaves extranjeras	80
A.2.5 Reglas de integridad	81
A.2.5.1 La regla de integridad de las entidades	81
A.2.5.2 La regla de integridad referencial	82
A.3 El modelo relacional y su aportación	83
A.4 Debilidades del modelo relacional	85
A.4.1 Normalización	85
A.4.2 Reglas de integridad y de gestión	86
A.4.3 Valores nulos	86
A.4.4 Tipos abstractos de datos y objetos complejos	87
A.4.5 Consultas recursivas	87

B. CONCEPTOS GENERALES DE LA METODOLOGÍA ORIENTADA A

OBJETOS	89
B.1 Introducción	89
B.2 Que son los métodos orientados a objetos	90
B.3 Terminología y conceptos básicos	92
B.3.1 Objetos	92
B.3.2 Clases	93
B.3.3 Métodos	93
B.3.4 Atributos	94
B.3.5 Encapsulamiento	94
B.3.6 Metaclases	94
B.3.7 Herencia	95
B.3.8 Abstracción	95
B.3.9 Polimorfismo	95
B.4 Ventajas de la metodología	96
B.5 Problemas y peligros de la metodología	99

ÍNDICE DE ILUSTRACIONES

FIGURAS

1	Consulta de datos de un estudiante utilizando el estándar SQL3	10
2	Consulta de datos de un estudiante utilizando el lenguaje SQL estándar	11
3	Estructura básica del entorno de acceso a la información	19
4	Proceso de diseño de tablas en la base de datos objeto relacional	20
5	Desempeño transaccional de bases de datos por minuto	34
6	Actividades de venta realizadas con los artículos de Xmarket	54
7	Proceso de creación y relación de artículos en la base de datos	55
8	Modelo entidad relación para las actividades de Xmarket	57
9	Migración automática de un esquema relacional hacia un esquema objeto relacional	60
10	Migración utilizando el modelado de objetos	61
11	Jerarquía para la entidad Artículo en el modelo objeto relacional de la empresa Xmarket	67
12	Valores atómicos	75
13	Relación cíclica con valores nulos en las llaves extranjeras	81

TABLAS

I	Usuarios y necesidades de las cuatro tecnologías de bases de datos	6
II	Criterios primarios que integran las bases de datos objeto relacionales	8
III	Características de la integración objeto relacional	18
IV	Consideraciones en la representación de objetos como tablas	22
V	Relaciones entre objetos en un sistema objeto relacional	23
VI	Sistemas de bases de datos por producto y compañía	33
VII	Comparación de doce aspectos de tecnologías de bases de datos	35
VIII	Ejemplos de precio y desempeño transaccionales por base de datos	51
IX	Historia de los métodos orientados a objetos	91
X	Lista de puntos de evaluación de calidad para un sistema de software	98

GLOSARIO

- 4GL** Lenguaje de cuarta generación; este lenguaje soporta instrucciones de SQL, para acceso a las bases de datos.
- ANSI/SQL2/SQL92** Estándar que define las reglas del lenguaje estructurado de consultas SQL (del inglés Structured Query Language) para recuperación y manipulación de datos .
- API** Interface para la recuperación y manipulación de objetos (del inglés Application Interface).
- Clase** Término que representa a un conjunto o universo de elementos, todos del mismo tipo, pero no particulariza un objeto individual.
- Colección** Término que define a una clase que contiene a otras clases dentro de su estructura.
- Datos persistentes** Información físicamente almacenada dentro de la base de datos.
- Distribución** Propiedad de las bases de datos para repartir información entre múltiples entidades, para el acceso continuo y transparente a los datos.
- Dominio** Conjunto total de valores sobre los cuales esta definido un atributo.

EER	Representa el modelo entidad relación extendido en un base de datos objeto relacional.
Expiración	Propiedad de un sistema que indica cuando este llega a su punto de funcionalidad y utilización; esta se da debido a nuevos productos con nuevas funcionalidades o debido a que este ya no puede satisfacer las necesidades actuales.
Extensibilidad	Extensión de representatividad a partir de un modeló teórico agregando extensiones de otra metodología, con el fin de representar al máximo la realidad.
Instancia/Objeto	Término que representa a un elemento particular de alguna clase, con atributos, relaciones y métodos.
ISO/ANSI SQL3	Estándar principal definido para el lenguaje de consultas estructurado en sistemas objeto relacionales.
Mapeo objeto relacional	Traslado de un esquema relacional hacia un esquema objeto relacional; este mapeo permite el traslado de datos hacia el nuevo esquema objeto relacional.
Migración	Cambio de una tecnología hacia otra.
Modelo objeto relacional	Modelo teórico basado en el modelo de datos relacional, que extiende las capacidades representativas de dicho modelo, aplicando la metodología orientada a objetos.

Modelo relacional	Modelo teórico de base de datos, es también mencionado como sinónimo de tecnología relacional. Este modelo se basa en entidades de datos, presentadas en forma plana al usuario.
OID	Término que significa identificador de objetos (del inglés Object Identifier), siendo un valor único que identifica a un objeto particular dentro de la base de datos; en una base de datos objeto relacional este valor es administrado por el manejador de la base de datos.
OMG/OMDG	Modelo estándares de datos que especifica las reglas para las bases de datos orientadas a objetos.
OMDG/OQL	Uno de los estándares como lenguaje de consultas en una base orientada a objetos (del inglés Object Query Lenguaje); permite la recuperación de objetos a partir de cláusulas SELECT ver ISO/ANSI SQL3.
ORDBMS	Sistema administrador de base de datos objeto relacional.
RDBMS	Sistema administrador de base de datos relacional.
Representación objeto relacional	Utilización de la metodología objeto relacional para la definición, especificación y representación de todos los elementos relacionales.
Tabla	Estructura relacional plana conformada por una cabecera y un cuerpo, la cual representa un conjunto de tuplas dentro de la base de datos.

Tecnología

Término que está enfocado a tecnología de bases de datos representa el modelo teórico que fundamenta a todo sistema de base de datos.

Tupla

Elemento particular de una tabla, conformada por un conjunto de valores de la forma atributo-valor, sobre la base de datos.

INDRODUCCIÓN

Las bases de datos relacionales han sido unos de los principales sistemas para la manipulación de información en ambientes computarizados de los últimos años. Actualmente, han surgido otros sistemas de bases de datos como los sistemas de bases de datos orientados a objetos; estos sistemas, gracias a su poder de representación de la realidad, fácil tratamiento de la complejidad y su alta flexibilidad, le han permitido alcanzar gran importancia en el medio.

Debido al éxito que los sistemas de bases de datos orientados a objetos tienen en el medio y todas aquellas necesidades que estos pueden solucionar, los usuarios de bases de datos relacionales que ven con gran interés adquirir las mejoras que la metodología orientada a objetos proporciona y ven con particular interés migrar hacia estos sistemas que poseen características de la metodología orientada a objetos.

Esta necesidad ha permitido la evolución de una nueva tecnología que integra las ventajas que posee el modelo relacional y la metodología orientada a objetos, creándose así la tecnología objeto relacional.

Dada esta nueva tecnología y un sin número de sistemas bases de datos objeto relacionales que han surgido con ella, los usuarios relacionales se enfrentan a problemas surgidos para migrar hacia esos sistemas objeto relacionales.

Sin embargo, existen muchas consideraciones que deben tomarse en cuenta para realizar dicha migración hacia estos sistemas, tales como del costo que les incurrirá, tanto a nivel conceptual como a detalle, factores a considerar como el beneficio que obtendrán con dicha migración, características comunes a considerar entre los dos sistemas y finalmente aquellas características en que difieren los clásicos sistemas de bases relacionales en relación con los sistemas de bases de datos objeto relacionales.

Por lo tanto, el presente trabajo permitirá a los usuarios de sistemas de bases de datos relacionales que deseen migrar hacia un sistema de bases de datos objeto relacional, tener una visión clara, analítica y descriptiva, de los aspectos relevantes, ventajas, desventajas y características primordiales que brindan estos recientes sistemas de tal forma que puedan considerarse aspectos de costo/beneficio para dicha migración y puedan servir de base para la toma de futuras decisiones.

1. ASPECTOS GENERALES DE LOS SISTEMAS DE BASES DE DATOS OBJETO RELACIONALES

1.1 Introducción

El modelo relacional brinda un conjunto de características tales como la facilidad de representación, visualización de los datos como tablas, existencia de un lenguaje de consultas para acceso a los datos y óptimo almacenamiento de la información. Estas características permiten a los usuarios relacionales una interacción fácil con los sistemas y aplicaciones relacionales, y parte de ello su preferencia a este modelo.

Como es propio de todo modelo, los sistemas de bases de datos relacionales presentan la dificultad de información compleja, lo que limita el alcance de las aplicaciones; además, posee la dificultad de representación de problemas, cuyo alcance también queda fuera de este modelo, lo que obliga a los usuarios de este modelo a la especialización de sus aplicaciones, para representar situaciones similares.

Por otro lado, la metodología orientada a objetos brinda características como la representación de información compleja, representación de problemas reales, descomposición e integración de escenarios con alta dificultad y otras características, que en el modelo relacional sería imposible o altamente difícil representar.

Debido a este conjunto de propiedades los usuarios de la metodología orientada a objetos evitan en su mayoría, migrar hacia sistemas de bases de datos relacionales debido a sus limitaciones.

Los sistemas actuales que se basan en la metodología orientada a objetos, también poseen ciertas dificultades, que se derivan de esta alta representación. El principal problema surge cuando se desea almacenar objetos u otra información compleja y obtener luego dichos objetos. Esta dificultad obliga a los sistemas orientados a objetos a crear bases de datos altamente especializadas para las necesidades de este conjunto de usuarios.

Debido a las diferencias entre estas dos poderosas tecnologías y con el propósito de proveer una solución, los fabricantes de bases de datos han combinado lo mejor de ambos mundos para dar origen a la tecnología objeto relacional.

Un sistema de base de datos objeto relacional, permite residir las aplicaciones relacionales con las nuevas aplicaciones de información basada en la orientación a objetos, brindando a los usuarios del modelo relacional todos los nuevos beneficios de la metodología orientada a objetos y a los usuarios de metodología orientada a objetos un fácil acceso, obtención y manipulación de la información, de los sistemas relacionales.

Esta nueva integración trae también consigo ciertas dificultades; una de ellas se debe a que los sistemas de bases de datos relacionales son estandarizados y los sistemas basados en la metodología orientada a objetos no lo son.

Esta falta de estandarización ha implicado que exista una gran diversidad para el desarrollo sobre aplicaciones propiamente orientadas a objetos, lo que influye en restricciones respecto a la portabilidad.

La principal característica de una base de datos objeto relacional es que unifica un modelo de datos que integra la conducta de los objetos con un lenguaje de consultas, y además provee un verdadero mecanismo para el almacenamiento orientado a objetos.

En el presente capítulo se expondrá los fundamentos de la tecnología objeto relacional, así como sus características primordiales, representación dentro de las bases de datos relacional y extensibilidad de objetos, y otros factores importantes como los lenguajes de consulta y los estándares bajo los cuales se regirá esta tecnología.

1.2 Estructura y base sobre el modelo relacional

En la realidad, existen muchas actividades que son llevadas a cabo por múltiples objetos, existiendo una alta complejidad, en la representación de dichos entes, y más aún en las relaciones entre estos. Esta cualidad de representación es una característica que los sistemas de bases de datos objeto relacionales tiene, y permite definir dichos escenarios respecto a su complejidad de información, así también almacenar los datos que de estos escenarios surgen.

Esta cualidad de representación, es obtenida por la extensibilidad de datos que es provista por la tecnología orientada a objetos. La extensibilidad de información dentro de un sistema de base de datos, representa la ampliación a partir de un mundo restringido de tipos básicos de datos a entes tan significativos como la misma realidad.

En un sistema de bases de datos objeto relacionales, la extensibilidad de información representa toda aquella estructura compleja que pueda ser almacenada en una base de datos con estructura relacional.

Debido a que la extensibilidad permite manejar cualquier estructura sin importar lo compleja que esta sea, también esta implica la manipulación de dichas estructuras, tanto para su obtención sino también para su almacenamiento.

Como ejemplo de un problema de extensibilidad de información se podría tener, la extracción de 100.00 de una cuenta bancaria, si el saldo de la cuenta es suficiente y si la firma y voz de un cliente coincide con las especificadas en la tarjeta.

Este ejemplo en la realidad implica, múltiples tipos de datos que solo la extensibilidad de la información puede lograr. Los tipos tradicionales de datos comúnmente son simples datos alfanuméricos; los tipos no tradicionales suelen ser datos como imágenes, series de tiempo, voces, etc. Estos suelen llamarse objetos o tipos de datos complejos.

Actualmente, los sistemas de bases de datos relacionales no manejan en su totalidad esta información compleja, entonces cuando un sistema relacional es extendido con tipos complejos de datos, se dice que este es un sistema con estructura objeto relacional.

La extensibilidad es aquella característica que permite a una base de datos relacional romper sus límites de representatividad de datos, y le permite entrar a un mundo complejo en donde la tecnología orientada a objetos tiene su dominio.

Debido a que un beneficio trae consigo también complicaciones, la extensibilidad también tiene ciertas complicaciones. Una de ellas es la dificultad de almacenamiento, lo que implica al manejador de la base de datos objeto relacional, diferenciar entre todos los tipos de datos almacenados dentro de la base de datos.

Otro problema que surge con la extensibilidad es que también implica el manejo de dominios. Si dentro del modelo relacional de datos los dominios eran de gran dificultad en su manejo, dentro de un modelo objeto relacional este crece proporcionalmente con las grandes ventajas que se obtienen con ella.

Debido a esta situación de complejidad de dominios, los sistemas objeto relacionales brindan al usuario, ciertas propiedades para definir dominios elaborados, en los cuales se especifican todas las posibles relaciones entre dichos objetos.

1.3 Extensibilidad de complejidad de datos

En esta sección se mostrarán los principales aspectos y necesidades que dieron origen a la tecnología objeto relacional, siendo la principal característica, la necesidad de extensibilidad de complejidad de datos.

Desde ya hace varios años la tecnología relacional ha sido la opción más fuerte, dentro de los sistemas de bases de datos. Desde entonces han aparecido nuevas tecnologías como es el caso de los sistemas de bases de datos orientados a objetos, y posteriormente los sistemas objeto relacionales.

Actualmente existen cuatro diferentes tecnologías de bases de datos, cada una respondiendo a diferentes necesidades y diferenciando a usuarios de distinta naturaleza. Estas tecnologías son: relacional, orientada a objetos, objeto relacional y relacional a objetos.

Estas tecnologías no necesariamente están en competencia, debido a que cada una de ellas satisface distintas necesidades. Las bases de datos relacionales poseen la característica de que soportan el lenguaje standard de datos SQL92. Las bases de datos orientadas a objetos soportan el modelo standard de datos OMG y OMDG. Las bases de datos objeto relacionales soportan una extensión del lenguaje SQL92, denominado SQL3. Finalmente, los sistemas relacionales a objetos son todos aquellos son una media entre ambos sistemas pero basados en la metodología orientada a objetos.

A continuación en la tabla I se muestra la clasificación de los usuarios que utilizan las cuatro tecnologías de bases de datos, así también sus principales necesidades de información que provee cada tecnología.

TABLA I. Usuarios y necesidades de las cuatro tecnologías de bases de datos

TECNOLOGÍA:	NECESIDAD:
Relacional	<i>Usuarios tradicionales de SQL, y vistas planas de datos.</i>
Orientada a objetos	<i>Usuarios de objetos con necesidades de bases de datos.</i>
Objeto relacional	<i>Usuarios relacionales con necesidades de extensibilidad de objetos e información compleja.</i>
Relacional objeto	<i>Usuarios híbridos relacionales y objetos, con necesidades migratorias a otros sistemas de bases de datos.</i>

Los sistemas bases de datos objeto relacionales están basados sobre una idea simple: el modelo de datos que ellos soportan, es una extensión del modelo relacional SQL92.

Existen, actualmente, muchas maneras de variar el modelo relacional y dar lugar a muchas tecnologías nuevas, sin embargo hay una razón única por la cual fue escogido los sistemas relacionales con extensión a objetos: una base de datos consiste en un conjunto organizado de tablas y filas múltiples, permitiendo una vista simple, clara y expresiva de la información, esto gracias a que las tablas son “la llave central conceptual en el modelo”.

Debido a esta facilidad, se dotó a la tecnología objeto relacional un número de características de objetos, que fueron provistas como una extensión para el núcleo de los sistemas de bases de datos relacionales, y dan origen a una nueva tecnología relacional con naturaleza de manejo de información compleja.

Esta tecnología brinda grandes beneficios a usuarios relacionales, quienes desean obtener el potencial de las extensiones de objetos. Estos usuarios se caracterizan por querer mantener un estricto marco de trabajo relacional y obtener los beneficios de la metodología orientada a objetos.

En la actualidad debido a que no existe un standard para la tecnología objeto relacional, permite a cada fabricante colocar sus propias extensiones de objetos; sin embargo, como se menciono al inicio de esta sección que existe un standard denominado SQL3, el cual servirá como un standard futuro.

1.4 Extensión de la metodología orientada a objetos

La arquitectura de la base de datos objeto relacional es representada por un híbrido entre tal tecnología relacional y la tecnología orientada a objetos. Debido a esto los sistemas de bases de datos objeto relacionales están dotados para el manejo de tipos complejos de datos y todas las características que posee la metodología orientada a objetos.

En base a esta característica se puede decir que un sistema de base de datos objeto relacional, es un nuevo sistema que expande sus alcances debido a los beneficios insuperables que brinda la orientación a objetos. Esta tecnología a diferencia de la tecnología relacional esta dotada para el manejo complejo de tipos complejos de datos.

Además, brinda las ventajas de un lenguaje extendido de SQL, que permite a los usuarios de los sistemas relacionales una fácil comprensión debido a la alta similitud con el SQL92 standard de los sistemas de bases de datos relacionales.

1.5 Aspectos básicos de la base de datos objeto relacional

Una base de datos objeto relacional esta integrada por dos criterios de definición; denominados: criterios primarios y criterios secundarios. Los criterios primarios se muestran a continuación en la tabla II.

TABLA II. Criterios primarios que integran las bases de datos objeto relacionales

No. criterio:
1. Soporte para el modelo tradicional de objetos. Como lo son objetos, clases, tipos de datos del usuario, atributos, métodos, jerarquías, herencia y polimorfismo.
2. Debe de cumplir las referencias de bidireccionamiento uno a uno, uno a muchos, y muchos a muchos.
3. Debe de contener un lenguaje recuperación de objetos API para la recuperación de objetos, con base a una base de datos orientada a objetos.
4. Debe de contener un lenguaje de consultas SQL/API, para la recuperación de información basada en una base de datos relacional.
5. Deberá basarse en la estructura tradicional relacional con características como índices, reglas de integridad, reglas de validación, funciones de verificación y políticas de seguridad.
6. Debe soportar el standard SQL3 para las bases de datos objeto relacionales.
7. Debe soportar un lenguaje de consultas para la técnica de modelado de objetos.

Estos criterios primarios establecen la funcionalidad principal y estándares que todo sistema de base de datos objeto relacional debe cumplir. Debe de notarse que al menos estos siete puntos deben cubrirse, para que un sistema de base de datos objeto relacional pueda tener un manejo y accesibilidad completa a la tecnología relacional y a la tecnología orientada a objetos.

Seguidamente se exponen los criterios secundarios, siendo estos: el soporte de interfaces de tablas, basada en el modelo relacional, y amplios tipos de datos. Estos criterios secundarios definen a la base de datos objeto relacional en un sentido conceptual de representación de información.

Los objetos reducen la complejidad y debido a que la base de datos objeto relacional hace uso de la técnica del modelado de objetos, permite capturar más fácilmente la información que las tablas relacionales. Sin embargo existen casos donde las tablas son un constructor efectivo de modelación e implementación.

Los amplios tipos de datos como el sonido, imágenes, videos y series de tiempo son comúnmente asociados con la definición de una base de datos objeto relacional. Esto se debe a que los amplios tipos de datos, son propuestos como una característica secundaria debido a muchas aplicaciones, que requieren un soporte de bases de datos para almacenar información, que contenga modelos complejos de entidades, y que estén más allá de lo que una base de datos relacional y una base de datos orientada a objetos puede proveer para su almacenamiento.

Se dice que un sistema de base de datos objeto relacional soporta un amplio conjunto de tipos de datos, debido a que estos tipos de datos son desarrollados por usuarios finales que utilizan, que utilizan directamente las aplicaciones y crean sus tipos de datos complejos dentro del sistema objeto relacional.

