

### 2. 1 Modelado de Datos

El manejo de información implica el saber como organizar los datos. Un apoyo lo encontramos en las herramientas de bases de datos que a su vez se apoyan en el modelo de datos. Para que éste se encargue de organizar y manipular los datos. En los sistemas de bases de datos, encontramos a los manejadores de bases de datos, que son lo que nos ayudan a almacenar la información y recuperarla mediante la realización de preguntas (queries) [Khoshafian 95]. ¿Pero qué tipos de modelados de datos podemos encontrarnos?, entre ellos nos encontramos los siguientes:

- Modelado de datos Relacional
- Modelado de datos Orientado a Objeto
- Modelado de datos Objeto-Relacional

Estos tres modelos serán analizados y definidos a continuación así como lo que son las bases de datos, para tener un mejor conocimiento del tema e irnos adentrando en la elaboración del modelo de datos a implementar en este trabajo.

### 2.2 ¿Qué es una base de datos Orientada a Objetos?

Algunos autores usan el término 'orientado-objetos' para sistemas de bases de datos sin los requerimientos de encapsulación, basado simplemente en el grupo de datos con objetos [Cattell 91].

La orientación a objetos permite una representación más directa y la modelación de problemas del mundo real. A través de la construcción orientada a objetos, el usuario puede disimular los detalles de implementación de sus módulos, objetos referenciales, y extensión de sistemas a través de especialización en módulos existentes. La funcionalidad de la base de datos es

necesaria para asegurar la persistencia o concurrencia de la información en las aplicaciones. Por medio de las bases de datos, los usuarios pueden tener el estado de objetos persistentes y actualizarse por medio de varios programas invocados. Además, vía la base de datos varios usuarios pueden compartir la misma información en forma concurrente. Las bases de datos orientadas a objeto combinan el beneficio y concepto de lo orientado a objeto con una base de datos funcional.

Las bases de datos orientado a objetos son iguales a [Khoshafian 95]:

- persistencia más
- concurrencia más
- transacción más
- recuperación más
- queries más
- versiones más
- integridad más
- seguridad más
- rendimiento.

Un ejemplo de este modelo podemos verlo en la Figura 2.1

Por lo que las bases de datos orientadas a objeto son igual a orientación mas bases de datos competentes.

Pero lamentablemente todavía no se encuentran implementadas por lo que la falta de estándares provoca que los realizadores de sistemas utilicen su propio modelo para la resolución de su sistema.

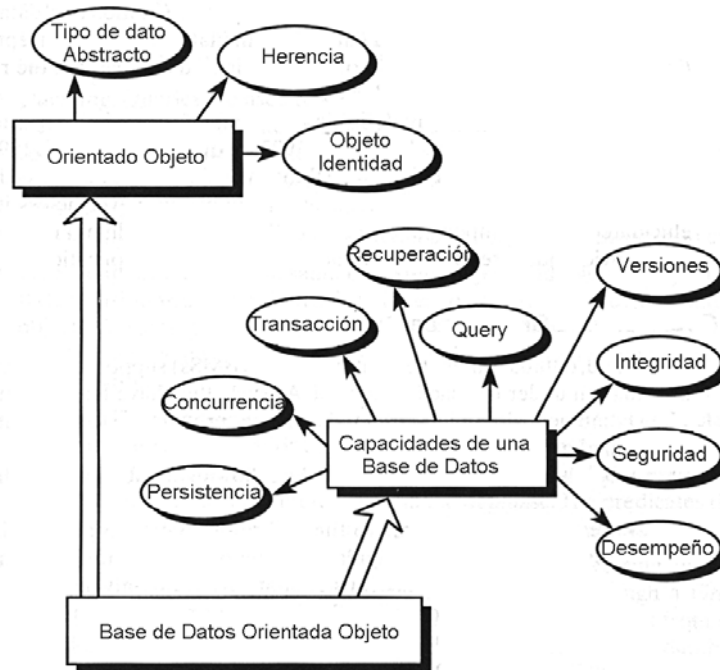


Fig. 2.1 Base de datos Orientada a Objeto

## 2.3 Modelos

### 2.3.1 Modelo relacional

A fin de que se provea mayor flexibilidad en organización de grandes bases de datos y aligerar algunos de los problemas de los primeros modelos a principios de 1970 Ted Cood introdujo el modelo de datos relacional. Después de varias investigaciones en implementación y subsecuentes comercializaciones por grandes compañías como IBM, el modelo de datos relacional comenzó y continuó siendo muy popular [Khoshafian 95].