El modelo lógico de las bases de datos objeto relacionales poseen un esquema adicional. respecto a propiedades heredadas de las bases de datos relacionales, entre los cuales se constituyen los nuevos dominios en términos de objetos, funciones de verificación a nivel de objetos y reglas de validación referentes a relaciones entre objetos.

Seguidamente el modelo lógico de la base de datos objeto relacional reúne características basadas en clases, relaciones e índices, los cuales definen su estructura básica; además, se incorpora las definiciones semánticas y cualidades de desempeño del modelo relacional.

Un componente esencial del sistema objeto relacional es un lenguaje de consultas. Existen dos estándares para este lenguaje: el OMDG OQL y el ISO/ANSI SQL3. Lo significativo de OQL es que permite recuperar objetos a partir de una proposición SELECT. La ventaja del SQL3 es que soporta consultas de objetos y objetos transversales sin cláusulas de reunión JOIN.

Sin embargo, el resultado coloca en SQL3 el contenido de atributos no de objetos. Sin embargo OQL y SQL3 están contruidos sobre un previo SQL standard. SQL3 y OQL son muy parecidos debido a su facilidad de uso para las interfaces de consultas en bases de datos de objetos y sistemas objeto relacionales.

Como ejemplo se muestra en la figura 1 una consulta desarrollada en el lenguaje standard SQL3.

FIGURA 1. Consulta de datos de un estudiante utilizando el standard SQL3

```
SELECT id, nombre, ciudad, num_depto(MAJOR), edificio(MAJOR)  
FROM estudiante
```

En esta consulta se extraen los atributos id, nombre y ciudad del objeto estudiante. El numero de departamento y la información del edificio son atributos extraídos del objeto departamento a través del direccionamiento de relación superior MAJOR, que extrae los datos del estudiante en un nivel de contención de objetos.

Para una comparación de los mismos datos, se muestra la figura 2, que presenta la misma consulta usando el SQL estándar.

FIGURA 2. Consulta de datos de un estudiante utilizando el lenguaje SQL standard

```
SELECT s.id, s.nombre, s.ciudad, d.num_depto, d.edificio  
FROM Departamento d, Estudiante s  
WHERE s.num_depto = d.num_depto
```

Otra característica fundamental del SQL3 es la capacidad para ejecutar métodos de SQL, definidos por el usuarios. La principal característica del OQL standard es la habilidad para retornar colecciones de objetos de una consulta de SQL.

1.6 Características migratorias y soporte

En muchos ambientes la introducción de la orientación a objetos crea una fundamental desigualdad entre el modelo de programación de objetos y la forma en que la información existente será almacenada.

Debido a esta desigualdad existen factores a considerar cuando se desea migrar información entre un sistema de base de datos a otro, así como migrar aplicaciones o simplemente un cambio de ambiente de trabajo. Para resolver esta desigualdad es necesario considerar ciertas características migratorias y de soporte que los sistemas de base de datos brindan a los usuarios.

Los sistemas de bases de datos objeto relacionales brindan una característica migratoria muy importante denominada "mapeo". El mapeo trata de minimizar al máximo, "la impedancia de la desigualdad entre las aplicaciones de objetos y datos relacionales".

Cuando se hace un mapeo entre aplicaciones se debe de crear un modelo de objetos de una base de datos ya existente. Este procedimiento es denominado una "ingeniería reversa", la cual extrae un esquema de base de datos de otro sistema. En la práctica sobre todo en el ciclo de vida de una aplicación, la ingeniería reversa y la ingeniería delantera necesita ser combinada, en un proceso iterativo de ingeniería entre los objetos y la representación de datos relacional.

Esta representación realizada gracias al mapeo da origen, a que una base de datos objeto relacional permita dar soporte de objetos, así como establecer referencias y relaciones con sistemas de bases de datos relacionales y una completa accesibilidad externa a dichos sistemas objeto relacionales

1.6.1 Soporte de objetos

Recientemente un numero de herramientas ha emergido, las cuales se han usado utilizado para establecer un mapeo objeto relacional. Estas herramientas algunas veces consisten en dos componentes. El primer componente es denominado capturador de esquemas. Esta utilería lee el esquema de la base de datos relacional y genera una representación de objetos.

El segundo componente es un mapeador de esquemas, y esta herramienta establece y mantiene la relación entre los objetos y la representación relacional. Se dice que si un

sistema de base de datos objeto relacional permite hacer una relación entre un esquema relacional y trasladarlo a un esquema orientado a objetos, este es un sistema con soporte a objetos.

Cuando se define como mapear los datos relacionales a aplicaciones de objetos, existen dos formas para establecer las tablas relacionales y los tipos de objetos. La primera forma es definir la tabla como un tipo de dato u objeto, y la segunda forma es utilizar el modelado de objetos, ambas técnicas permiten un amplio soporte de objetos.

Con las tablas como un tipo u objeto, cada fila en la tabla representa un objeto o instancia, y cada columna en la tabla corresponde a un atributo del objeto. Este mapeo uno a uno, de las tablas representadas como un tipo provee literalmente una traducción entre la representación de información y una aplicación de objetos. Este método es simple pero ofrece poca flexibilidad.

1.6.2 Referencias y relaciones del modelo relacional

En contraste con el modelado de objetos, se encuentra las referencias y relaciones al modelo relacional, ya que se puede representar el esquema completo de cualquier base de datos relacional, en un verdadero modelo orientado a objetos. Este esquema relacional es representado por objetos y sus relaciones, de manera independiente del marcado formato de datos.

Cuando se crea un modelo de objetos, de un esquema relacional, es como una traducción semántica. No solo se puede en un modelo de objetos enmascarar un esquema relacional, sino que se puede prever una integración transparente con una o más bases de datos relacionales físicas.

Además, en el modelado de objetos es posible añadir valores a la información de las relaciones, proveyendo una nueva y flexible forma de visualizar y manipular la información.

La manera tradicional para realizar las referencias relacionales y los vínculos de entidades del modelo relacional, es a través del desarrollo manual de objetos.

Finalmente, al ser representadas las tablas y las relaciones de una base de datos relacional como objetos, estos son ingresados como colecciones de objetos al sistema, permitiendo que los usuarios puedan utilizar todos los componentes definidos dentro del sistema como clases para la definición de otros objetos

1.6.3 Accesibilidad externa

Una vez mostrada la forma en que un sistema de base de datos objeto relacional se encarga del soporte de objetos y brinda acceso a las aplicaciones y esquemas relacionales, es necesario especificar la accesibilidad externa al sistema objeto relacional.

Una vez los esquemas de datos son almacenados dentro del sistema objeto relacional, los usuarios externos al sistema realizan múltiples accesos al catálogo de objetos, que contienen las definiciones de los esquemas definidos como clases, y posteriormente hacen acceso a objetos que son representantes de esas clases.

Existen paquetes de librerías de clases, que son provistas ya por los sistemas objeto relacionales, para manejo de los tipos básicos y otros elementos importantes dentro del

esquema. Estos proveen solo el mapeo de tablas como tipos de datos entre la información relacional y objetos.

En el caso de acceso a información externa de naturaleza heterogénea con ambientes de información relacional, existen los sistemas de bases de datos múltiples, los cuales facilitan el acceso y sirven como intermediarios entre otros sistemas y las bases de datos objeto relacionales.

Finalmente, debido a que los ambientes orientados a objetos y los sistemas basados en la tecnología relacional son altamente compatibles con muchos sistemas, estos permiten que, un sistema de bases de datos objeto relacional pueda ser accedido bidireccionalmente, por usuarios o conexiones a otros sistemas de una manera fácil y funcional.

2. REPRESENTACIÓN RELACIONAL EN LAS BASES DE DATOS OBJETO RELACIONALES

2.1 Introducción

En el capítulo anterior se expuso todos aquellos aspectos y características que integran las bases de datos objeto relacionales. En este capítulo se expondrá las principales representaciones que son necesarias a considerar por los usuarios relacionales y los usuarios de sistemas orientados a objetos, y que desean migrar hacia un sistema de base de datos objeto relacional.

Una de las principales características de las representaciones dentro de un sistema objeto relacional, es que permiten definir toda clase estructuras que son familiares a los usuarios en otros sistemas, de tal forma que estos puedan mantener sus características de desempeño y obtener un nuevo conjunto de características para el desarrollo de aplicaciones complejas.

Las representaciones que un sistema objeto relacional realiza, utiliza la metodología orientada a objetos para representar todos los elementos existentes en los sistemas relacionales, como lo es el caso de tablas, tuplas, índices, etc. Esto con el objetivo de que los usuarios relacionales puedan definir sus estructuras básicas de datos y además puedan expandir sus limitaciones de complejidad de información a estructuras más elaboradas.

Este esquema de representación también permite a los usuarios de sistemas orientados a objetos crear y almacenar sus objetos sin estar restringidos por estructuras

relacionales, ya que los sistemas objeto relacionales tienen una base relacional para almacenamiento de información y métodos orientados a objetos para la representación de la información.

También se trata en este capítulo el marco de trabajo que logra la integración objeto relacional y se brinda un conjunto de características básicas para el manejo de la información persistente dentro de la base de datos objeto relacional.

El marco de trabajo objeto relacional que se expone, hace uso de todas aquellas posibles representaciones que sean definidas en un sistema, brindando a los usuarios una interface amigable relativa al sistema que el usuario prefiere; en este caso a los usuarios relacionales y usuarios de métodos orientados a objetos.

2.2 Integración objeto relacional

La unión que existe entre la metodología orientada a objetos y la tecnología relacional dan lugar a la tecnología objeto relacional. Esta tecnología objeto relacional sirve de base a muchos nuevos sistemas de bases de datos relacionales. Debido a esta integración, surge la necesidad de una interface mixta que integre el potencial de ambos mundos; para esto la tecnología objeto relacional posee una interface denominada "entorno de acceso a la información" (Data Access Framework).

Esta interface es un componente de almacenamiento persistente, que provee la tecnología objeto relacional para organizar mejor la información. Además brinda el acceso a información de sistemas diferentes, con el fin de importar distintas clases de datos de múltiples sistemas.

Debido que la base del entorno de acceso a la información se basa en estructuras de objetos, este puede manipular imágenes, sonido y un sin fin de modelos de objetos e información compleja.

Esta interface utiliza la forma natural de las bases de datos relacionales para la presentación tabular de la información, con el objetivo de mostrar vistas simples a los usuarios; también permite el modelado de objetos e información compleja. En la tabla III se muestran las características obtenidas gracias a la integración objeto relacional.

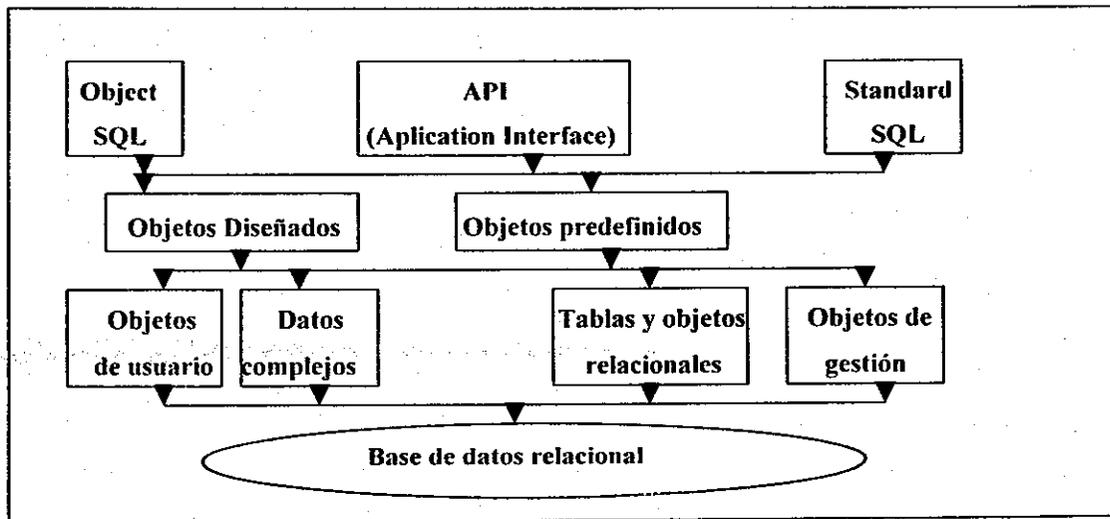
TABLA III. Características de la integración objeto relacional

Flexibilidad:	Minimiza las diferencias para el cambio a la nueva tecnología.
Integración:	Integra la información de sistemas diferentes.
Simplicidad:	Simplifica el manejo, mantenimiento y desarrollo de aplicaciones.
Vistas:	Provee vistas simples y vistas globales a la información.
Multi Lenguajes:	Posee lenguajes múltiples para acceso a bases de datos heterogéneas prerrelacionales, relacionales y orientadas a objetos.

El modelo de datos orientado a objetos dentro del entorno de acceso a la información es un núcleo del modelo de objetos, que provee capacidades como definición de tipos de datos, métodos, identidad de objetos, encapsulación, herencia a multinivel y polimorfismo, sobre una base de datos relacional.

El modelo de datos relacional dentro del entorno de acceso a la información, representa un medio de almacenamiento que soporta grupos repetitivos de datos y tablas; además permite el empleo de un lenguaje SQL de acceso a la información. En la figura 3 se muestra la estructura básica del entorno de acceso a la información.

FIGURA 3. Estructura básica del entorno de acceso a la información



2.3 Diseño de tablas

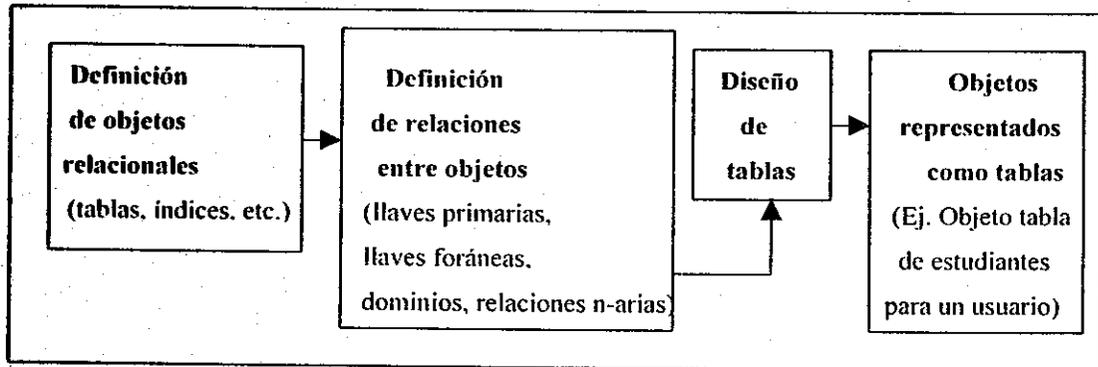
El lado relacional que brinda un sistema objeto relacional, permite la definición de tablas, con el objetivo de determinar como y cuando es mejor definir el esquema de la base de datos. También brinda los medios para almacenar y definir múltiples objetos dentro de estas tablas.

El diseño de tablas permite durante el desarrollo, la definición del esquema relacional, representando objetos como tablas así como la representación de relaciones entre objetos, colecciones, definición de llaves primarias y llaves foráneas.

El diseño de tablas permite además dentro de la base de datos objeto relacional, la definición de los identificadores de objetos (OID's Object IDentifiers), estableciendo la identidades de objetos dentro de la base de datos relacional que es parte del entorno de acceso a la información.

La mejor forma del diseño de las tablas relacionales dentro de un sistema objeto relacional, se logra si se basa este diseño en el modelo de objetos. Este debe ser definido para cada uno de los elementos dentro del conjunto de objetos. Este proceso de diseño de tablas se describe en la figura 4.

FIGURA 4. Proceso de diseño de tablas en la base de datos objeto relacional



Como primer paso en la figura tiene lugar la definición de los objetos básicos como tablas (cabecera y cuerpo), índices y otros objetos que representan elementos del modelo relacional.

Seguidamente, se representan las relaciones entre esos objetos; estas relaciones se representan a su vez como otros objetos, que representan llaves primarias, foráneas, dominios, etc. A continuación se procede al diseño formal de las tablas del esquema en base a las definiciones previas. Este diseño incluye relaciones completas entre tablas.

Finalmente, se procede a la representación de objetos como tablas de tal forma que el esquema de la base de datos, sea aun conjunto de objetos definidos pertenecientes a un usuario particular de la base de datos, y que a su vez sea una tabla del modelo relacional.

Los actuales sistemas de bases de datos objeto relacionales, brindan interfaces que realizan este diseño complejo de objetos, basándose en un esquema relacional que realiza una importación de datos de un sistema relacional, obteniendo como resultado final el último paso del diseño de tablas. Actualmente, se denominan procesos de importación y exportación de esquemas de bases de datos relacionales.

2.4 Representación de objetos como tablas

Como se mencionó en la sección 2.2, los sistemas de bases de datos objeto relacionales, utilizan una base de datos relacional para el almacenamiento persistente de la información. Debido a esta característica, surge la necesidad de mapear las estructuras de los objetos dentro del esquema de la base de datos relacional.

El problema con el mapeo básico de los objetos, instancias y clases es que estas no poseen una llave. Las tablas no tienen la misma propiedad de identidad que los objetos si poseen. Esto surge debido a que los tipos de datos de las tablas en la base de datos relacional, no concuerdan con las clases en el modelado de objetos. Los objetos pueden referenciar otros objetos complejos y colecciones de objetos.

Para solucionar esta diferencia, los sistemas de bases de datos objeto relacionales crean una tabla para cada objeto persistente en su modelo de objetos. Esto determina que tipo de objeto representa cada instancia, y a su vez la estructura que será contenida. Para cada objeto que es representable en una base de datos, como un tipo de dato básico (cadenas, números, fechas, hora, caracteres), se crea una columna en la tabla correspondiente, para que esa instancia de variable sea nombrada como un objeto básico y de lugar a la definición de otros objetos en términos de dichos objetos básicos.

Para la definición de los objetos dentro de la base de datos relacional, se utilizan las siguientes clasificaciones: si una instancia contiene una colección de subclases, se usa una representación de colecciones en una base de datos relacional. Si una instancia contiene cualquier valor se usa una referencia de llaves foráneas. Finalmente se define una columna para contener estos objetos, denominados identificadores de objetos OID's.

Cuando se requiere el diseño de un esquema de una base de datos dentro de un sistema objeto relacional, este puede requerir modificaciones debido a la desnormalización que implica el modelo relacional, para el almacenamiento de objetos. Se debe tener en cuenta múltiples características que es necesario tomar en cuenta para la representación de objetos como tablas, dentro de un sistema objeto relacional; estas se listan en la tabla IV.

TABLA IV. Consideraciones en la representación de objetos como tablas

- 1.- Una clase de objeto representa a una tabla.
- 2.- Una clase de objeto mapea múltiples tablas.
- 3.- Múltiples clases de objetos mapean una tabla.
- 4.- Colecciones del mismo tipo mapean una tabla.
- 5.- Múltiples clases de objetos mapean múltiples tablas.

Todas estas consideraciones deben ser manejadas por la base de datos objeto relacional, debido a que estas permiten el direccionamiento de cualquier relación entre objetos, así como la representación de estos objetos como tablas en el diseño de objetos, y finalmente permiten el almacenamiento y recuperación de la base de datos relacional. Los actuales sistemas de bases de datos relacionales implementan este manejo a través del entorno de acceso a la información.

2.5 Representación de relaciones de objetos como tablas

La representación de objetos como tablas de la sección anterior, permite la representación de un objeto dentro de un esquema relacional, y posteriormente la definición de un objeto como una tabla del modelo relacional.

En esta sección se trata el problema de la representación de las relaciones que existen entre múltiples objetos. Debe de tomarse en cuenta que las relaciones descritas aquí son relaciones que pueden existir y no el termino de "relación" en el modelo relacional.

Debido a que la base de datos objeto relacional, se fundamenta en una base de datos relacional, existen cinco posibles relaciones que pueden darse entre sus objetos almacenados, además se pueden dar varias relaciones que es típico en el modelo de objetos. En la tabla V se muestran las posibles relaciones que un sistema objeto relacional maneja entre los elementos que esta manipula.

TABLA V. Relaciones entre objetos en un sistema objeto relacional.

Tipo de relacion:	Ejemplo:
Uno a uno.	<i>Esposo - Esposa.</i>
Uno a muchos.	<i>Madre - Hijos.</i>
Muchos a muchos.	<i>Acestros - descendientes.</i>
Asociaciones ternarias (n-arias).	<i>Estudiantes - clases - profesores.</i>
Asociaciones calificadas.	<i>Compañías - oficinas - personas - puestos.</i>

Una asociación calificada, es una asociación entre dos objetos donde la asociación es construida o identificada en muchas formas. Por ejemplo, una compañía puede estar asociada con una persona, que a su vez puede estar asociada a múltiples puestos. El puesto califica la asociación entre la compañía y la persona.

La asociación entre objetos puede representar contenciones, asociaciones, o tener un significado semántico en su propio derecho. Por ejemplo el matrimonio es una relación especial entre un hombre y una mujer.

Por otra parte, las relaciones uno a uno, y uno a muchos sirven para una asociación dentro de una clase o crear una clase basada en una asociación. Es importante recordar que la semántica de una relación entre objetos puede ser significativa. Esto es regularmente usual para crear clases para la representación de asociaciones, especialmente si las relaciones tienen este mismo tipo de asociación.

Estas clases serán representadas como tablas en una base de datos objeto relacional. Para las relaciones muchos a muchos, uno a muchos y uno a uno, cuando una asociación tiene un significado de existencia de dominios, se debe crear una clase para esta asociación. Un significado de existencia es cuando una relación por si misma puede tener valores en términos de sí misma, así como valores de declaración, calidad o tipos. Por ejemplo un matrimonio puede tener todas estas propiedades.

Para las relaciones uno a uno, la solución es crear una tabla para cada una de ellas. Si estas tiene un significado especial se crea una tabla basada en la clase derivada de la asociación. En el caso de las asociaciones uno a muchos se crea una tabla de relaciones, que represente una colección dentro de la base de datos. Para las relaciones muchos a muchos siempre se mapea para una tabla que contenga columnas referenciadas por llaves foráneas, de los dos objetos.

En las relaciones ternarias (n-arias), estas pueden tener su propia tabla que haga referencia a todas las llaves foráneas de las clases relacionadas en tales asociaciones. Finalmente una asociación calificada puede tener su propia tabla que represente todas las relaciones existentes posibles, para esa asociación.

Se debe de tener en cuenta que estas relaciones básicamente cubren la mayoría de las existentes entre objetos, lo que permite a usuarios objeto relacionales definir toda relación posible entre elementos.

2.6 Representación de jerarquías en la base de datos objeto relacional

Una de las características importantes que el modelo relacional posee, es la representación simple de jerarquías; estas jerarquías son representadas gracias a las llaves extranjeras las cuales permiten realizar de forma natural esta relación.

Debido a que las base de datos objeto relacionales, están basadas en una estructura relacional, la representación de esas jerarquías utilizando objetos, es una consideración que debe tomarse en cuenta para el diseño y migración hacia estos sistemas objeto relacionales.

El problema básico surge cuando se desea representar un conjunto de clases en una jerarquía de herencia dentro de la base de datos relacional. Este problema se da, debido a que la base de datos relacional no provee soporte para atributos de herencia en su estructura conceptual. Además es imposible hacer mapeo uno a uno entre una tabla relacional y una clase, cuando esa clase posee atributos de herencia de otras clases, o si otras clases son herencia de esa clase.

Existen dos soluciones de contexto posibles, que son usadas para este problema, dependiendo de que es más importante para la aplicación particular, como velocidad de consultas, mantenimiento o flexibilidad para el esquema relacional sobre la base de datos objeto relacional.

La primera solución esta orientada cuando se desea un esquema sencillo. Esta solución se obtiene creando una tabla para cada clase en sus atributos heredados, pudiendo esta incluir concretamente ambas clases.

Las tablas contendrán columnas para cada uno de los atributos definidos en las clases, más una columna adicional que represente las llaves comunes y compartidas entre todas las tablas de subclases.

Una instancia de una subclase concreta es recuperada por medio de un JOIN relacional, de todas las tablas en una ruta para esa clase particular. Esto significa que se establece una raíz de la herencia y posteriormente una subclasificación de clases heredadas.

Esta solución representa un mapeo directo, el cual hace fácil un cambio si una clase en cualquier lugar cambia. Si una clase cambia, se debe cambiar hasta la más lejana de las tablas de la jerarquía. Desafortunadamente la creación de los JOINS a nivel de múltiples tablas puede traer consigo muchos problemas si se tienen herencias a un nivel muy profundo.

La segunda solución se aplica cuando son más importantes las velocidades de las consultas. Para esta, se crea una tabla a cada subclase concreta de la herencia, que contenga todos los atributos definidos en las subclases o heredados en las superclases. Una instancia es recuperada por la consulta a esa tabla.

Esta solución hace las consultas más eficientes, pero tiene el problema que si una superclase es cambiada, entonces muchas tablas pueden ser modificadas. También tiene la dificultad de disminuir el diseño de objetos del esquema relacional.

Existe una tercera solución que puede ser mas apropiada en un ambiente de multiherencia, pero no es muy recomendada mas allá de eso. Es posible crear una simple tabla que represente todos los atributos de las superclases y subclases; seguidamente se utilizan cláusulas SELECT para extraer los atributos apropiados para cada clase.

Sin embargo esta solución puede conducir a un numero grande de valores nulos en la base de datos. La representación de jerarquías dentro de una base de datos objeto relacional se realiza a través del modelo entidad relación extendido (EER), el cual es una técnica que permite definir una solución general en el diseño de jerarquías.

2.7 Representación de colecciones en la base de datos relacional

Otro problema al que se enfrentan los usuarios de las bases de datos objeto relacionales, es la representación de colecciones de objetos dentro del esquema puramente relacional. Este problema surge debido a que los datos persistentes de una base de datos objeto relacional, se realizan mediante un esquema puramente relacional.

La primera forma normal de las bases de datos relacionales previene a una relación de contener atributos multivaluados o a lo que se conoce en la terminología de objetos como una "colección", o comúnmente contención de objetos. La clase de relaciones de 1-N representada en los lenguajes orientados a objetos por clases de colección, son representadas de diferente forma en una base de datos relacional.

La solución para la representación de clases en una base de datos objeto relacional implica representar cada colección en el modelo de objetos, en donde una clase de objeto es asociada a otra clase de objeto por una relación "tiene un", de 1-N; esto se hace por medio de una tabla de relaciones.

La solución básica involucra la creación de una tabla, que consiste de una lista de dos columnas, una para la cual representa la llave primaria del objeto, comúnmente denominado OID (Object Identifier), esto del objeto que *contiene*; la otra columna corresponde a la llave primaria del objeto es *contenido*.

Cada entrada en la tabla muestra una relación entre el objeto que contiene y el objeto que es contenido. La llave primaria de la relación esta conformada por ambas columnas. Una tercera columna puede ser necesitada para indicar la tablas dentro de la cual, la clase del objeto esta localizada.

Las colecciones pueden representar objetos de varias clases. Existe otra posible representación de relaciones 1-N, incluyendo punteros de referencia. Esta tiene el problema de mucho dificultad cuando se desea representar un objeto en más de una colección, y al mismo tiempo cuando las dos colecciones están contenidas en diferentes instancias de la misma clase.

La simple y más común información adicional a incluir en una tabla de relaciones es una columna que indica el tipo del objeto contenido. Esto es necesario cuando una colección puede ser heterogénea.

Si una colección ordenada es utilizada, y el orden es significativo, la posición del objeto en la colección puede ser almacenada en una columna adicional. Esto puede ser notado sin que una columna distinguida indique una posición, o si el OID esta añadido a

una tabla relación y sea hecha parte de la llave primaria, entonces la solución básica representa una colección más general.

2.8 Representación individual de objetos en la base de datos relacional

Una vez establecidas las principales representaciones de objetos en la base de datos objeto relacional, es muy necesario comprender como dichos objetos son identificados y recuperados. Surge entonces el problema de cómo representar individualmente a un objeto dentro de la base de datos relacional la cual constituye el medio de almacenamiento en una base de datos objeto relacional.

Para este problema se asigna un identificador para cada objeto, de tal forma que cada identificador es único en toda la base de datos; estos identificadores también son denominados OID's (Object Identifier). Este identificador deberá ser parte del objeto y deberá ser usado para identificar a este, durante operaciones de consulta, eliminación y actualización.

Los identificadores de objetos pueden ser generados internamente por la de base de datos objeto relacional, o pueden también ser identificados externamente por los usuarios. Algunos sistemas objeto relacionales mantienen generadores de secuencias numéricas para los OID's.

Los identificadores de objetos deben ser definidos de manera única dentro de una clase como valores enteros de longitud grande. Si los identificadores de objeto son generados internamente por la base de datos objeto relacional, es muy común tener una tabla que represente las listas de identificadores definidos para cada clase. Esta tabla será asegurada, consultada, actualizada y desasegurada siempre que un nuevo objeto sea requerido.

En ciertos casos en donde los valores numéricos no sean tan necesarios para los identificadores de objetos, se puede implementar el uso de valores tipo caracter o columnas de cadenas de caracteres, para la asignación de las secuencias; después de todo, lo que es indispensable es la especificación única de un objeto.

2.9 Representación de llaves extranjeras

El problema de la representación de llaves extranjeras en una base de datos objeto relacional surge, cuando en el modelo de objetos, un objeto puede contener no solo tipos de datos base u objetos predefinidos como cadenas, enteros y fechas, sino también objetos o tipos complejos de datos.

Debido a la primera forma normal en las bases de datos relacionales que excluye a los valores no atómicos para cada atributo de las relaciones, es necesario crear otra representación para que un objeto pueda ser contenido, y posteriormente referenciado.

La solución es asignar a cada objeto en el modelo un identificador de objetos. Seguidamente se añade una columna para cada instancia o variable para que almacene un objeto que no represente una colección, y un tipo de dato básico. En esta columna se almacena el identificador de objetos contenido en el objeto previo.

Si el esquema de datos requiere la representación de referencia de llaves extranjeras, el objeto que representa la clase del identificador de objetos, este deberá ser almacenado en esta columna para referenciar dicho objeto.

Debido a que la primera forma normal restringe el diseño orientado a objetos, y representa una dificultad en las bases de datos relacionales, es necesario un mecanismo indirecto como el que se planteó. Este esquema de representación de llaves extranjeras de objetos se vuelve demasiado complejo cuando existen múltiples conexiones entre objetos en forma de raíz, para las consultas de datos.

Es importante considerar que un modelo de datos con alto grado de complejidad referencial, puede ser demasiado complicado representarlo en el diseño de objetos, dentro de un sistema objeto relacional, el cual dentro de un sistema relacional sería simple de definir, gracias a su modelo entidad relación.

3. ANÁLISIS COSTO/BENEFICIO DE MIGRACIÓN DE UNA BASE DE DATOS RELACIONAL HACIA BASES DE DATOS OBJETO RELACIONALES

3.1 Introducción

En el presente capítulo se mostrarán doce características para evaluar el costo/beneficio de una base de datos; debe de tomarse en cuenta que existen muchos otros factores para considerar este análisis, pero para el objetivo de esta investigación serán necesarios solamente estos y deben ser considerados todos si se desea migrar hacia un sistema objeto relacional.

Los usuarios necesitan comprender las diferencias, costos y beneficios que obtendrán cuando realizan una migración. Los doce aspectos considerados son aspectos generales de bases de datos que pueden ser tomados en cuenta para la migración a cualquier sistema de base de datos con excepción de al universalidad del SQL.

Debido a que la tendencia de las nuevas tecnologías de bases de datos es sobre los sistemas relacionales, estas poseen características peculiares que difieren de otros sistemas y que en contraste benefician o representan un costos a no existir en los otros sistemas a los cuales se desea migrar.

El costo al que se hará referencia en este capítulo será de dos tipos: el primero es el costo para obtener los nuevos beneficios que la nueva tecnología a la cual se desea migrar posee. El segundo costo representará aquellas características que no posee la nueva tecnología pero si existen en la antigua y los cuales se perderán al realizar dicha migración. Seguidamente se hará una comparación del beneficio que se obtiene con

la utilización de las nuevas características que posee la nueva tecnología, y las cuales son la razón principal por la que un usuario desea migrar a una base de datos objeto relacional.

En contraste a las bases de datos relacionales y las bases de datos objeto relacionales existen las bases de datos orientadas a objetos. Estas han evolucionado hacia otra área que depende de aplicaciones propiamente más especializadas y con usos específicos. Actualmente existen múltiples compañías que fabrican bases de datos, tanto relacionales como no relacionales, esta se muestran en la tabla VI.

TABLA VI. Sistemas de bases de datos por producto y compañía

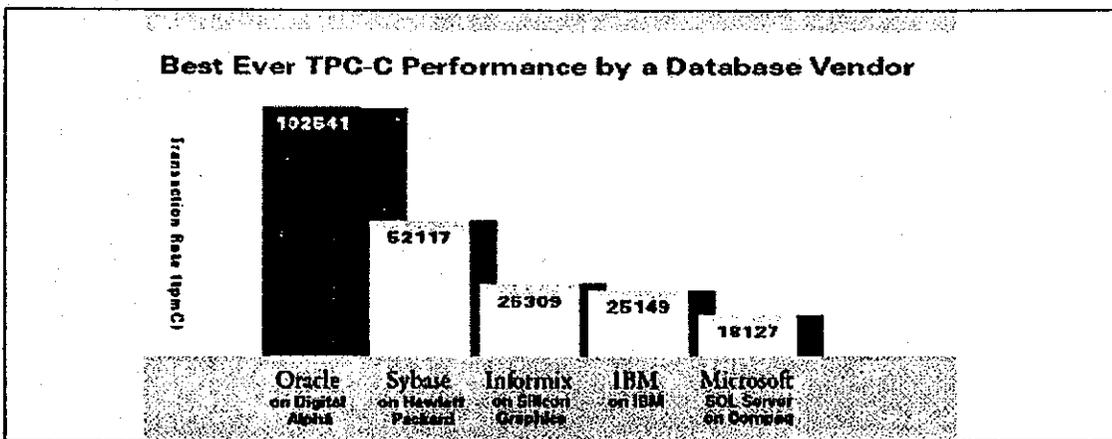
Compañía	RDBMS	ORDMBS	ODBMS
Oracle	<i>Oracle 7.x</i>	<i>Oracle 8.x</i>	
Sybase	<i>System 10/11</i>		
Informix	<i>Dynamic Server</i>	<i>Universal Server</i>	
IBM	<i>DB/2</i>	<i>DB/2 extenders</i>	
UniSQL		<i>UniSQL/X</i>	
Unisys		<i>OSMOS</i>	
Computer Associates	<i>OpenIngres</i>		<i>Jasmine</i>
Gemstone			<i>Gemstone</i>
O2			<i>O2</i>
Object Disign			<i>Object Store</i>
Objectivity			<i>Objectivity/DB</i>
Versant			<i>VersantODBMS</i>

Debido a que el mercado actual de bases de datos esta en transición y alta competencia es necesario determinar los beneficios que permitirá la migración a cualquier sistema de base de datos. Los sistemas de bases de datos listados en la tabla VI cumplen con los doce aspectos que serán evaluados en el presente capítulo.

En el mercado actual de sistemas de bases de datos existe alta competencia, los usuarios evalúan estos productos bajo muchos aspectos; para ello en este capítulo se hará una evaluación con doce aspectos importantes a considerar, necesarios para esta evaluación.

Entre estos aspectos existe el costo, siendo este un aspecto importante en la toma de decisiones, al adquirir cualquier producto. El costo de una base de datos esta sujeto a muchas consideraciones, pero un aspecto importante es el desempeño en el procesamiento transaccional por minuto. En la figura 5 se muestra una comparativa entre algunos de los productos de la actualidad.

FIGURA 5. Desempeño transaccional de bases de datos por minuto



En la figura 5 los resultados muestran el numero de transacciones atendidas por minuto; Oracle8, 102541.85 transacciones por minuto (tpm) a un costo de \$139.49 \$/tpm; Sybase, 52117.80 tpm, \$81.17 \$/tpm; Informix, 25309.20 tpm, \$139.04 \$/tpm; DB2 25149.75 tpm, \$127.93 \$/tpm; Microsoft SQL Server, 18127.40 tpm, \$26.10 \$/tpm.

Seguidamente en la tabla VII se presentan los doce aspectos que deberán ser evaluados para migrar hacia una base de datos objeto relacional; se añade a esta figura los aspectos evaluados para las bases orientadas a objetos como un complemento de análisis.

TABLA VII. Comparación de doce aspectos de tecnologías de bases de datos

ASPECTO	RDMBS	ORDMBS	ODBMS
- Estándar definido	SQL2	SQL3/4	ODMG-V2.0
1. Simplicidad de uso	Estructuras de tablas de fácil entendimiento; muchas herramientas de usuario final disponibles	Igual que en los RDBMS con algunas extensiones Confusas.	Bien para programadores; algunos accesos a SQL limitado para usuario final.
2. Simplicidad de desarrollo	Independencia de datos de las aplicaciones, buena para relaciones simples.	Independencia de datos de las aplicaciones, buena para relaciones simples.	Manejo natural de objetos; puede acomodar variedad de tipos y relaciones.
3. Extensibilidad y contenido	Ninguno.	Limitado más allá de los tipos de datos nuevos.	Puede manejar complejidad arbitraria
4. Relaciones y representación compleja de datos	Dificulta el modelo	Dificulta el modelo	Puede manejar complejidad arbitraria
5. Desempeño contra interoperabilidad	Difiere del fabricante de la base de datos, y requiere pruebas extensas.	Difiere del fabricante de la Base de datos, y requiere Pruebas extensas.	Difiere del fabricante de la base de datos y la complej. de los tipos de datos.
6. Distrib., replic. y soporte de datos	Extensa	Extensa	Varia con el fabricante de la base de datos.
7. Expiración del sistema	Muy expirable	No expirable, debido a las Nuevas extensiones.	Relativamente Expirable
8. Utilización y universalidad del SQL	Posee herramientas de soporte y desarrolladores entrenados.	Puede tomar ventajas de las bases relacionales y Desarrolladores.	Posee SQL pero es orientado a programadores orientados a objetos.
9. Soporte de software	Provisto por los fabricantes de las bases relacionales.	Provisto por los fabricantes de las bases relacionales.	Emulado para soportar aplicaciones relacionales.
10. Soporte técnico	Provisto por los fabricantes de las bases relacionales.	Provisto por los fabricantes de las bases relacionales.	Menor debido a la diversidad de aplicaciones.
11. Admin. de la base de la base de datos	Simple y restringida a grupos de datos.	Aplica forma natural de admón. Individual de datos.	Aplica forma natural de admón. Individual de datos.
12. Precio transaccional	Bajo por simplicidad de las Transacciones	Bajo. Prevalece el modelo transaccional relacional	Alto por la complejidad de las transacciones

3.2 Simplicidad de uso

Las bases de datos relacionales con su metáfora tabular y su estructura columna fila, proveen una simple y fácil interface para los usuarios. Dichas características permiten un beneficio a sus usuarios para la concepción y manipulación de la información y cuyo acceso es por medio de un lenguaje simple denominado lenguaje de consultas (SQL).

Sin embargo, se debe de notar que muchos usuarios usan directamente el SQL para obtener datos. Los proveedores de bases de datos relacionales brindan herramientas comúnmente basadas en formas o interfaces que oculta el SQL del usuario y es generado en segundo plano.

El costo de migrar a hacia los sistemas orientados a objetos, radica en que este tipo de herramientas de interface es poco común en estos sistemas, esto debido a la existencia de información compleja y por tanto no puede ser percibida simplemente como tablas o como atributos fila columna.

Sin embargo, esta situación ha permitido la especialización de programadores de bases de datos, quienes se han introducido en el traslado de tablas hacia estructuras de aplicación: estos han encontrado la forma directa de usar objetos sin tener que forzar dichos objetos dentro de las tablas.

De hecho es que la mayoría de los programadores de hoy en día, están siendo entrenados y especializados en programación de objetos, los cuales los cuales abren el uso de la tecnología de bases de datos también basadas en objetos.

Para evitar la pérdida de la simplicidad de uso, los sistemas objeto relacionales han creado el standard SQL3, que es un lenguaje de consultas con soporte de objetos mientras se mantiene la compatibilidad con el SQL2 de los sistemas relacionales. Las bases de datos objeto relacionales tienen limitado el soporte para jerarquías, con una inconsistente definición de los tipos básicos de datos respecto al SQL3. En todas las compañías de bases de datos objeto relacionales los tipos básicos de datos son tomados como un tipo de dato abstracto u objeto.

La técnica básica que utilizan los sistemas objeto relacionales para almacenar estructuras complejas, es descomponer dichas estructuras a componentes más pequeños o datos básicos y almacenarlos. Una analogía que es atribuida a Ester Dyson podría ser el requerimiento de desensamblar su vehículo antes de guardarlo en su garaje cada noche y entonces re ensamblarlo antes de usarlo por las mañanas. Esta codificación puede costar 25% del esfuerzo de desarrollo, sin embargo el mantenimiento de las aplicaciones que hacen uso de este método es muy simple.