Los lenguajes de query relacionales tales como *Structure Query Lenguaje* (SQL) son más declarativas que otros modelos y lenguajes. Con SQL el usuario indica que busca en la base de datos un estilo de alto nivel declarativo de programación, específicamente que es lo que se accesa a la base de datos mejor dicho *como accesarlo*. El modelo relacional es simple y elegante. Esta teoría esta

basada en lo matemático, fundada en conceptos de álgebra relacional y el cálculo de predicado de primer orden. El álgebra relacional consiste de algunas operaciones como: el conjunto de operaciones (unión, intersección, diferencia, producto cartesiano) y las operaciones relacionales (selección, proyección, join).

El modelo relacional es de gran importancia, porque es popular en la comunidad de investigación y en negocios comerciales de procesamiento de datos, donde el modelo relacional está supliendo a sistemas basados en modelos recientes. Por lo que, el modelo relacional es un buen lugar para comenzar a manejar nuestros datos, sirviendo como una utilidad para la comparación de más sistemas avanzados.

La popularidad del modelo relacional se debe primariamente a este modelo simple. Hay únicamente una sola estructura de datos: una tabla con renglones y columnas conteniendo datos de tipo específico, tales como *integer* o *string*. El lenguaje de query está basado en simples operaciones con tablas, y las más complejas características (tales como la integridad limitada en datos y definición de vistas, ocultar o reestructurar datos) que no necesitan ser entendidos por el usuario [Cattell 91].

¿Y como utilizamos dicho modelo en un SIG? La respuesta no resulta sencilla, ya que como se mencionó en el párrafo anterior la estructura de el modelo relacional está formada por tablas con renglones y columnas, así que la información es vaciada a cada renglón en forma ordenada para que en el momento de la recuperación sea compatible la información obtenida. Además de que la tabla que se genere dependerá de cuanta información es la que se esté manejando en ese momento y una vez que ya se recuperó dicha información hay que estructurarla en el ambiente de programación que se esté utilizando. SIG son manejados por capas, más adelante se explicará que es una capa, por ejemplo: tenemos un territorio, este puede estar compuesto de ríos, carreteras y ciudades.

Cada uno de estos componentes es una capa, por supuesto independientes unas de otras, así que al momento de guardar toda esta información se generan tantas tablas, como capas se tengan, y estas a su vez generan renglones y columnas que se necesiten. Como podrá verse el trabajo es muy repetitivo y hasta cierto punto dificultoso, por lo que al llegar a manejar volúmenes de información demasiado grandes este modelo resulta ser dificultoso para un SIG.

### 2.3.2 Modelo Orientado a Objeto

Los sistemas basados en modelos de datos orientados a objeto se originaron con el paradigma de programación orientada a objeto. Los lenguajes de programación frecuentemente mencionan como originar y popularizar el paradigma antes mencionado, tales como Smalltalk y Simula. Ejemplos más recientes son C++, CLOS y CLU.

El paradigma de programación orientada a objeto incluye el concepto de tipos de datos abstractos en lenguajes de programación. Las declaraciones de tipos de datos abstractos explícitamente se definen públicos y privadas en algunas porciones de la estructura de datos, u *objetos*. Los tipos de datos abstractos en lenguaje orientado a objeto, son llamados *clases*, es decir *encapsula* porciones privadas de datos del objeto con procedimientos públicos, llamados *métodos*. El argumento para encapsulación es uno de los más simples en la construcción y mantenimiento de programas a través de modularización. Un objeto es como una caja negra, que puede ser construida y modificada independientemente del resto del sistema, tan grande como una interfaz pública (método) en la cual las definiciones no cambian.