3.3 Simplicidad de desarrollo

Las bases de datos objeto relacionales tiene una forma natural para aquellas organizaciones que tengan enlaces de programación orientada a objetos o manejo de información compleja. El fin principal de una base de datos objeto relacional es proveer a los desarrolladores una correspondencia directa entre las aplicaciones de objetos y el almacenamiento de estos objetos, en donde el desarrollo es muy eficiente.

Este beneficio se logra gracias a que el paradigma de desarrollo orientado a objetos, puede representar de manera natural el mundo real, de un dominio de aplicaciones: los objetos en el mundo real son modelados uno por uno en el software.

Estos objetos referidos en las proposiciones son los mismos objetos representados en el modelo de aplicación, en una herramienta de análisis y diseño, en el código del software y en la base de datos objeto relacional.

Esta correspondencia uno a uno no solo trae consigo el beneficio de simplificación de desarrollo, sino también una simplificación a lo largo del ciclo de vida del software, como además una reducción considerablemente alta durante el mantenimiento. Esto también simplifica la integración de diferentes herramientas de diferentes compañías de bases de datos objeto relacionales y a través de las fases del ciclo de vida de software.

Un costo asociado a la simplicidad de desarrollo radica en que en un modelo relacional, los datos son simples y evitan que las aplicaciones dependan de los datos. En un modelo objeto relacional los objetos y sus métodos pueden representar una dependencia directa con las aplicaciones.

3.4 Extensibilidad y contenido

El tradicional uso de los sistemas de bases de datos ha sido limitada por el restringido y mínimo conjunto de tipos de datos como son los tipos enteros, reales de punto flotante, fecha, cadena, carácter y otros que son soportados por los principales sistemas de administración de información.

El beneficio que brindan las bases de datos objeto relacionales con la extensibilidad es proveer el medio para que los usuarios definan y soporten nuevos tipos de datos. Esto conlleva a que las compañías de bases de datos creen extensiones a sus actuales sistemas.

Por ejemplo IBM creó la extensión para su tradicional base de datos DB/2 denominada DB/2 extenders; así también Informix transformó su base de datos a Informix Universal Server. Oracle también produjo su más reciente versión Oracle 8.x Server; estas se pueden observar en la figura 5. En todos los casos las bases de datos objeto relacionales tienen un número de validaciones necesarias, que los desarrolladores necesitan considerar en los tipos de datos definidos por los usuarios.

Un costo asociado con la extensibilidad consiste en que los tipos de datos construidos, pueden no ser permitidos como tipos abstractos de datos. Por ejemplo las referencias que son usadas entre tablas, puede ser necesario nombrar la fila en que esta situada dicha referencia. Este caso implica direccionamiento a nivel de objetos en forma explícita para representar sus datos. Por ejemplo TABLA.FILA.OBJ_REF; esto conlleva al crecimiento de nombre para identificación de un elemento.

Un costo adicional con el diseño de nuevos tipos de datos, es el contenido de los mismos; esto radica en que deben hacerse consideraciones entre velocidad de referencia y alto desempeño contra extensibilidad de los tipos definidos por el usuario. En otras palabras, qué información debe y qué información no debe contener un tipo de dato abstracto u objeto. Esta decisión debe tomarse en consideración si el usuario desea rapidez, desempeño o diseños adecuados del modelo de datos que se representa.

3.5 Relaciones y representaciones complejas de datos

Las compañías fabricantes de sistemas de bases de datos relacionales y objeto relacionales han tomado en consideración en que significa añadir tipos complejos de datos, y principalmente que los nuevos tipos de datos traen consigo demasiadas relaciones de tipo muchos a muchos.

Un ejemplo podría ser en la administración de un hospital, en donde un paciente puede tener muchos médicos; un doctor puede tener muchos pacientes. Además, pueden existir muchos cuartos de operación y mucho personal staff relacionado con los cuartos de operación.

Un beneficio que permiten las bases de datos objeto relacionales radica en que es posible representar cualquier relación de cualquier naturaleza, sin importar el grado de complejidad de estas, como lo es el caso de relaciones n-arias. Como se mencionó, en la sección anterior, existen también costos asociados al manejo de la información compleja, como lo era el caso del contenido de la información compleja.

Las relaciones también poseen un costo asociado debido que ya no son simplemente relaciones de tipos básicos sino entre objetos o estructuras complejas. Esto repercute en velocidad de ejecución y baja en el desempeño. Debido a la existencia de los identificadores de objetos (OIDs) en las bases de datos objeto relacionales, es posible procesar de forma más simple algunas relaciones complejas.

Esto se logra gracias a la habilidad para navegar por muchas referencias entre tablas en forma bidireccional en relaciones de muchos a muchos y propagando las relaciones a través de múltiples objetos. Además las bases de datos objeto relaciones poseen un pequeño mecanismo motor para las búsquedas de referencias complejas; las bases de datos relacionales no poseen ninguno.

Otro beneficio que se obtiene en las bases de datos objeto relacionales es debido a que el modelado de las referencias es directa y es el manejador de la base de datos quien las opera. En el modelo relacional se requiere la creación de otra tabla para las relaciones muchos a muchos, en el modelo objeto relacional no se requiere debido a que la asociación de entre objetos es una propiedad misma de los objetos.

Como ejemplo considere una base de datos de partes de avión, en donde un avión esta integrado por múltiples elementos. En esta base de datos dependiendo de la complejidad del diseño se requieren al menos 30 o más referencias (JOINS) entre las entidades, para formar un avión completo en el modelo relacional. Esta aplicación fue diseñada en la fuerza aérea francesa en donde se comparó el mismo diseño en una base de datos orientada a objetos. Debido a que la base de datos orientada a objetos no requiere referencias o JOINS para direccionar dato, sino que las relaciones son propias de los objetos, la prueba de desempeño resulto a favor de esta base de datos.

Entonces se puede comprobar que en un modelo relacional no importa que información esta maneje (compleja o no compleja), las relaciones entre entidades son determinan el desempeño. El desempeño es proporcional a las relaciones entre entidades que se tengan. En contraste esta es una ventaja que brindan las bases de datos objeto relacionales al tener que hacer estas relaciones a través de los OIDs.

3.6 Desempeño contra interoperabilidad

La introducción de nuevas extensiones en las bases de datos relacionales como son los objetos dan origen a las bases de datos objeto relacionales. De cierto es que la tecnología objeto relacional es una nueva metodología pero los actuales sistemas objeto relacionales son simples adaptaciones de una base de datos relacional.

En este caso el desempeño es un costo asociado con la adquisición de más tipos de datos. También lo es el tiempo de respuesta de las aplicaciones que dependen de la base de datos objeto relacional. Sin embargo se gana con estas nuevas extensiones la posibilidad de almacenar y operar cualquier objeto en la base de datos, y manipularlo según se requiera; este es el beneficio de interoperabilidad.

Debido a que un incremento en la interoperabilidad es una disminución en el desempeño, esta debe ser una característica a considerar para los usuarios que sean migrar a un sistema objeto relacional. Este se basa en el siguiente enunciado: cuantas más tareas se deban manejar menos tiempo se tiene para cada una de ellas.

3.7 Distribución, replicación y soporte de datos

Una base de datos distribuida transparentemente almacena información a través de múltiples volúmenes y en distintas locaciones. Una base de datos replicada tiene toda o porciones de la información replicada en uno o más diferentes sitios y periódicamente sincroniza el contenido de dicha información replicada.

La replicación de la información es la fundación del Data Ware Housing (bodega de información). El soporte de datos integra bases de datos heterogéneas dentro de un esquemas virtual singular para ser usada por aplicaciones, incluyendo el procesamiento de transacciones.

Todas estas cualidades son soportadas en las bases de datos relacionales y también son soportadas en las bases de datos objeto relacionales, pero además soportan otras características que típicamente no están en las bases de datos relacionales. Entre estas características tenemos, la evolución dinámica de los esquemas. Esto se debe a que cada vez van creandose nuevos objetos en la base de datos, por tanto el esquema esta un constante cambio, por tanto esta cualidad de adaptación dinámica es una característica considerablemente de beneficio en esta nueva tecnología.

Además, se cuenta con otras características nuevas como la automática migración de instancias, versiones y transacciones largas. Se puede concluir respecto a distribución, replicación y soporte de datos que las bases de datos objeto relacionales son totalmente un beneficio para los usuarios debido a que se mejoran las tres cualidades, permitiendo mejores aplicaciones que se apoyan en estas tres cualidades.

3.8 Expiración del sistema

Las bases de datos objeto relacionales han estado bajo desarrollo y usadas por mucho más tiempo que cualquier base de datos con orientación a objetos. Sin embargo, las bases de datos relacionales son productos de expiración.

La expiración ha sido enmarcada debido al conjunto muy limitado de tipos de datos, sin embargo la expiración ha ido aplazándose gracias a características como nuevas funcionalidades, incluyendo soporte de características avanzadas como lo es el procesamiento paralelo, replicación, alta disponibilidad, seguridad y distribución.

Existe una amplia variedad de herramientas y aplicaciones que soportan las bases de datos relacionales trabajando con SQL. Por tanto, las bases de datos objeto relacionales deben estar habilitadas para tomar ventaja de este soporte debido a que ellas son extensiones de las antiguas bases de datos relacionales.

Debido a que una base de datos relacional ha sido limitada por su restringido potencial de representación, se obtiene beneficio de un producto que al menos con las actuales y futuras necesidades se mantendrá lejos de la expiración.

Este avance que han dado las bases de datos relacionales a objeto relacionales ha dado paso a que las bases de datos orientadas a objetos se acerquen más a su expiración, esto debido a que la diferencia principal entre ellas y las bases de datos relacionales ya ha sido alcanzada por las bases de datos objeto relacionales.

3.9 Utilización y universalidad del SQL

El otro beneficio que las bases de datos objeto relacionales poseen es el SQL. Gracias al SQL existe una alta la disponibilidad de desarrolladores experimentados en este lenguaje y la amplia fuente de herramientas de desarrollo, libros y consultas al respecto.

Esta característica benéfica radica en que el SQL es el lenguaje más universal en las bases de datos. Este lenguaje también es 100% soportado en las bases de datos objeto relacionales. con algunas adiciones y nuevas características que constituyen el nuevo estándar SQL3.

Este beneficio que se obtiene del SQL es respaldado por 15 años de investigación por múltiples organizaciones. y porque muchos usuarios están familiarizados con el SQL y tiene herramientas 4GL para desarrollar sus aplicaciones.

Por tanto, se puede decir que el beneficio de este lenguaje se mantiene en las bases de datos objeto relacionales. pero con el valor agregado de nuevas extensiones como lo es la recuperación de objetos y relaciones referenciadas a través de una consulta.

3.10 Soporte de software

Las arquitecturas de las bases de datos han tenido impacto directo sobre los proveedores de bases de datos. Una arquitectura que es más abierta posibilita a través de interfaces de aplicación la portabilidad a través de múltiples plataformas, lo cual implica la expansión del mercado de productos de software para las bases de datos.

Debido a que las bases de datos objeto relacionales son respaldadas por las bases de datos relacionales y sus fabricantes, existe un sin fin de productos que también están disponibles para estos sistemas objeto relacionales. Entre más abiertas sean las bases de datos más fácilmente será la creación de software de soporte y el respaldo de productos que dan apoyo a estos sistemas.

Sin embargo, existen costos asociados y riesgos técnicos para soportar y crear productos para este medio. Éste radica en que la tecnología objeto relacional aun presenta variaciones sin un standard específico entre diversos sistemas, esto en contraste a los sistemas de base de datos relacionales que son un standard bien definido.

Debido a esta característica los fabricantes de las mismas bases de datos han iniciado a crear diversas aplicaciones que dan soporte a las bases de datos objeto relacionales, con el objetivo de aumentar el soporte de software.

En la actualidad es muy restringido el conjunto de herramientas objeto relacionales con que se cuenta pero la alta compatibilidad que existe con las bases de datos relacionales permite la utilización también de estas herramientas.

Por tanto, puede considerarse un beneficio el soporte de software relacional, ya que en su mayoría todas estas herramientas pueden acceder los sistemas objeto relacionales; además se cuenta con el beneficio que muchos de las herramientas relacionales del mercado son los mismos fabricantes de los sistemas de bases de datos.

3.11 Soporte técnico

Una característica benéfica brindada por las bases de datos objeto relacionales es debido al soporte técnico. Debido a que las bases de datos objeto relacionales son de naturaleza evolutiva, y parten del modelo relacional, los fabricantes de las bases de datos relacionales son en su mayoría los mismos fabricantes de las bases de datos objeto relacionales.

Entonces los usuarios relacionales que desean migrar a un sistema de base de datos objeto relacional únicamente deben acudir a su proveedor de base de datos relacional, esto para obtener el soporte técnico deseado, y a su vez tener la ventaja de obtener la misma calidad servicio que le proveedor le brinda con la base de datos relacional. Un costo asociado a esto es debido al actual soporte técnico que se tenga con las bases de datos relacionales. Si el proveedor ofrece mala calidad en el servicio de soporte para las bases de datos relacionales, se corre el riesgo de obtener el mismo servicio en los nuevos sistemas objeto relacionales.

También existe la posibilidad a encontrar nuevos proveedores de soporte técnico con las nuevas tecnologías, lo cual puede tener beneficios en mejoras de la utilización de los nuevos productos y sobre todo un aprovechamiento de todo el poder que la nueva tecnología objeto relacional posee.

3.12 Administración de la base de datos

Existen múltiples aspectos de administración de base de datos que pueden ser analizados para una base de datos. Para efecto del análisis costo/beneficio que se realiza en este capítulo se tomará en cuenta únicamente la administración de datos o de información en la base de datos. Los datos a los que se hará referencia son objetos de bases de datos e información persistente dentro de la base de datos, respecto a datos de usuario.

Los datos en una base de datos objeto relacional son elementos únicos con un identificador de objeto (OIDs) para cada uno de ellos. Debido a que todo acceso a la información en una base de datos objeto relacional se realiza a través de su OID, la administración también toma como base esta característica.

La administración entonces se realiza por elementos únicos, de los cuales se realizan de una manera única. Los controles de concurrencia, acceso, seguridad y mantenimiento de los datos se administran a nivel de las unidades mínimas u objetos mínimos de la base de datos.

Por tanto, se cuenta con el beneficio que cada usuario es propietario y a la vez administrador de sus objetos, entonces corresponde a él conceder cualquier permiso de cualquier elemento. Este método de administración es eficiente debido a que se administra bajo el concepto de un inventario de información (un paso para el Data Ware Housing).

Este concepto es uno de los principales objetos y beneficios al aplicar la administración de datos por elemento, y es el método que las bases de datos objeto relacionales aplica 100% a toda la información que administra, debido a su simplicidad de elementos.

Sin embargo, existe un inconveniente que se debe de considerar, como un costo de este tipo administración de datos. Debido a la necesidad de administrar cada dato individualmente, implica métodos elaborados y de tallados, para llevar un control correcto de la información por cada usuario.

Entonces esta metodología puede representar exceso de controles, información respecto a cada objeto y tablas adicionales, que permitan definir y controlar el acceso relacionados a los datos; en la actualidad los sistemas de bases de datos objeto relacionales asignan esta tarea de control de datos por usuarios a los manejadores de la base de datos, para evitar trabajo a los usuarios administradores.

Se debe de tener en consideración, que cuando es un manejador de base de datos quien se encarga de realizar los controles de administración, puede no existir una política adecuada para compartir todos lo objetos y recursos con que se cuenta dentro de la base de datos.

Por tanto, se recomienda como práctica de orden beneficio, que cada usuario administrador de información aplique sus propios controles de información, acceso a los elementos y se tomen políticas para almacenamiento de la información según sean las necesidades. Para ampliar este concepto puede considerarse aspectos de control de información persistente o conceptos de Data Ware Housing.

3.13 Precio transaccional

El precio transaccional es el factor que permite valuar el precio de una o varias transacciones en una unidad de tiempo especificada. Este factor de precio es comúnmente asociado con el desempeño por la siguiente razón.

Puede existir una base de datos que posea los precios más baratos en una transacción pero el tiempo en ejecutarla puede ser extremadamente largo. Debido a esta razón el precio transaccional es obtenido a partir del desempeño de un número de transacciones en una unidad de tiempo; lo que indica cuán económico es la utilización de la base de datos.

Por ejemplo, una base de datos denominada A puede procesar 1500 transacciones en un minuto y podría costar \$150 procesar dichas transacciones. Sin embargo puede existir otra base de datos llamada B que procese también 1500 transacciones a un costo de \$100, entonces se puede decir que dicha base de datos es de mayor beneficio que la primera: esto es debido a que B no solo tiene un mejor desempeño que A sino que además será más económica por transacción.

Es importante considerar como costo/beneficio el siguiente aspecto. Existen en la actualidad bases de datos con un precio alto en comparación a otras bases de datos, pero con la diferencia que estas bases de datos poseen un mejor desempeño que los productos de menor precio; entonces aunque una base de datos sea de mayor costo al inicio, a largo plazo representará una mejor inversión debido a que el desempeño es el factor que marca el precio transaccional de una base de datos.

En una migración de una base de datos hacia otra es importante comparar este aspecto ya que el fin primordial de dichas migraciones es ampliar la productividad, y este solo es logrado gracias a lo que otros sistemas de bases de datos pueden brindar.

En una migración de una base de datos relacional hacia una base de datos objeto relacional, el precio de diferencia puede resultar igual o en algunos casos menor, esto es debido a que los fabricantes de los sistemas objeto relacionales son los mismos de los sistemas relacionales, por tanto se mantiene todas sus características benéficas, solo que se obtiene un mejor desempeño al mejorarse el manejo de la complejidad.

Finalmente, se debe tomar en cuenta que cuando se desea migrar a un sistema objeto relacional del mismo fabricante, los resultados pueden ser benéficos porque se espera el mismo o un mejor desempeño.

Sin embargo, cuando el se desea migrar a los productos de otro fabricante debe ser necesario esta evaluación de precio transaccional. En la Tabla VIII se muestran ejemplos actuales reales de desempeño.

TABLA VIII. Ejemplos de precio y desempeño transaccionales por base de datos

Database	Hardware Vendor	System	tpmC	Price/tpmC
Oracle7 7.3	Compaq	AlphaServer 8400 5/350 c/s	11456	\$ 285.80
Informix OnLine 7.2	Compaq	AlphaServer 8400 5/350 c/s	13646	\$ 276.93
Oracle7 7.3	Compaq	AlphaServer 8400 5/350 4 Node c/s	30390	\$ 304.67
Oracle Rdb7 7.0	Compaq	AlphaServer 8400 5/350 c/s	14227	\$ 252.08
Microsoft SQL Server 6.50.205	Compaq	Prioris ZX 6166MP c/s	5517	\$ 121.32
Microsoft SQL Server 6.50.205	Compaq	Prioris ZX 6166MP c/s	5740	\$ 116.93
Sybase SQL Server 11.0	Compaq	AlphaServer 1000A 5/300 c/s	1691	\$ 158.24
Informix OnLine Workgroup Server 7.30	Compaq	ProLiant 5000 6/200 Model 2 c/s	6842	\$ 99.63
Sybase SQL Server 11.0.3	Compaq	ProLiant 5000 6/200 Model 2 c/s	8311	\$ 95.37
Microsoft SQL Server 6.5 SP3	Compaq	ProLiant 5000 6/200 Model 2 c/s	7521	\$ 77.67
Oracle7 7.3.3	Compaq	ProLiant 5000 6/200 Model 2 c/s	6393	\$ 108.61
Microsoft SQL Server 6.5 SP3	Compaq	ProLiant 2500 6/200 (2 CPU) c/s	4039	\$ 68.94
Sybase SQL Server 11.1	Compaq	AlphaServer 4000 5/466 (2 CPU) c/s	6767	\$ 118.80
Microsoft SQL Server 6.5 SP3	Compaq	ProLiant 2500 6/200 (1 CPU) c/s	2502	\$ 71.68
Sybase SQL Server 11.0.3	Compaq	AlphaServer 4100 5/466 c/s	10350	\$ 124.45
Microsoft SQL Server 6.5 SP4	Compaq	AlphaServer 4000 5/466	6429	\$ 68.03
Microsoft SQL Server Enterprise Edition 7.0	Compaq	ProLiant 7000-6/400-M1	18127	\$ 26.10
IBM DB2 for Windows NT 2.1.1	IBM	PC Server 720 c/s	1879	\$ 167.83
IBM DB2 for AS/400 V4R1	IBM	AS/400e Server Model s30-2257	3235	\$ 160.06
Microsoft SQL Server Enterprise Edition 6.5	IBM	IBM NetFinity 7000 (c/s)	10559	\$ 47.43
Oracle Oracle8 Enterprise Edition 8.0.4	Sequent	NUMA Center 2000	86252	\$ 170.60
Sybase SQL Server 11.0	Sun	UltraServer I Model170 c/s	1332	\$ 207.55

4. CASO PRÁCTICO PARA UNA MIGRACIÓN DE UNA BASE DE DATOS RELACIONAL HACIA UNA BASE DE DATOS OBJETO RELACIONAL

4.1 Introducción

En el presente capítulo se muestra un caso práctico en donde se realiza una evaluación de una migración a partir de un sistema de base de datos relacional hacia un sistema de base de datos objeto relacional.

El caso analizado se realiza en una empresa de venta de artículos de toda clase al detalle, la cual se llamará Xmarket. En esta empresa se describirá su situación inicial en donde se hará mención de sus principales características de negocio y su sistema de manejo de inventarios, haciendo uso de una base de datos relacional; además se describirá la estructura o esquema de la base de datos.

Además, se mostrarán las limitantes de esta organización con sus actuales sistemas y se mostrarán algunas de las nuevas necesidades de sus clientes que no pueden satisfacer con el sistema actual de bases de datos relacional.

Posteriormente, describe cómo son cubiertas las necesidades que esta empresa afronta haciendo uso de la tecnología de base de datos objeto relacional. Seguidamente se muestra un análisis costo/beneficio de uno a uno, los aspectos que esta organización deberá considerar, para una posible migración hacia la tecnología objeto relacional.

También se muestra, para esta empresa, el esquema de la base de datos objeto relacional, el cual aplica el modelado de objetos, para los datos persistentes en la base de datos objeto relacional, y que cubre los nuevos requerimientos organizacionales de información de esta empresa. Finalmente, se describen nuevas y futuras necesidades que esta tecnología podría ayudar a dicha organización.

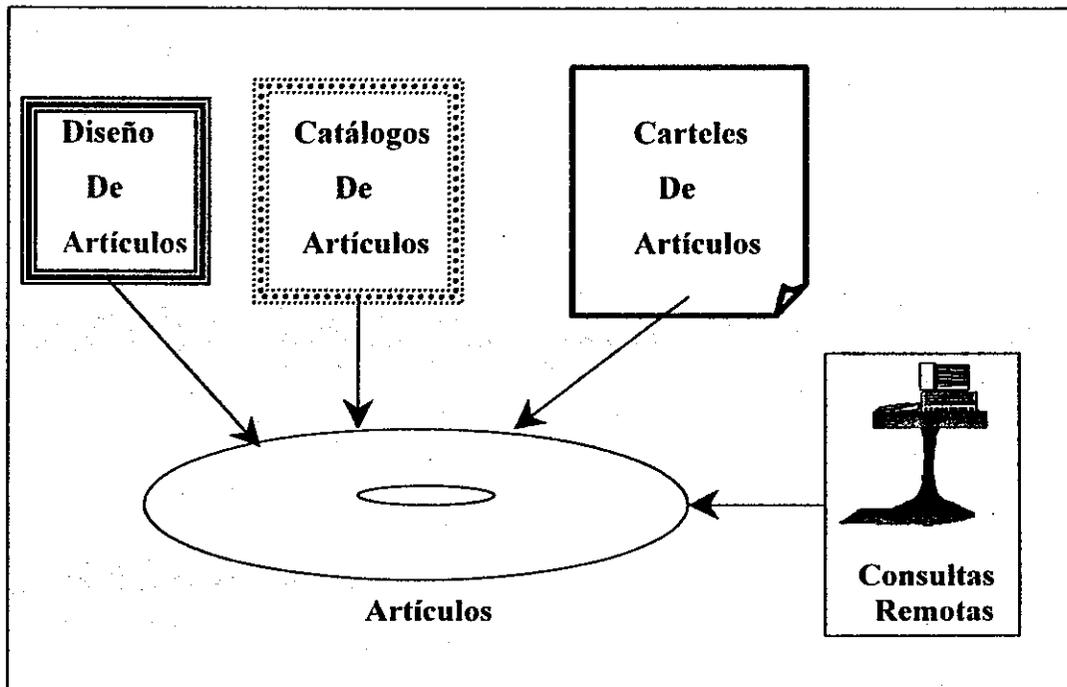
4.2 Situación inicial de la empresa

La empresa Xmarket realiza venta de artículos mecánicos para maquinarias de todo tipo como por ejemplo piezas de vehículos, barcos, maquinaria pesada y otros; todo esto al detalle como en un supermercado normal, solamente que distribuye productos de múltiples partes del mundo, los cuales son conocidos con otros nombres y además poseen otras características adicionales. Como otro servicio que presta Xmarket esta el diseño y fabricación de piezas cuando estas no están disponibles. También cuando Xmarket promociona algún producto que no es muy conocido debe realizar la impresión de carteles para que el cliente conozca el producto mencionado.

Xmarket también realiza ventas remotas, en donde el cliente puede consultar desde su casa el catalogo de artículos y precios. Entre los productos que esta empresa distribuye, existen productos denominados Kits, los cuales son productos integrados por múltiples productos; además existen artículos que son sustitutos y complementos de otros, existiendo un alta complejidad en las relaciones de un producto con otros muchos.

Existen también temporadas de ventas especiales, en donde Xmarket envía a sus clientes especiales un catalogo en donde muestra gráficamente sus productos y precios, así como otros productos relacionados, estas actividades de negocio se muestran en la figura 6.

FIGURA 6. Actividades de venta realizadas con los artículos de Xmarket



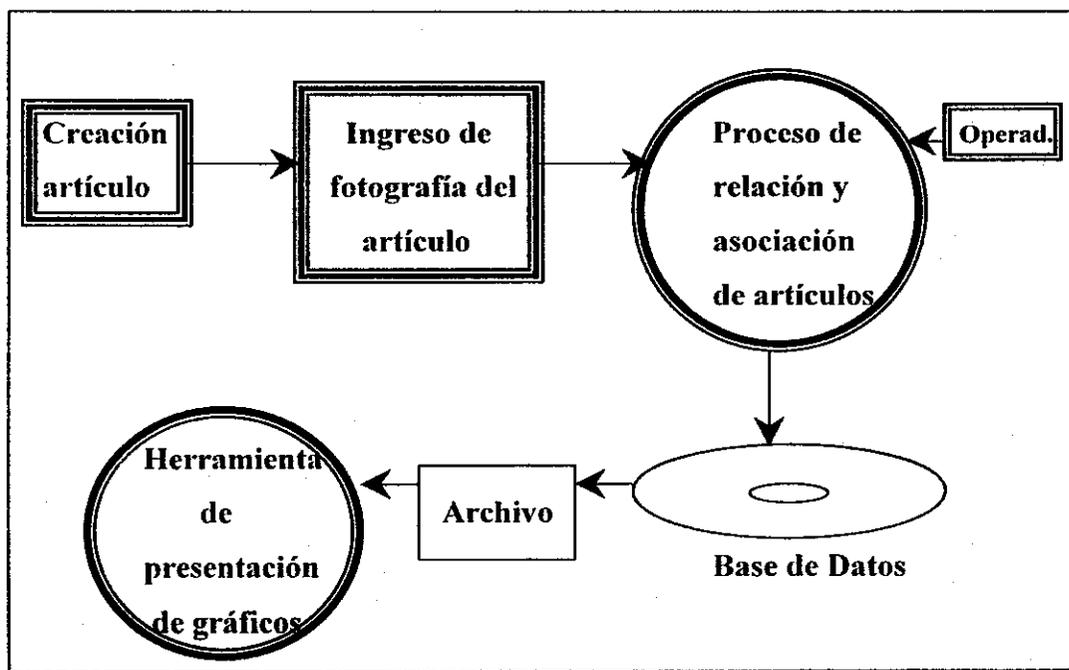
4.2.1 Actividades iniciales de la organización

El enfoque inicial que se considerará para este caso práctico será el manejo de los artículos, ya que estos representan uno de los puntos complejos dentro de las actividades de Xmarket.

La empresa inicialmente adquiere artículos a distintos proveedores; dichos artículos son fotografiados y almacenados dentro de la base de datos. Seguidamente se crean todas las relaciones que este artículo pueda tener con otros artículos de la base de datos; todo esto se realiza a través de la ejecución de una aplicación especializada que al ingresar el artículo a la base de datos relacional, esta es ejecutada por una persona y hechas las relaciones con los artículos correspondientes.

Una vez los artículos han sido ingresados en la base de datos y se han relacionado con otros objetos, se procede a trabajar con dichos artículos. Cuando se envía catálogos de artículos en oferta a los clientes existe una aplicación que extrae los artículos de la base de datos, obtiene su fotografía y lo genera hacia un archivo, en donde se presenta la información en forma tabular. Posteriormente este archivo es trabajado manualmente para la creación del catálogo de artículos, este proceso se muestra en la figura 7.

FIGURA 7. Proceso de creación y relación de artículos en la base de datos



Otras actividades relacionadas con los artículos de Xmarket son las consultas remotas que los clientes pueden hacer desde sus hogares. La actividad de consulta remota inicia cuando un cliente se conecta por medio del Internet hasta Xmarket y consulta artículos y sus respectivos catálogos gráficos de artículos. En este servicio, actualmente se dispone una base de datos con apuntadores a archivos, los cuales despliegan el archivo en la pantalla del usuario.

4.2.2 Estructura de datos relacional de la empresa

Dada la situación de la empresa Xmarket, se procederá a presentar el esquema de la base de datos relacional, a nivel conceptual, que dicha empresa utiliza para satisfacer sus necesidades de información.

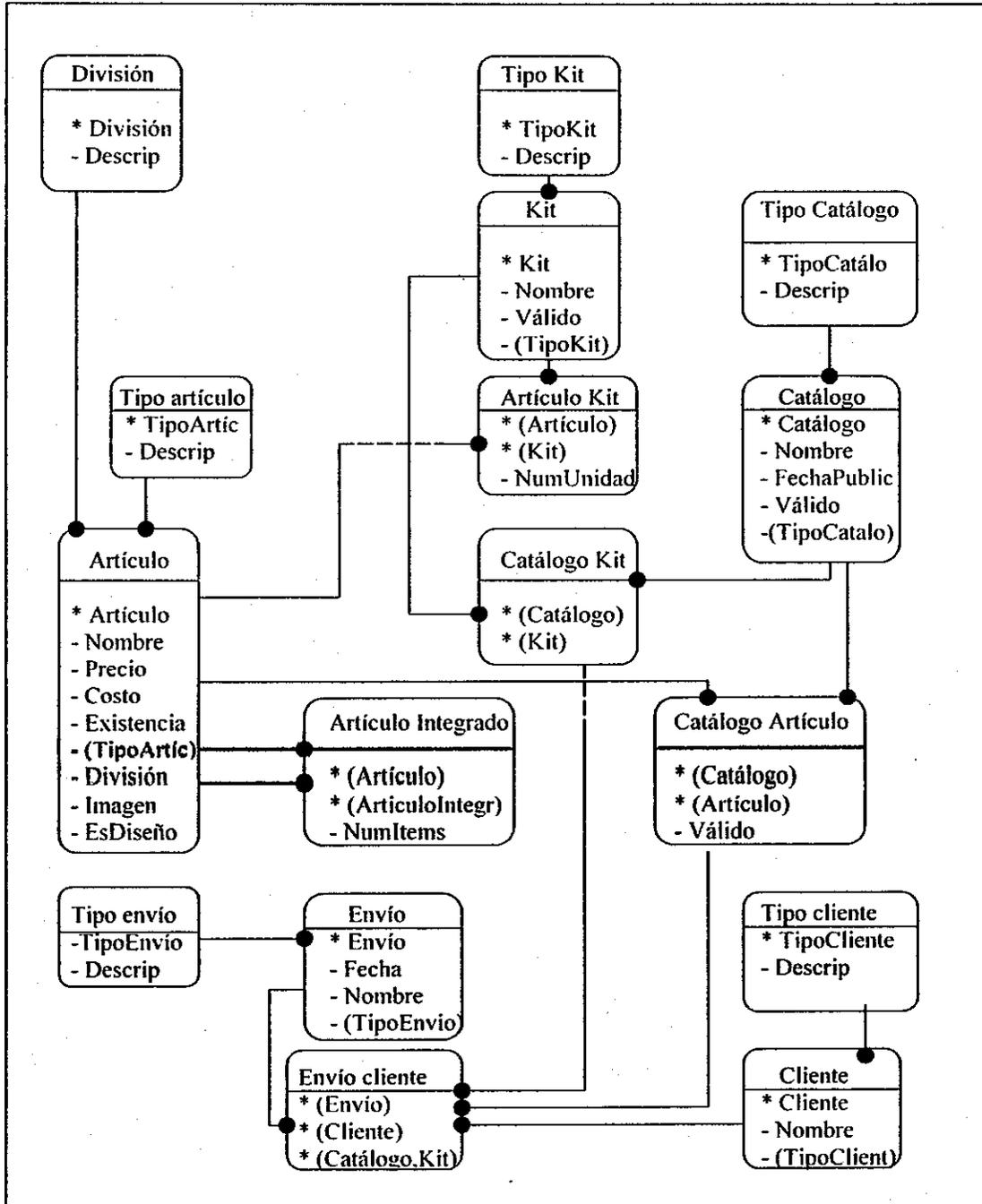
El esquema de datos utilizado ha satisfecho las necesidades actuales de información de Xmarket hasta la actualidad, sin embargo no ha podido integrar la necesidad de una visión a detalle de sus artículos y Kits, para que clientes con múltiples necesidades a detalle puedan realizar una consulta y visualizar ampliamente cada artículo.

El esquema de la base de datos relacional de Xmarket tiene las siguientes entidades: tipo de artículo, artículo, tipo de cliente, cliente, tipo de kit, kit, división, artículo en kit (entidad asociativa de muchos a muchos), artículo integrado (entidad asociativa de muchos a muchos), tipo de catálogo, catálogo, catálogo de kit (entidad asociativa de muchos a muchos), catálogo de artículo (entidad asociativa de muchos a muchos), clase de envío, envío y envío a cliente (entidad asociativa de muchos a muchos).

Las entidades denominadas tipo representan categorías para representar una agrupación de alguna entidad como por ejemplo kit y tipo de kit. Las entidades asociativas de muchos a muchos se describen a continuación: artículo en kit representa que artículos integran un kit en particular; artículo integrado representa a los artículos que están integrados por otros artículos y que pueden ser vendidos individualmente pero no representan un kit; catálogo de kit y catálogo de artículo representan que elementos integran un catálogo en especial; y finalmente envío a cliente representa que catálogos se han enviado a un cliente especial.

A continuación, en la figura 8, se presenta el esquema de la base de datos relacional de Xmarket, de las entidades antes descritas, utilizando el modelo entidad relación para mostrar dichas entidades, atributos y sus relaciones.

FIGURA 8. Modelo entidad relación para las actividades de Xmarket



4.2.3 Nuevas necesidades de la empresa

El esquema de la base de datos relacional presentada en la sección anterior, posee un esquema simple y funcional acorde a los actuales requerimientos de Xmarket, pero sin embargo existen grandes limitantes para poder cubrir nuevas necesidades de la empresa.

Una de las nuevas necesidades de Xmarket, es la presentación de sus múltiples artículos. en donde el cliente pueda visualizar los productos en tres dimensiones y además el cliente pueda descomponer cada producto en sus elementos, de tal forma que el cliente conozca la pieza tal y como es en la realidad.

Otra de las necesidades de Xmarket es permitir a sus clientes a través de las consultas remotas, solicitar diseños de piezas con ciertas especificaciones, utilizando en una base de datos de componentes que se pondrá a disposición del cliente.

Finalmente, una necesidad primordial es la creación dinámica de sus catálogos de venta. ya que por la variedad de sus productos requiere una rápida creación, actualización y eliminación de un catálogo, esto a diferencia del actual método manual de diseño que Xmarket emplea para su elaboración.

4.2.4 Limitantes del sistema actual

Para la cobertura de estas nuevas necesidades el esquema de bases de datos relacional. presenta ciertas limitantes relacionadas con la información compleja.

Estas limitantes surgen debido a que dentro del modelo relacional, no pueden ser representadas colecciones (o clases contenedoras), las cuales si es posible representar dentro de un sistema objeto relacional, utilizando el modelado de objetos. Sin embargo, puede ser posible hasta cierto punto dentro del modelo relacional, realizar dichas representaciones (al menos hasta donde las reglas de normalización lo permiten) utilizando una base de datos relacional, pero se requieren aplicaciones especializadas para el manejo (creación, recuperación, etc) de dichos objetos abstractos.

Debido al problema anterior, las nuevas necesidades no pueden ser satisfechas con la tecnología relacional, debido al soporte de objetos, consulta remota a través de Internet, generación de objetos dinámicos y otras características que son la base esencial de las necesidades nuevas.

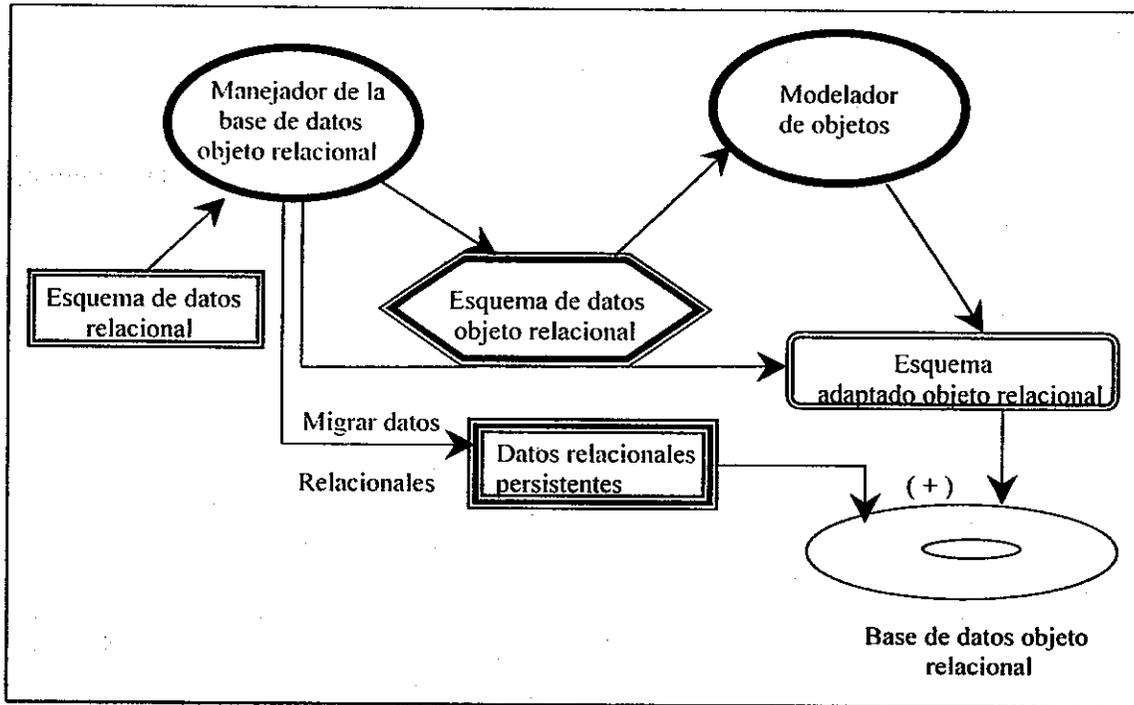
4.3 Solución a las necesidades aplicando la tecnología objeto relacional

La tecnología objeto relacional aplicada a estas nuevas necesidades resuelve dichos problemas de la siguiente manera. Para este caso en particular, en donde existe información almacenada dentro del actual modelo de datos, es necesario primero representar todas las tablas o relaciones del modelo relacional como un objeto en el modelado de objetos. Este procedimiento puede realizarse de dos maneras (ver capítulo 2): utilizando una migración automática y realizando un modelado de objetos OMT (técnica de modelación de objetos).

La primera forma es migrar automáticamente a través del manejador de la base de datos objeto relacional, todo el esquema de datos relacional. Seguidamente realizar una migración de los datos persistentes de la base de datos relacional hacia la base de datos objeto relacional y finalmente modificar el nuevo esquema objeto relacional cambiando los objetos tabla, según las propiedades y necesidades que se requiera de los nuevos

objetos. Para las nuevas necesidades de Xmarket en particular las entidades que requieren un cambio aplicando esta forma son las relaciones Artículo, Catálogo, Catálogo Artículo y Catálogo Kit; todas las demás entidades pueden ser migradas automáticamente por el manejador de la base de datos objeto relacional; esto procedimiento se muestra en la figura 9.