No hay un sólo paradigma orientado a objeto, y por lo tanto hay una variedad de modelos y como consecuencia diferentes estándares [Khoshafian 95]. Generalmente, los lenguajes de programación orientados a objeto parten de

conceptos comunes además de encapsulación, en particular el uso de jerarquías de tipos de objetos con herencias en sus atributos y métodos. De cualquier modo, las características específicas varían, y pueden regular la definición estricta de encapsulación provista por tipos de datos abstractos - que los procedimientos son públicos, cuando los datos son privados; es también restrictivo para los DBMS's orientados a objeto.

### 2.3.3 Modelo Objeto-Relacional

El camino más obvio que remedia errores del modelo relacional es el *modelo extendido o modelo objeto-relacional*. El modelo de datos "Extended Relational" es usado en sistemas basados en el modelo de datos relacional incorporando procedimientos, objetos, versiones y otras nuevas capacidades. Ya que los sistemas de base de datos son simples, más comprensibles y populares en búsquedas y productos, los beneficios del sistema relacional extendido vienen de ésta búsqueda y popularidad.

No hay un sólo modelo relacional extendido; mejor dicho, hay una variedad de éstos modelos, cuyas características dependen del camino, y el grado en el cual las extensiones son realizadas. Todos los modelos hacen parte del relacional (básico) tablas y lenguaje de query, todos incorporados al mismo concepto de "objeto", y todos tienen la habilidad de almacenar procedimientos de la mejor manera como datos en la base de datos [Cattell 91].

¿Pero como utiliza este modelo un SIG? Los SIG tradicionalmente han sido caracterizados por modelos de bases de datos relacionales. Los SIG se basan en capas. Cada capa representa un concepto que tiene una semántica bien definida, por ejemplo los ríos. Cada capa puede representa una de las figuras básicas (*punto, línea o polígono*) y representándolos en escalas sencillas. El término 'escala' en éste contexto es debido al nivel de generalización de los datos, no

debe ser confundido con el significado tradicional de escala que se conoce en Cartografía. Aunque puede usar procedimientos de generalización para derivar entidades a escala pequeña de entidades que son "large-scale/detailed", el problema de redundancia e inconsistencia no puede ser evitado.

Para resolver el problema mencionado anteriormente en base de datos, los SIG convencionales crean capas separadas para cada escala, comenzando del nivel más cercano, posteriormente la construcción y finalmente el *floor plans*. La capa floor plans debe ser fácilmente disponible para cada construcción, lo que quiere decir que el tamaño de la base de datos es incrementada por datos similares que pertenecen a uno sólo de estos tipos de datos [Agung 99]. Esto es sólo un ejemplo, por lo que en situaciones diferentes ello puede ser mejorado, para adaptarse lo más posible a la solución del problema.

### 2.4 Base de datos Relacional Vs. Orientada a Objetos

El modelo relacional se convierte en un modelo apto para bases de datos a fines de 1970. El modelo ofrecía algunas ventajas sobre sus predecesores. Entre estas ventajas están: la sencilla y entendible teoría capaz de crear tablas virtuales dinámicamente como si fueran en el aire y el estándar de lenguaje de queries.

Una debilidad importante del modelo, es que, el modelo dificulta en expresión la semántica de objetos complejos. Las bases de datos relacionales tradicionalmente han sido implementadas para manejar la construcción de tipos de datos sencillos (número, texto, fecha o booleanos). Y como la complejidad de los objetos se incrementan, los DBMS's deben ser capaces de manejar tipos de datos definidos por el usuario y tipos no convencionales, tales como imagen, videos y datos espaciales.

Después de 1970, otros modelos fueron propuestos debido al direccionamiento (cuestión, número) de objetos semánticos en bases de datos. Dos de ellos son el modelo Entidad-Relación y el modelo funcional. No obstante, este último modelo nunca ha sido completamente operacional. El modelo Entidad-Relación comenzó a usarse como una herramienta de análisis y diseño, mientras que, el modelo funcional permanece como un modelo fuerte en teoría pero limitado en implementación, aún se sostiene un futuro prometedor para desarrollos nuevos.