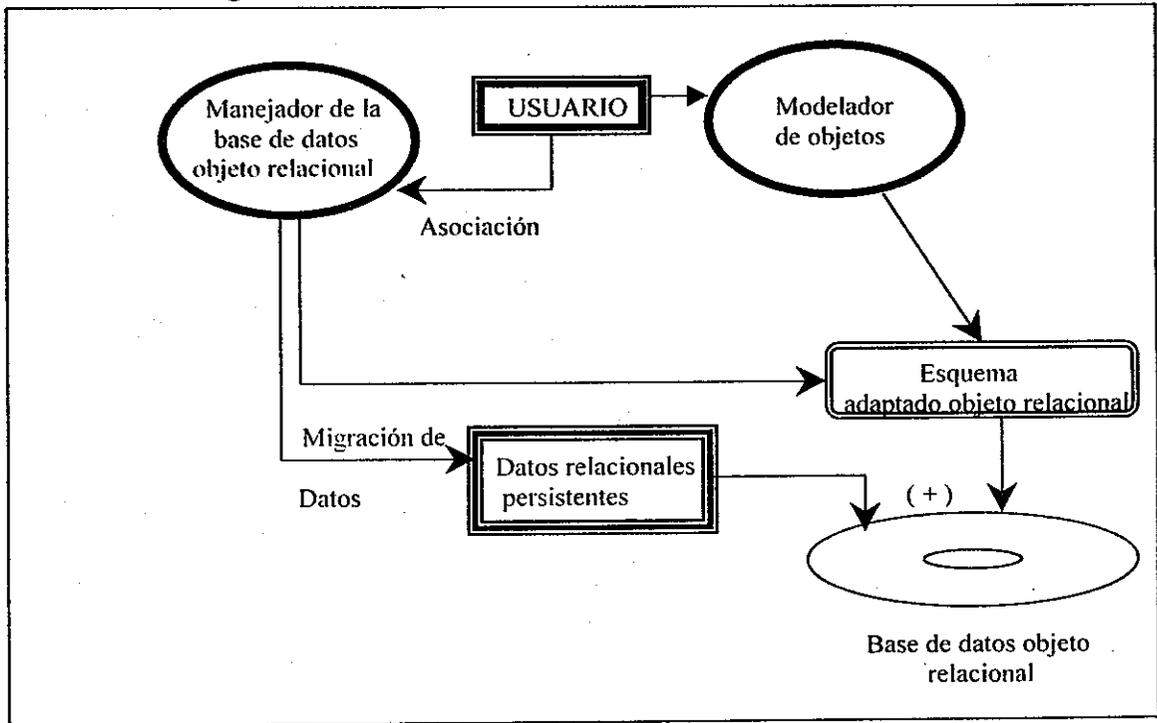
FIGURA 9. Migración automática de un esquema relacional hacia un esquema objeto relacional



La segunda forma se realiza haciendo un modelado de objetos, en el cual se diseña uno a uno los objetos, los cuales representan las tablas del esquema relacional. Seguidamente del modelado de objetos, se realiza una asociación entre los atributos de cada objeto, con elementos de las tablas del esquema relacional y posteriormente, se realiza la migración de los datos persistentes. Este método es aplicable cuando los cambios en el esquema relacional son drásticos y se requiere un diseño diferente.

En la figura 10 se muestra el empleo de la técnica del modelado de objetos. Esta forma es óptima cuando los cambios a realizarse en el esquema relacional son drásticos; esto significa que los cambios a realizar en el esquema relacional son de tal magnitud que dicho modelo no prevalecerá en el moderno, solamente algunas entidades.

FIGURA 10. Migración utilizando el modelado de objetos



Una vez creado el esquema de datos objeto relacional y migrados los datos persistentes a partir de la base de datos relacional, se pueden hacer varias representaciones así como se menciona en el capítulo 2. Para el caso de Xmarket se necesita realizar una representación de jerarquía del modelo relacional recién importado para establecer las clases y especificar la técnica de polimorfismo con el objeto Artículo, el cual aplicará distintos métodos para desplegar sus gráficas. Posteriormente, se puede especificar los métodos del resto de objetos.

Una característica importante que se debe de considerar es que las relaciones uno a uno, uno a muchos, muchos a muchos no son importadas con el esquema relacional, sino que se deben especificar una a una, esto con excepción de las entidades asociativas de muchos a muchos, las cuales si son obtenidas. Estas relaciones muchos a muchos pueden ser no necesarias porque en el modelado de objetos las maneja como una propiedad inerte a los objetos que se importaron (ver capítulo 2, representación de relaciones de objetos como tablas).

4.4 Evaluación costo/beneficio de la migración objeto relacional

Hasta este punto las ventajas y beneficios de una migración a un sistema objeto relacional son altamente atractivas, pero surge la interrogante de cuanto costará y sobre todo de que tan seguro es dicha migración. Existen preguntas como: ¿en dónde encontraremos personal capacitado para manejar dicha base de datos?, ¿qué precio habremos de pagar por transacción?, ¿qué tan complicado será el uso de dicho?, ¿qué tan productivo será nuestro actual equipo de desarrollo en esta base de datos?, ¿qué tan compatibles serán nuestras antiguas aplicaciones con el nuevo entorno?; estas y muchas otra preguntas son hechas por los gerentes de sistemas ante tal situación.

Para los gerentes de sistemas de Xmarket esta situación no es ninguna excepción; entonces para ello se elabora el siguiente análisis costo/beneficio, en donde se comparan los 13 puntos de esta investigación, para obtener el resultado si conviene o no dicha migración.

Debe de tenerse en cuenta como ya se menciona den los capítulos precedentes, que estos 13 puntos aquí evaluados no son los únicos a considerar, pero sí es fundamental que ninguno de ellos sea omitido.

Simplicidad de uso: debido a que el equipo de desarrollo y usuarios Xmarket poseen habilidad para el manejo de sistemas relacionales a través del SQL, la migración hacia el nuevo sistema objeto relacional permite el manejo del mismo SQL, a través del SQL3; este posee el mismo manejo que el estándar SQL únicamente que trae adicionalmente la recuperación de objetos (con sus propiedades y métodos), así como la búsqueda de la jerarquía a la cual pertenece dichos objetos; se puede decir que es un SQL extendido para el manejo de objetos; como costo únicamente se deberán adquirir los conocimientos para el uso de la extensión de objetos que el lenguaje SQL3 y la administración dichos objetos en la base de datos objeto relacional.

Simplicidad de desarrollo: esta característica es lograda gracias a que los nuevos objetos integrados a los sistemas relacionales, dan una forma simple de desarrollo a través del paradigma de objetos, el cual se basa en el ocultamiento y descomposición de los problemas, traducido a una iteración de objetos; como costo deberá instruir su equipo de desarrollo en la programación orientada a objetos. Esto más que un costo será un beneficio a corto plazo ya que obtendrá una programación de mayor calidad en sus aplicaciones.

Extensibilidad y contenido: Xmarket obtendrá una extensibilidad mayor en las definiciones de sus datos, lo que beneficiará a esta en la solución de sus nuevas necesidades (ver sección 4.3.2). Existe un costo asociado con este beneficio, este es el contenido, el cual implicará que el manejador de la base de datos utilice más recursos para obtener y manipular los datos, esto debido a las que los objetos persistentes poseen cierta información asociada (como los OID, relaciones entre objetos, etc), que implica una baja en el velocidad y incremento en el espacio físico de almacenamiento. Después de todo, la mejora en recursos de tecnológicos y de hardware siempre van de la mano de toda organización para incrementar la productividad y brindar un mejor servicio al cliente.

Relaciones y representaciones complejas de datos: la nueva tecnología objeto relacional permitirá el incremento en la representación de problemas y datos complejos, lo que incrementará la solución de problemas a gran escala. También se incrementarán las relaciones entre entidades y objetos a relaciones n-arias. Sin embargo, como en el punto anterior, la representación de estas relaciones implicará una disminución en el rendimiento en la velocidad de las aplicaciones.

Desempeño contra interoperabilidad: Xmarket obtendrá el beneficio de interoperabilidad brindando a sus clientes la posibilidad de interactuar y a sus usuarios de operar con mayor facilidad los datos que en la base de datos objeto relacional estén almacenados, sin embargo el desempeño que se espera dependerá del manejador de la base de datos que adquiera. Para esto Xmarket deberá elegir entre múltiples proveedores de bases de datos objeto relacionales que brinden un buen desempeño y que mantengan compatibilidad con sus actuales sistemas de bases de datos relacionales.

Distribución, replicación y soporte de datos: un sistema objeto relacional brinda las cualidades de distribución, replicación que también existen en las base de datos relacionales, obteniendo el beneficios de soporte de datos, para el almacenamiento y movimiento de estos datos.

Expiración del sistema: toda tecnología tiende a la expiración; esto debido al surgimiento de otras nuevas que brindan características más modernas para realizar las actividades. Debido a que la tecnología objeto relacional es sucesora de la tecnología relacional, esta brinda mayores y mejores características que la anterior. Entonces Xmarket adquirirá una tecnología que tarde o temprano desplazará en su totalidad a la anterior. Esto implicará un cambio en un futuro cercano, posiblemente cuando ya las necesidades sean mayores a las que Xmarket se enfrenta ahora.

Utilización y universalidad del SQL: el SQL es una cualidad que se mantiene en la migración, por lo tanto es una cualidad que no cambia en dicha migración, únicamente que mejoran las técnicas de recuperación, gracias a las características jerárquicas de los objetos.

Soporte de software: debido a que las Xmarket posee sus herramientas para el manejo y enlace con la base de datos relacional, una migración hacia una base de datos objeto relacional, le implicará que sus herramientas de desarrollo, sean modernizadas, para la recuperación de los objetos adicionales con los que esta nueva tecnología cuenta. Sin embargo un objeto como un tipo abstracto de dato no implicará un incremento excesivo en dichas herramientas, pero requerirán una versión que si soporten objetos.

Soporte técnico: Xmarket puede obtener el soporte técnico de dos formas. La primera forma es adquiriendo su nueva base de datos objeto relacional, con los proveedores de su antigua base de datos relacional, así adquirir la misma calidad de servicio y soporte. La segunda forma de obtener soporte técnico, es adquiriendo la nueva base de datos objeto relacional con un nuevo proveedor, de este forma le permite a Xmarket obtener nuevos y más eficientes servicios de soporte, y establecer mejores relaciones con el nuevo proveedor. Sin embargo como costo, puede incurrir en la obtención de un mal servicio del que actualmente se tiene.

Administración de la base de datos: este aspecto es uno de los más importantes a considerar por Xmarket, esto debido a que los actuales administradores conocen aspectos de administración relacional pero no de administración objeto relacional. Estos aspectos nuevos que se adquieren con la base de datos implican un costo en la administración en la nueva base de datos. Sin embargo este punto puede acompañarse de un buen soporte técnico que pueda brindar una total y completa asesoría al respecto de la administración y mejora del desempeño de la base de datos objeto relacional.

Precio transaccional: finalmente este punto permitirá definir a Xmarket el costo/beneficio económico que en realidad la base de datos objeto relacional brindará. Actualmente Xmarket trabaja en promedio 1280 transacciones/minuto en su base de datos, a un costo de \$86.00. Sin embargo se espera que con el nuevo sistema se tengan alrededor de 3000 transacciones/minuto. Xmarket por estas 3000 transacciones/minuto tendría un costo de \$202.00; ahora la evaluación final a realizar sería la comparación entre los precios transaccionales de las bases de datos objeto relacionales y los \$202.00, que la base de datos relacional ofrece. Esta evaluación deberá ser acompañada con otros factores de precio, pero que solo el producto final puede brindar.

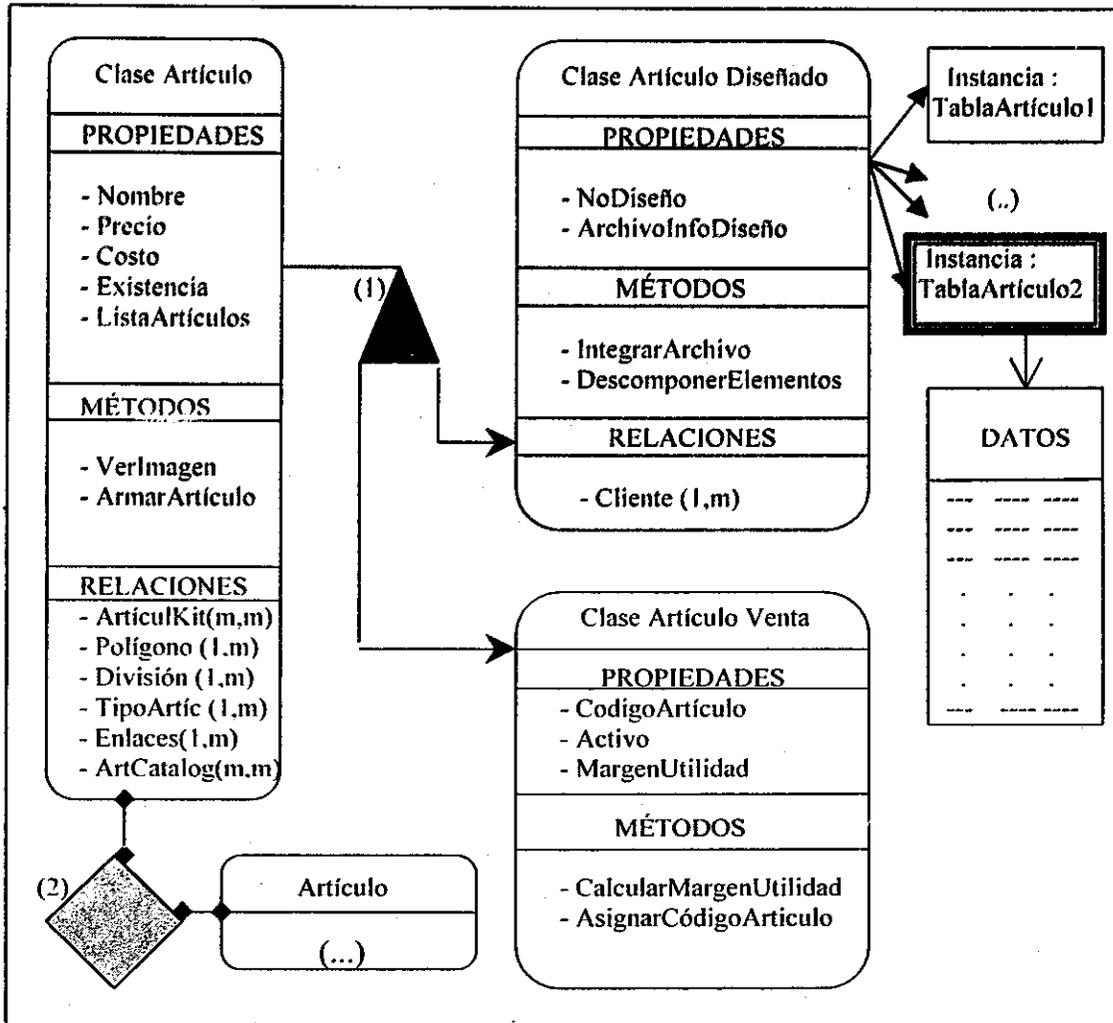
4.5 Estructura de datos objeto relacional de la empresa

En esta sección se presentan algunos de los posibles esquemas de datos que Xmarket puede utilizar ampliando su actual esquema relacional, traducido a un esquema de datos objeto relacional. Como se mencionó, las entidades que básicamente sufren cambio en el esquema relacional son Artículo, Catálogo, Catálogo Artículo y Catálogo Kit; para fines de esta investigación se utilizará la entidad Artículo, para exponer como el esquema relacional puede ser trasladado formalmente a un esquema objeto relacional.

Como parte principal del modelo objeto relacional, es la definición de las entidades que formarán una jerarquía. Esta jerarquías deberán ser definidas como clases para la utilización posterior en la definición de las entidades base (o instancias de esa clase) del modelo objeto relacional; esto como tablas en el modelo relacional.

Para el caso de Xmarket debido a sus necesidades de integrar elementos gráficamente, estos deberán ser definidos como en la realidad se integran artículos más grandes. Esta jerarquía inicial se define a continuación en la figura 11.

FIGURA 11. Jerarquía para la entidad Artículo en el modelo objeto relacional de la empresa Xmarket



En la figura 11 se muestra la jerarquía para la clase Artículo en base al modelo objeto relacional para Xmarket. Primero se muestra la Clase Artículo con tres partes principales: las propiedades de la clase (o bien pueden considerarse como los atributos de una tabla en el modelo relacional), los métodos o interfaces de la clase y finalmente sus relaciones con otras clases (uno a uno, uno a muchos, muchos a muchos a muchos y relaciones n-arias). Seguidamente se encuentran dos clases derivadas de la clase artículo, utilizando el símbolo (1) para esto se utiliza el triángulo.

La generalización significa derivación o especialización a partir de otras. De la Clase Artículo se desprenden dos clases derivadas: Clase Artículo Diseñado (artículos diseñados al cliente) y Clase Artículo Venta (artículos para la venta). Una vez se han especificado las clases derivadas se crean instancias para esas clases; dichas instancias formarán tablas en las cuales se aplicará el lenguaje SQL3 para su manipulación y creación de datos. Finalmente existe el símbolo (2), mostrado como un rombo, representando esta composición y significa que una clase contiene dentro de él, otras clases.

4.6 Solución y cobertura de las nuevas y futuras necesidades de la empresa

La nueva tecnología objeto relacional aplicada por Xmarket, ha dependido únicamente del resultado de la evaluación costo/beneficio que dicha empresa realizó a través de la toma de decisiones para el cambio. Este cambio ha permitido que las nuevas necesidades de representación y descomposición para nuevos servicios puedan ser satisfechas.

Además, este nuevo cambio objeto relacional permitirá a Xmarket poder realizar cualquier proyecto sin límites de representación. A mediano plazo la inversión realizada podrá ser recuperada y Xmarket obtendrá el valor agregado al haber realizado un cambio que en futuro deberá realizar para mantenerse al paso de la tecnología y brindar mejores servicios a sus clientes.

CONCLUSIONES

1. Las bases de datos objeto relacionales son la nueva generación de los sistemas relacionales; por tanto, la migración hacia estos sistemas será definida por las nuevas necesidades empresariales de cada organización en particular, mismas que van surgiendo con el día a día. Sin embargo, es visión de los actuales gerentes de sistemas, definir cuándo realizarla, acorde a sus objetivos.
2. El modelo objeto relacional es evolutivo del modelo relacional, por tanto está diseñado para cubrir todas las necesidades relacionales y, además, un conjunto mucho mayor de necesidades y problemas de múltiple aplicación, que el paradigma orientado a objetos puede enfrentar.
3. Es posible, dentro de un sistema objeto relacional, realizar todas las representaciones relacionales tales como tablas y sus relaciones, llaves primarias y foráneas, concepto de dominio y otros; además, permite representación de datos complejos, igual que la misma realidad.
4. Un sistema de base de datos objeto relacional permite la importación y exportación de esquemas de bases de datos con toda su información, a partir sistemas relacionales y puramente orientados a objetos.
5. Es posible realizar una evaluación consistente para una migración hacia un sistema de base de datos objeto relacional a partir de un sistema relacional, tomando en cuenta como mínimo los doce aspectos evaluados en el capítulo 3;
6. Un gerente puede realizar un análisis costo/beneficio basado en los doce aspectos principales de migración hacia un sistema objeto relacional, que le permitan una visión más amplia para la toma de decisiones de migración.

RECOMENDACIONES

1. Al momento de migrar hacia un sistema de base de datos objeto relacional, debe realizarse un análisis costo/beneficio, que permita al gerente de sistemas evaluar los beneficios/costos que su organización obtendrá al realizar un cambio de sistema de base de datos relacional hacia estos sistemas.
2. Además, debe analizar detalladamente los doce puntos expuestos en el capítulo 3, como una base, y agregar otros puntos acordes a las necesidades de la organización, para evitar un cambio de tecnología sin necesidad (es decir lo que no esta roto no hay que componerlo).
3. Al momento de realizarse una migración a un sistema objeto relacional, se recomienda considerar los aspectos y ventajas adicionales que la metodología orientada a objetos brinda; esto es significativo cuando se utiliza una base de datos objeto relacional, simplemente para hacer uso de los antiguos conceptos relacionales y no de las características de objetos.
4. Se debe considerar, en especial, la compatibilidad todas las herramientas externas al sistema de base de datos objeto relacional, ya que para hacer uso de la tecnología relacional al 100% deberán utilizarse herramientas modernas con soporte a objetos.

REFERENCIAS BIBLIOGRÁFICAS

1. C.J. DATE. **Introducción a los Sistemas de Bases de Datos.** (Vol I. Estados Unidos de Norte América: ADDISON-WESLEY IBEROAMERICANA, S. A., 1993). pp. 73-84.
2. PAUL HELMAN. **The Science of DataBase Management.** (Estados Unidos de Norte América: RICHARD D. IRWIN, INC, 1994), p. 73.
3. TYMOTHY BUDD. **Programación orientada a objetos.** (Estados Unidos de Norte América: Addison-Wesley IberoAmericana, 1994), pp. 92-95
4. KILE BROWN et. al. **A pattern language for object RDBMS Integration.** (Estados Unidos de Norte América: Knowledge Systems Corporation, 1996), pp.17-27.
5. JUDITH DAVIS. **Extending The Relational DBMS To Manage Complex Data.** (Estados Unidos de Norte América: Informix Software Inc, 1996), pp.1-8.
6. ONTOS. **Introduction to ONTOS DB.** (Estados Unidos de Norte América: ONTOS Inc, 1994), p. 85.
7. IAN GRAHAM. **Métodos orientados a objetos.** (2ªed. Inglaterra: editorial ADDISON WESLEY/DIAZ DE SANTOS, 1996), pp. 90-100.
8. JAMES RUMBAUGH et. al. **Modelado y diseño orientado a objetos.** (Estados Unidos de Norte América: PRENTICE HALL, 1991), p. 67.
9. ELISA BERTINO et. al. **Sistemas de bases de datos orientados a objetos, conceptos y arquitecturas.** (México: ADDISON - WESLEY/DIAZ DE SANTOS, 1995), pp. 89-95.
10. FRANCOIS BANCILHON. **Object, Object Relational & Relational Object.** (www.sigs.com, 1996), pp. 8-13.
11. STEVE McCLURE. **Object DataBase vs. Object Relational DataBases.** (Estados Unidos de Norte América: International Data Corporation, 1997), pp.36-46.

BIBLIOGRAFÍA

1. DAVIS, Judith. **Extending The Relational DBMS To Manage Complex Data.** Estados Unidos de Norte América: Informix Software Inc, 1996.
2. GRAHAM, Ian. **Métodos orientados a objetos.** 2ª ed. Inglaterra: editorial ADDISON WESLEY/DIAZ DE SANTOS, 1996.
3. BERTINO, Elisa et. al. **Sistemas de bases de datos orientados a objetos, conceptos y arquitecturas.** México: ADDISON - WESLEY/DIAZ DE SANTOS, 1995.
4. RUMBAUGH, James et.al. **Modelado y diseño orientado a objetos.** Estados Unidos de Norte América: PRENTICE HALL, 1991.
5. HELMAN, Paul. **The Science of DataBase Management.** Estados Unidos de Norte América: RICHARD D. IRWIN, INC, 1994.
6. BANCILHON, Francois. **Object, Object Relational & Relational Object.** www.sigs.com, 1996.
7. ONTOS. **Introduccion to ONTOS DB.** Estados Unidos de Norte América: ONTOS. Inc, 1994.
8. DATE, C.J. **Introducción a los Sistemas de Bases de Datos.** Vol I. Estados Unidos de Norte América: ADDISON - WESLEY IBEROAMERICANA, S.A., 1993.
9. McCLURE, Steve. **Object DataBase vs. Object Relational DataBases.** Estados Unidos de Norte América: International Data Corporation, 1997.
9. BROWN, Kyle et. al. **A pattern language for object RDBMS Integration.** Estados Unidos de Norte América: Knowledge Systems Corporation, 1996.
11. BUDD, Tymothy. **Programación orientada a objetos.** Estados Unidos de Norte América: Addison-Wesley IberoAmericana, 1994.

A. CONCEPTOS DE LOS SISTEMAS DE BASES DE DATOS RELACIONALES

A.1 Introducción

En la actualidad, los sistemas de bases de datos son el principal medio para el control y manipulación de información a pequeña y gran escala, lo que ha permitido a estos sistemas evolucionar constantemente en múltiples áreas.

Con el paso del tiempo y conforme han crecido las necesidades empresariales e individuales de información, las bases de datos se han ramificado en áreas y tecnologías específicas, permitiendo el manejo adecuado de la información, cada una basada en alguna metodología que tiene un alcance en particular.

Una de estas tecnologías es el modelo de datos relacional, el cual se ha caracterizado como uno de los esquemas más aceptados para la esquematización de datos, y que ha dado origen a un conjunto de nuevas tecnologías ampliadas.

Este modelo debido a su simplicidad de representación ha sido base de otros modelos específicos de datos, los cuales han originado varios sistemas de bases de datos con aplicaciones específicas que satisfacen las necesidades de pocos.

Por tanto, el presente capítulo tiene como objetivo presentar los aspectos, conceptos y fundamentos básicos, que constituyen al modelo relacional, con el fin de conocer su naturaleza así como sus alcances y la forma en que dicho modelo representa la información.

A.2 Fundamentos del modelo relacional

Un aspecto importante a considerar, es el límite real o alcance de este modelo; en este existen grandes ventajas, sin embargo también existen debilidades, de las cuales se presentaran algunas al final de este capítulo; esto con el fin de comprender aquellos límites reales donde el modelo relacional termina, y donde otros modelos de datos inician, como lo es el caso del modelo de datos objeto relacional.

Básicamente, el modelo relacional esta integrado por dos partes intrínsecas y dos partes extrínsecas. La primera de las partes intrínsecas es una parte estructural que utiliza nociones de dominios, relaciones n-arias, atributos, tuplas, llaves primarias y llaves extranjeras. La segunda parte es una parte manipulativa, cuyas herramientas principales son el álgebra relacional, el calculo relacional y la asignación relacional, cuya teoría queda fuera del alcance de esta investigación.

Una de las partes extrínsecas está representada por la integridad de las entidades así como por la integridad de las referencias. El componente final que normalmente se considera como interno al modelo relacional, aun cuando esto no es estrictamente así, es la parte de diseño, que consta de la teoría de las formas normales.

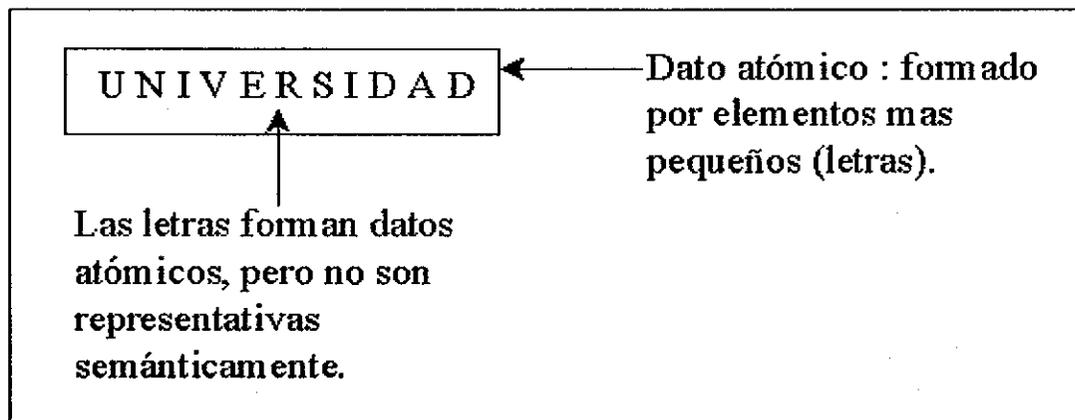
El modelo relacional como modelo de datos, define en su estructura, las características que la información a manipular debe poseer, así como la forma en que ésta debe ser estructurada y organizada, para formar grupos de datos correspondientes y relacionados entre sí. La idea básica es que los datos se representan como una serie de tablas o de archivos planos, "además no deben permitirse grupos de datos repetitivos, ni jerarquías implícitas y ningún enlace estructural fijo debe formar parte de la base de datos". Debido a esta característica, el modelo relacional define dos conceptos claves para la estructura de su información como lo son los dominios y las relaciones.

A.2.1 Dominios

Una vez comprendida la necesidad de integrar la información en unidades lógicas, el modelo relacional maneja el concepto de datos atómicos, los cuales son definidos como “la menor unidad semántica de información”; estos valores atómicos son denominados “valores escalares” y desde el punto de vista del modelo no pueden descomponerse en unidades más pequeñas.

De esto se puede agregar que un dato atómico, aunque este construido de elementos más pequeños, estos no son considerados como representativos dentro del modelo relacional, esto se puede visualizar en la figura 12.

FIGURA 12. Valores atómicos



Ahora se puede definir un dominio, ya que se ha comprendido la estructura mínima que deben poseer los valores o datos que el modelo relacional manipula. Un dominio se define como “un conjunto de valores escalares todos del mismo tipo. De esto se puede decir que un dominio, representa todo un universo de posibilidades, que cumplen con ciertas reglas que definen su estructura interna y da lugar aun tipo de dato atómico.

Los datos atómicos son la base de cualquier modelo de datos, por tanto entre varios modelos difieren las reglas que constituyen la estructura de dichos valores atómicos.

Por ejemplo, en el modelo de orientado a objetos, las reglas de datos atómicos son altamente complejas, ya que la existencia de objetos que integran otros objetos puede ser sumamente compleja y puede hasta cierto punto romper con el concepto de dato atómico.

Aunque la posibilidad de definir dominios en ciertos modelos es amplia, siempre existe un conjunto de datos atómicos o tipos, que ya existen por definición en el sistema que implementa dicho modelo de datos.

Como ejemplo de estos tipos ya definidos se puede mencionar valores enteros, valores reales o decimales, cadenas de caracteres u otros tipos asociados a un dominio atómico específico; muchos de estos dominios pueden poseer conjuntos finitos o infinitos de elementos atómicos

Un dominio puede ser clasificado en dos categorías: dominios simples y compuestos. Un dominio simple es un valor escalar atómico, mientras que un dominio compuesto es un conjunto de dominios simples, cuya unión genera un nuevo dominio complejo.

Debe notarse que el modelo relacional como tal permite la definición de los dominios compuestos, pero el manejo en la realidad dentro de un sistema de base de datos resulta altamente complejo de manejar.

En la realidad no existe un sistema que implemente el modelo relacional al 100%, ya que la dificultad de implementar los dominios implica costos elevados. Entonces se puede decir que, un manejo completo de dominios requerirá al menos todo esto:

- a) La capacidad de especificar un conjunto D completo de dominios.
- b) La capacidad de especificar con exactitud, para cada dominio D_i que operadores unarios se aplican valores d_i de ese dominio.
- c) La capacidad de definir con exactitud para cada par de dominios D_i y D_j cuales operadores binarios se aplican a cada par de valores d_i y d_j . Así como para cada tres dominios, cuatro, etc.
- d) La capacidad de especificar, para cada expresión escalar legal, el dominio del resultado de evaluar esa expresión.

Por último es prudente mencionar, que la importancia de los dominios es “restringir las comparaciones”, ya que existe un conjunto de valores particulares para un mismo tipo de dato.

A.2.2 Relaciones

Una vez comprendido el significado de un dominio dentro del modelo relacional, es necesario especificar la estructura de los valores o datos a los que dichos dominios hacen referencia.

Una relación “sobre un conjunto de dominios, se compone de dos partes, una cabecera y un cuerpo”. La cabecera de una relación “esta formada por un conjunto fijo parejas de atributo y dominio”.

El cuerpo de la relación “esta formado por un conjunto de tuplas, el cual varía con el tiempo”. Una tupla esta formada por “una pareja atributo y valor, para cada atributo de la cabecera”.

Una relación utiliza los términos “grado y cardinalidad” para medir el tamaño y la variación de las relaciones en el tiempo. Según estos términos “la cardinalidad varía con el tiempo, pero no el grado”.

Como se menciona al inicio de este capítulo, existen propiedades que definen a los modelos de datos, por tanto también las relaciones poseen propiedades, “las propiedades son cuatro. Dentro de una relación dada:

- 1) No existen tuplas repetidas.
- 2) Las tuplas no están ordenadas.
- 3) Los atributos no están ordenados.
- 4) Todos los valores de los atributos son atómicos.”

Las propiedades anteriormente descritas definen las reglas que deben cumplir las relaciones tanto en su cabecera como en su cuerpo. La regla numero cuatro a diferencia de las otras tres restantes, indica que “en cada posición de fila y columna dentro de la relación, siempre existe un solo valor, nunca una lista de valores”. Esta regla indica que “las relaciones no contienen grupos repetitivos”. Se dice que si una relación cumple con esta condición, “esta normalizada”.

Esta regla indica que en el modelo relacional todas las relaciones deben estar normalizadas; esta característica de normalización se analizará adelante como una debilidad del modelo relacional.

A.2.3 Llaves primarias

Una llave primaria en una relación R , esta definida como “un identificador único para esa relación”. Este identificador único permite identificar a una tupla, entre un conjunto de tuplas dentro del cuerpo de una relación.

Dentro de una relación R , existe un conjunto de “llaves candidatas”, de las cuales una y solo una pueden ser llave primaria. Una llave es candidata si solo si cumple las propiedades de “unicidad y minimalidad”.

Se dice que una llave candidata cumple con el principio de unicidad si “en cualquier momento dado no existen dos tuplas en R con el mismo valor K ” de la llave. Y finalmente una llave candidata cumple con el principio de minimalidad si el valor de la llave “ K es compuesto, no será posible eliminar ningún componente de K sin destruir la propiedad de unicidad”.

Debe tenerse en cuenta que en una relación, debe existir cuanto menos una llave candidata, ya que no deben existir tuplas repetidas. Cuando es seleccionada la llave primaria entre las llaves candidatas, las llaves candidatas restantes se denominan “llaves alternativas”.

La principal funcionalidad que una llave primaria brinda en una relación, se denota como el “mecanismo de direccionamiento a nivel de tuplas”, el cual es fundamental en un sistema relacional. Por tanto en un sistema relacional, el modo único de localizar una tupla específica es por medio del “valor de su llave primaria”.

A.2.4 Llaves extranjeras

Una llave extranjera es un atributo posiblemente compuesto en una relación, tal que dicho atributo es llave primaria en otra relación. Esto implica que una primaria sirve de referencia a un conjunto de cero o más tuplas en otra relación.

Las llaves extranjeras presentan el problema de “garantizar que la base de datos no incluya valores inválidos de una clave extranjera”, esto se conoce como “el problema de la integridad referencial”. Esto indica que los valores de una llave extranjera deben corresponder a los valores de una llave primaria en otra relación, esto se conoce como “restricción referencial”.

Otro punto importante que surge con las llaves primarias y las llaves extranjeras radica en que ambas deben ser correspondientes, entre los dominios subyacentes de las relaciones; esto implica que la relación referenciada debe concordar exactamente con los dominios en la relación donde la llave extranjera es llave primaria.

Otro aspecto importante de las claves extranjeras, es que pueden o no formar parte de la llave primaria de la relación donde estas son extranjeras. Además cuando una relación referenciada posee atributos o llaves extranjeras sobre ella misma, estos pueden presentar “ciclos referenciales”.

Finalmente, se puede agregar que una llave extranjera puede presentar en algunas condiciones valores nulos; esto se explica a continuación: supongamos que se tiene una relación con llave extranjera a ella misma, en donde se forma una jerarquía; esto implica que la referencia cíclica debe poseer un origen que no posee referencia y que necesariamente debe contener un valor nulo; esto se visualiza mas claramente en la figura 13.

FIGURA 13. Relación cíclica con valores nulos en las llaves extranjeras

<u>Nombre de la relación</u> : <i>Jerarquía Empresarial</i>		
<u>Atributos</u> :	<i>Puesto</i>	<i>Puesto Superior</i>
	Gerente General	NULO
	Gerente Ventas	Gerente General
	Vendedor	Gerente Ventas
Llave primaria = Puesto, Llave extranjera = Puesto Superior (Ambas sobre el mismo dominio.)		

A.2.5 Reglas de integridad

Ahora se pueden definir las 2 reglas fundamentales del modelo relacional, conocidas como la regla de las entidades y la regla de integridad referencial.

A.2.5.1 La regla de integridad de las entidades

La regla de integridad de las entidades se puede interpretar de la siguiente forma: “ningún componente de la llave primaria de una relación base puede aceptar nulos”.

Se sabe que una tupla en una relación, es un elemento que representa algo con existencia en el mundo real. por tanto un valor nulo es algo que cuya existencia se desconoce. y por tanto no posee aplicación en la realidad.

Existen varias justificaciones prácticas para esta regla: las llaves primarias son la forma única de direccionamiento dentro del modelo relacional, por tanto si un nulo representa algo que se desconoce, una tupla con valores nulos sería un elemento sin identidad, representando datos desconocidos.

Finalmente, se puede expresar esta regla de la siguiente manera: “en una base de datos, nunca registremos información acerca de algo que no podamos identificar”.

A.2.5.2 La regla de integridad referencial

Esta regla basada en las llaves ajenas se puede interpretar de la siguiente forma: “la base de datos no debe contener valores de llave ajena sin concordancia.”

Esta regla expresa que para todos los valores de una llave foránea en una relación, deben existir valores que concuerden como llave primaria, en la relación base. Debido a que las llaves primarias representan identificadores de entidades, las llaves extranjeras son referencias a dichas entidades, por tanto debe existir necesariamente esa relación de concordancia.

En términos simples se dice que si “B hace referencia a A, entonces A debe de existir”. Se debe de considerar además, que las llaves ajenas son necesariamente llaves primarias de otras relaciones, por tanto la existencia de esa información referenciada, debe ser entidades del mundo real existente.

A.3 El modelo relacional y su aportación

El modelo relacional ha adquirido gran importancia comercial, debido a la base teórica que lo fundamenta, este es un estándar de los sistemas de bases datos, de los últimos tiempos.

Una de sus principales características es su sencillez de representación y su amplio nivel de manipulación de información. Su estructura de tablas permite que toda información pueda ser vista de muchas formas lógicas; permitiendo éste, una amplia concepción de la información. Se dice que “el modelo relacional, fue producido con varios objetivos, entre los cuales se cuenta con el deseo de utilizar métodos formales en el diseño de bases de datos, en las consultas y en las actualizaciones”.

Otra de las principales razones que caracteriza al modelo relacional fue “por el deseo de ser capaz de probar la corrección de los programas basados en descripciones no procedimentales”. Además, se fundamentó su sencillez debido a la necesidad de “satisfacer la cuchilla de Ocam: las teorías deben ser lo más sencillas posibles, manteniendo su potencia expresiva. Las relaciones lógicas entre los datos en el modelo relacional, se construyen en el momento de la ejecución o en tablas, pudiendo almacenarse el mismo tipo de objeto para representar tanto sus relaciones como las entidades; esto permite que el modelo relacional se aplique a empresas sometidas grandes cambios organizacionales.

Es notable que en la actualidad, muchos métodos para el desarrollo de sistemas continúen haciendo hincapié en enmarcar los conjuntos de información, o bien las entidades para indicar si están bajo la presencia de cambios a lo largo del tiempo o si no lo están, es considerable que dentro de una base de datos relacional, todo puede cambiar con el tiempo, a no ser que algún suceso externo lo impida.

Este cambio en el tiempo es típico del grado hasta el cual el modelo relacional no ha sido comprendido por aquellos que lo proponen, ni tampoco por aquellos que son enemigos. La fuente principal de incomprensión es la confusión entre modelos de datos lógicos y físicos. Esto es debido a que el modelo relacional es un modelo lógico.

Por otra parte, no existía ningún modelo real, motivante, ni jerárquico, ni redes en el sentido lógico, lo cual lo llevo a ser el principal como modelo básico de datos. Los productos de bases de datos relacionales se pueden construir como bases de datos en red para mayor eficiencia.

El modelo lógico relacional permite que los usuarios puedan visualizar una estructura de datos de muchas formas distintas, mediante las denominadas vistas de usuario, y de esta manera, el beneficio más importante de las bases de datos relacionales es un mayor grado de aceptación por parte de los usuarios.

Los sistemas relacionales tienen muchas más ventajas. Estos hacen que sea sencillo aplicar cambios a la estructura de los datos, y protegen a los usuarios frente a la complejidad, mediante lenguajes de consulta no procedimentales, que pueden ser utilizados automáticamente.

Los problemas de rendimiento han sido vencidos gradualmente. Después de una resistencia inicial, las bases de datos relacionales han alcanzado ahora una aceptación tan amplia en la industria que la mayoría de los constructores ya no se molestan, siquiera, en considerar dimensiones prácticas o en red, salvo para aquellos tipos de aplicaciones que producen grandes transacciones en forma intensiva.

A.4 Debilidades del modelo relacional

Ahora se procederá a exponer algunas de las dificultades, a las cuales se enfrenta el modelo relacional. Entre estas dificultades se encuentra la dificultad de tratar con consultas recursivas, los procesos relacionados con valores nulos y la falta de apoyo para los tipos abstractos de datos.

Existen otros problemas, tales como: graves deficiencias en la representación de datos y la semántica funcional, también existen problemas de normalización, así como la falta de un apoyo real para las reglas integridad. Todos estos problemas se exponen a continuación.

A.4.1 Normalización

Desde el punto de vista real, en un mundo donde toda actividad esta determinada por objetos, la normalización presenta varios problemas evidentes.

En primer lugar, las relaciones normalizadas no suelen corresponder a ningún objeto del mundo real. Esto surge debido a que la descomposición esta sujeta por consideraciones de la computación o de la lógica, en lugar de ser guiada por la estructura de la aplicación.

Esto evita que toda relación cuya estructura sea normalizada, pueda ser reutilizable aisladamente, e impide todo progreso del ensamblaje de sistemas a partir de componentes. Esto implica que toda tabla normalizada no corresponden con absolutamente nada de la aplicación, así también la normalización oculta totalmente la semántica que esta representa.

La razón fundamental por la cual, la normalización oculta el conocimiento de la aplicación, es debido a que esta es representada por dependencias funcionales entre los atributos. Esto elimina toda posibilidad de ingeniería inversa completa y haciendo al proceso de normalización, irreversible.

A.4.2 Reglas de integridad y de gestión

En términos generales las reglas de integridad, son “reglas semánticas” que ponen de manifiesto lo que es incorrecto, siendo en el modelo relacional como ya se mencionó: la regla de integridad de entidades y la regla de integridad referencial.

Es importante mencionar que la integridad de las entidades no es lo mismo que la identidad de objetos del mundo real, porque en el modelo relacional, se identifica a una tupla por el valor de la llave, y no por una identidad modificable independiente de los valores de los atributo, haciendo a este dependiente de una cualidad y no de su propia identidad en el mundo real.

Por último, se puede decir que el apoyo que el modelo relacional brinda a cualquier tipo de reglas de gestión o integridad se realiza por medio de ediciones en sus datos y no forma parte del modelo en sí, creando así una dependencia de procesos aplicados fuera del alcance del modelo relacional.

A.4.3 Valores nulos

Un valor nulo dentro de una tupla, puede representar como ya se menciona, un valor “desconocido”, sin embargo presentan una dificultad de interpretación.

Los valores nulos pueden interpretarse como un valor que no es aplicable a tuplas concretas, o podría significar que todavía no se han proporcionado valores.

En conclusión, todas estas interpretaciones implican que los valores nulos presentan problemas de la semántica de aplicación. Debido a que “el modelo relacional es un modelo puramente sintáctico: no contiene apoyo intrínseco para ninguna clase de semántica”.

A.4.4 Tipos abstractos de datos y objetos complejos

Debido a que la primera forma normal prohíbe el almacenamiento de objetos complejos, de tal manera “no es posible tratar tipos abstractos de datos mediante el modelo relacional”; esto implica también, que los métodos no pueden ser almacenados dentro de la base de datos, para su utilización.

Esta dificultad es debido a que los datos dentro del modelo relacional son estáticos y no dinámicos, siendo estos creados en el momento en que se diseña la base de datos. Además el modelo tiene la limitante de que “los atributos de las relaciones son útiles para registrar la duración en la que es válida una relación.

A.4.5 Consultas recursivas

La primera forma normal del modelo relacional, impide las consultas recursivas, esto es, “las consultas acerca de las relaciones que una cierta entidad tiene consigo misma”.

Una forma de evitar este problema sería, “una búsqueda de jerarquía de herencia”, generando con esto, altas consultas y manejos complejos dentro del modelo relacional, que en algunos casos quedarían fuera del alcance de los usuarios.

B. CONCEPTOS GENERALES DE LA METODOLOGIA ORIENTADA A OBJETOS

B.1 Introducción

Aquí se mostrarán las principales características de la metodología orientada a objetos, y que el lector pueda tener una amplia visión del potencial de dicha metodología, orientado a cualquier sistema de información, sin importar su campo de aplicación.

Se mostrará, en forma general, qué son los métodos orientados a objetos, así como una breve descripción de su evolución en las últimas décadas, respecto a su campo de aplicación.

También se expone su principal terminología, y los conceptos básicos que fundamentan esta técnica. Cada uno de sus términos, serán expresados en forma general, según los estándares especificados por el Grupo de Gestión de Objetos, para la normalización de esta metodología.

Por último se presentan dos secciones; una de ellas tiene como objetivo el mostrar las ventajas que se obtienen al aplicar esta metodología en cualquier campo de acción, y finalmente se expone en la última sección, algunos de los problemas insospechados, en los que se incurre por el uso de esta técnica.

B.2 Que son los métodos orientados a objetos

Una de las tecnologías más populares de los últimos tiempos ha sido la metodología orientada a objetos, debido a su gran poder representativo de información.

Entre los aspectos más relevantes de esta técnica se encuentra ataque a la complejidad de interfaces, la representación compleja de información y la representación compleja problemas; entre otras también se encuentra el bajo costo de su desarrollo e implementación, y finalmente la reutilización de sus interfaces. En términos generales, los métodos orientados a objetos o tecnología orientada a objetos, se refiere a “programación, bases de datos, análisis y diseño orientado a objetos”, y su filosofía es la aplicación completa en el desarrollo de sistemas, “basada en una potente metáfora”.

Esta técnica refleja y resume la historia de la computación “como un todo”, visualizando al mundo, como un lugar en donde una infinidad de objetos interactúan entre sí, realizando tareas de toda naturaleza.

Por medio de esta metodología se puede definir cualquier actividad en términos de objetos que las realizan, describiendo estas actividades como secuencias lógicas de pasos y cuyo resultado puede ser la solución a problemas, simulación escenarios o simplemente la representación de ambientes, en donde tiene lugar los hechos de un mundo en constante cambio. Algunos de los principales motivos que contribuyó al éxito de la metodología, fue la gran complejidad de ambientes que esta podía describir en termino de unidades mínimas denominados objetos, así como el alto grado de reutilizabilidad de estos objetos para futuros problemas.

Otra característica de esta técnica, es el bajo costo que se incurre al aplicar esta metodología en cualquier campo de acción, debido a la facilidad de desarrollo y a la facilidad para adaptar, a un ambiente, sometido a cambios. Esto implicó grandes ahorros en el diseño y construcción de sistemas de cualquier naturaleza.

La metodología orientada a objetos ha evolucionado considerablemente, desde su aparición formal en 1967 en el lenguaje Simula; su evolución consta desde la programación orientada a objetos hasta la definición de estándares para dicha metodología. Esto se muestra en la tabla IX.

TABLA IX. Historia de los métodos orientados a objetos.

Fase I Años 70 Epoca de Invención	Fase II Años 80 Epoca de la confusión	Fase III Años 90 Epoca de la madurez
<ul style="list-style-type: none"> - Simulación de sucesos discretos - Simula - Kay : máquina FLEX - PARC: Dynabook - Smalltalk 	<ul style="list-style-type: none"> - Interfaces WIMP - Xerox y Apple - Extensiones LISP - Entornos de inteligencia artificial - Nuevos lenguajes: Eiffel, C++, ... 	<ul style="list-style-type: none"> - Énfasis en el análisis y el diseño - Sistemas abiertos - Aplicaciones - Base de datos orientadas a objetos - Estándares

B.3 Terminología y conceptos básicos

Ahora se definirán todos aquellos términos que forman parte de la metodología. Para esto, se emplearán los estándares establecidos por el Grupo de Gestión de Objetos, cuyo objetivo es la normalización de la teoría de objetos.

B.3.1 Objetos

La definición formal de un objeto puede considerarse como, “las unidades básicas de construcción, para conceptualización, diseño, o programación; son instancias organizadas en clases con características comunes”.

Estas características son atributos que describen las cualidades de un objeto, y sus métodos que son operaciones que los objetos realizan como parte de su funcionamiento real.

Los objetos deben representar en la medida de lo posible, entidades del mundo real y deben estar basados en los conceptos de dominio, para el tipo que representa. Un objeto en particular, representa a un ente o elemento específico de la realidad, que pertenece a un universo o conjunto de valores, y a su vez representa un ejemplar de una clase.

Un objeto puede considerarse en cierto sentido como un “tipo de dato abstracto”, que puede definir entes con alta complejidad, únicamente guiándose con las características reales de un objeto y la formas en que este hace las cosas en la realidad, básicamente se define un comportamiento.

B.3.2 Clases

Una clase representa a un conjunto o universo de elementos, todos del mismo tipo, de tal manera que todos poseen las mismas características y forma de hacer las cosas, pero no particulariza a un objeto individual; esto lo hace el objeto en sí.

Una clase a su vez puede estar formada por meta clases, o clases superiores, que definen características comunes para un grupo de clases, pero al igual que los objetos de una clase específica no puede ser particularizada, sino únicamente por sus cualidades individuales de clase.

B.3.3 Métodos

Un método es la forma o metodología con que un objeto realiza una actividad. Los métodos están integrados por un conjunto de pasos que el objeto realiza en forma oculta, para lograr una actividad que es propia del objeto.

Un método está ligado a un “mensaje”, que constituye un llamado a un dicho del objeto. La funcionalidad de un mensaje es la siguiente: la acción se inicia mediante la transmisión de un mensaje a un objeto, responsable de la acción.

El mensaje tiene codificada la petición de una acción y se acompaña de cualquier información adicional necesaria para llevar a cabo la petición”. Entonces el objeto responderá con “algún método para satisfacer la petición”.

B.3.4 Atributos

Un atributo en un objeto, representa una característica particular de un objeto específico en la realidad. Son los atributos los que hacen las diferencias entre cualquier par de objetos de la realidad. Si un objeto constituye “algo que ocupa un lugar en el espacio”, entonces los atributos son esas cualidades que hacen la diferencia de un objeto a otro.

B.3.5 Encapsulamiento

El encapsulamiento es la cualidad que tiene un objeto, de ocultar sus atributos y el detalle de sus métodos, a otros objetos.

Esta característica permite que “la única forma de acceder al estado de un objeto es enviar un mensaje, que haga que uno de los métodos lo ejecute” (Budd, 1994.57); por tanto es el propio objeto, el único que puede alterar su propia estructura o estado, a través de sus propios métodos.

B.3.5 Metaclases

Una metaclase representa una clase superior jerárquicamente, de otras clases y cuyos elementos particulares, de la metaclase lo constituyen clases de objetos. Como ya se mencionó, un objeto es un elemento particular de una clase, entonces las propias clases son elementos de una metaclase.

B.3.7 Herencia

La herencia, es la propiedad que tiene un objeto de una clase particular, de heredar características de objetos que pertenecen a clases superiores jerárquicamente. Esto hace que un objeto denominado hijo, pueda utilizar atributos y métodos de un objeto denominado padre, y utilizarlos de la misma forma que su clase superior. La herencia es la cualidad de un objeto, que permite especializarse dentro de una rama de la jerarquía, que caracteriza a esa familia de objetos.

B.3.8 Abstracción

La abstracción, según la definición “del Oxford English Dictionary, puede emplearse como: la acción de separar mentalmente”; también puede considerarse como: “la representación de las características esenciales de algo sin incluir antecedentes o detalles irrelevantes”. La abstracción trata de reunir las características comunes entre las clases, reagrupandolas, y generando una nueva clase con las características que son necesarias para un conjunto nuevo de elementos.

B.3.9 Polimorfismo

Este se conoce como “la posibilidad de emplear la misma expresión para denotar operaciones diferentes”. El polimorfismo, es un término relacionado con la herencia y el encapsulamiento. La herencia permite a un objeto, poseer los métodos de un objeto de una clase superior, ocultando las características de dichos métodos, permitiéndole ejecutarlos de igual o diferente forma que el objeto de la clase predecesora. Esto permite que un objeto realice una actividad de diferente forma, basándose en las cualidades que ha heredado de clases superiores.

B.4 Ventajas de la metodología

Ahora se mostrará algunas de las principales ventajas que presenta el uso de la metodología orientada a objetos, en cualquier campo de aplicación. Estas ventajas se hacen relevantes para usuarios de datos, desarrolladores, analistas y usuarios de sistemas en general. Una de las principales ventajas, que caracteriza a metodología es obtenidas en base a los objetos bien diseñados, siendo estos la base para otros sistemas que se ensamblan posteriormente; esta generación de módulos reutilizables, lo que redundo en una mayor productividad.

Debido a esta reutilizabilidad que existe, los proyectos posteriores cuentan con la confiabilidad que ya ha sido probada en proyectos anteriores, debido al uso de las mismas clases de objetos, teniendo como resultado "sistemas de mayor calidad, que satisfacen mejor los requisitos de los negocios y contiene menores errores".

Debe de tomarse en cuenta que entre mayor sea el número de proyectos anteriores, mayor será el grado de reutilizabilidad y más completa las librerías de objetos a utilizar, ya que estos pueden mejorarse con el tiempo.

Otra ventaja que aplica a la metodología orientada a objetos, es que la herencia relacionada a una clase en particular, hace posible "utilizar y definir módulos incompletos, permitiendo su extensión sin transformar la operación de otros módulos o de sus clientes". Esto implica que los sistemas sean más "flexibles y más fácilmente extensibles y de mantenimiento menos costoso".

El paso de mensajes constituye otra ventaja importante, esta surge con la comunicación de interfaz entre entidades y módulos externos, haciendo que se realice de una manera mucho más fácil. Esto facilita el enlace entre cualquier par de entidades de cualquier naturaleza, únicamente con la correcta indicación de mensajes entre los objetos que encabezan la conexión.

La escalabilidad representa otra considerable ventaja, debido a que al dividir los sistemas, tomando como base tipos de objetos encapsulados, ayuda a resolver los problemas a gran escala. Esto sugiere que el esfuerzo, no necesariamente tenga que aumentar exponencialmente con el tamaño y la complejidad del proyecto, haciendo que se trate una parte a la vez, sin necesariamente tener los demás elementos de dicho proyecto.

La metodología orientada a objetos se basa en una descomposición natural de la realidad, siendo esta forma más fácil, que una descomposición funcional para. Los sistemas orientados a objetos "admiten mejor el cambio de pequeña a gran escala". Con frecuencia, en el análisis y diseño que dividen un dominio de objetos, es mayormente manipulable, ya que existe como teoría la forma natural de la realidad.

Otra ventaja considerable la constituye el ocultamiento de la información, permitiendo, mayor seguridad en la información, dando como resultado un sistema más seguro. También permite capturar una mayor parte de la semántica del modelo de forma realizable. Esto es muy importante en los sistemas comerciales, debido a que estos necesitan un respaldo significativo en las operaciones que se realizan, como lo es el caso de una contabilidad y cuentas.

La orientación a objetos constituye además una herramienta para gestionar la complejidad. Además, la evolución, crecimiento y mantenimiento de un sistema, quedan mitigados por la división del encapsulamiento, y de las interfaces entre objetos.

Otra ventaja surge debido a que los métodos orientados a objetos se utilizan para describir sistemas, entonces facilitan el modelado de escenarios y facilitan cambios dentro de la empresa, haciendo que el producto final, sea más reversible y mejora la posibilidad de efectuar la ingeniería inversa, permitiendo volver a partir de las características que originaron el estado actual. Debido a la carencia de otras metodologías, para moldear escenarios, esta se ha convertido en un vínculo entre la realidad y la representación, siendo actualmente la única forma de representar y construir aplicaciones reales.

Finalmente, puede considerarse que los métodos orientado a objetos, brindan características potencialmente altas a cualquier sistema de software.

Un sistema que es diseñado con esta metodología puede ser considerado de alta calidad, según expertos en la materia de calidad. Se puede decir que un sistema de software es de calidad si cumple con 13 puntos mostrados en la tabla X;

La metodología orientada a objetos, contribuye a que un sistema de software cumpla con muchos de estos objetivos. Algunos sistemas, cuentan con mayor puntuación en algunos de estos puntos, pero el énfasis se obtiene en el carácter reutilizable, de las aplicaciones orientadas a objetos.

TABLA X. Lista de puntos de evaluación de calidad para un sistema de software

- **Corrección.**
Los programas deben satisfacer sus especificaciones correctamente.
- **Flexibilidad y fiabilidad (Solidez).**
Los programas deben ser sólidos, aun en condiciones anormales.
- **Facilidad de mantenimiento.**
Los programas deben ser fáciles de modificar y de extender cuando se produzcan cambios en los requisitos.
- **Reutilizabilidad y generalidad.**
Los programas deben construirse con módulos reutilizables.
- **Interoperabilidad.**
Los programas deben ser fácilmente compatibles con otros sistemas; deben ser "sistemas abiertos".
- **Eficiencia.**
- **Transportabilidad.**
Los programas deben ser transportables de un equipo (hardware) a otro, y de un sistema operativo a otro.
- **Verificabilidad.**
- **Seguridad.**
Los datos, los conocimientos e, incluso, las funciones pueden requerir un ocultamiento selectivo y efectivo.
- **Integridad.**
Los sistemas requieren protección contra actualizaciones inconsistentes.
- **Amabilidad**
Los programas deben ser fáciles de usar para la mayoría de los usuarios, sin llegar a pecar de excesivos en palabras.
- **Facilidad de descripción.**
Debes ser posible crear y mantener documentación de los programas
- **Inteligibilidad.**

B.5 Problemas y peligros de la metodología

Como toda metodología, siempre existen problemas insospechados que surgen en la aplicación de dicha técnica. En esta sección se presentará algunos de los problemas más comunes, que surgen en la aplicación de los métodos orientados a objetos.

Uno de los principales problemas que surge, es el aumento en el costo, debido a la reutilizabilidad. Esto surge a causa de la creación estructuras reutilizables, que deben ser diseñadas de la manera más general posible, para que tengan aplicación en proyectos futuros, y tengan un considerable efecto respecto a atrasos y costos, en el proyecto actual.

Entonces, el verdadero ahorro que brinda la metodología orientada a objetos, respecto a reutilizabilidad, "puede ser justificada en términos de ahorros futuros". Implica entonces, que los ahorros reales que se obtienen, son producto de proyectos elaborados en el pasado. Esto tiene el peligro de inversiones presentes, con incertidumbre de su futura utilización, no garantizándose que serán un ahorro posterior.

Otro problema relacionado, surge debido a la falta de bibliotecas de objetos disponibles comercialmente. Verdaderamente si no se dispone, de bibliotecas, que hayan sido desarrolladas en proyectos reales, no se puede proporcionar a los usuarios los beneficios de la reutilización.

Debido a que las bibliotecas, son diseñadas por un sin fin de usuarios, surge una sobrecarga de información, haciendo que sean demasiadas las bibliotecas, y esto da origen a que los usuarios no sepan que información se encuentra en cada una, haciendo no eficiente el uso de estas librerías.

Surge, entonces, la necesidad de administración de bibliotecas, para componentes de métodos orientados a objetos, siendo este otro problema de manejo de información, debido a quien se hará cargo de dicha información.

Otro problema surge debido a la reutilizabilidad. Este se presenta cuando existen modificaciones o arreglos de urgencia, que son realizadas dentro de un objeto. Esto repercute, en todo aquello que haga referencia al objeto, tanto en las anteriores, como en las presentes actividades de dicho objeto.

También surge el problema de la carencia de protocolos necesarios para la construcción de objetos, control de cambios y modificaciones en las versiones. Serán necesarias entonces, herramientas que controlen toda esta carga de información y controlen todos los enlaces y relaciones entre objetos, con el objetivo de comprender la influencia del cambio de un objeto sobre otros.

Otro problema surge debido a la herencia. Esta compromete la reutilización, a no ser que se sigan procedimientos estrictos para su control. Esto implica que la herencia en general, requiere un alto grado de creatividad para la generalización de objetos de clases, clases y jerarquías, que involucren dominios, para las aplicaciones construidas para modelarlos.

También surge el problema del paso de mensajes. En ocasiones es necesario seguir el comportamiento de un sistema, para depurarlo, siguiendo el paso de mensajes entre un objeto y otro, siendo una tarea altamente dificultosa. Esto implicaría un seguimiento de "estructuras de paso de mensajes como un plato de espagueti".

Como toda nueva tecnología, implica nuevas inversiones respecto a hardware, software y métodos necesarios, para el entrenamiento inicial, reeducación, tanto para los que desarrollan los sistemas como para la gerencia, además como ya se mencionó la administración de bibliotecas y el desarrollo de componentes reutilizables.

Todo esto incurre en un alto costo, para el desarrollo inicial de aplicaciones. Debido a que en la realidad, se paga un costo para que los proyectos finalicen a tiempo, y no para que se construya código que beneficie a otros proyectos, implicará decisiones para dicha implantación de la metodología orientada a objetos.