El único carácter del modelo relacional comparado con otros modelos es el énfasis en la computación lógica, es decir, en el orden de procesamiento eficiente. La información clave a cerca de un objeto en base de datos relacional es llevada por el valor de un campo en cada tupla (un renglón de cada tabla). Cada tupla puede ser entendida de forma independiente. El resultado de una operación de base de datos relacional da siempre el mismo tipo de datos como el tipo de datos de entrada.

Un paso crítico en la implementación del modelo de datos relacional es llamado *normalización*, es un proceso que consiste de la descomposición de tablas para un procesamiento eficiente, como en el caso de la primera forma normal que nos dice que la información debe ser atómica. Desde la perspectiva de sistemas orientados a objetos, este proceso crea algunos problemas. Primero la relación como resultado de la normalización no corresponda con el objeto del mundo real. Segundo, no hay un camino para volver a reutilizar parte de la relación, y el proceso es irreversible una vez que la relación es mapeada fuera, es imposible desmontarlo en componentes significativos.

El cambio significativo en la estructura de la base de datos aparece con la interacción de la programación orientada a objeto (Oriented Object Programming OOP). Una de las poderosas características con las que cuenta, es la *abstracción*,



la habilidad de distribuirse con objetos complejos y encapsulación con métodos y propiedades. Otra característica es la *herencia* (la cual será descrita en detalle en el capítulo III), el mecanismo automático de reutilización de métodos y datos de uno a otro objeto, y *objeto identidad*, acerca de un objeto que permanece invariable a través de todas las modificaciones en este estado.

En los recientes desarrollos OOP, hay un número de debilidades en las cuales las bases de datos orientadas a objeto están probando soluciones. Estas bases de datos básicamente cubren las tres características mencionadas antes y otras solamente algunas. Una crítica que puede hacerse es el manejo de objetos persistentes. En OOP, un objeto es creado y destruido durante un tiempo transcurrido. Este puede no ser un problema para una aplicación con objetos en un ciclo de vida corto. Pero para un objeto de aplicación intensiva, hay objetos persistentes, objetos que continúan existiendo aún después terminada la sesión. La pregunta es si el objeto se almacena en una base de datos convencional (normalización) o se almacena en la forma de la estructura de un objeto [Angun 99].

Resumiendo, la evolución estructurada de las bases de datos del relacional al modelo orientado a objeto es básicamente intentar capturar la semántica de objetos del mundo real. El beneficio potencial de algunos métodos son: soporte de datos complejos, la reutilización automática de métodos y datos, aumentando el rendimiento de la base de datos, hacer modelos y diseños de procesos más cognitivos y fáciles.

Características que son importantes mencionar en el modelado de datos orientado a objetos son los siguientes:

- Generalización
- Asociación
- Agregación

- Clasificación

La cuales serán definidas a continuación.

## 2.5 Conceptos Orientado a Objeto

### 2.5.1 Generalización

La *generalización* es la relación entre una clase y una o más versiones refinadas de esa misma clase [Rumbaugh 96 et. al]. La que se está refinando se denomina la *superclase* y cada versión refinada se denomina una *subclase*.

Existen dos caminos por los cuales la generalización puede ser afectada [Hughes 91]:

- Propiedades comunes y funciones de un grupo de objeto similares son agrupadas junto a una nueva generación de tipos de datos.
- Subtipos de un tipo de objeto dado pueden ser definidos usando predicados, forzando los valores de los atributos.

La generalización es una construcción útil tanto para el modelado conceptual como para la implementación. La generalización facilita el modelo estructurando las clases, capturando de forma concisa lo que es similar y lo que es distinto con respecto a las otras clases.

### 2.5.2 Asociación

La *asociación* permite hacer referencia de un objeto a otro, es decir describe un grupo de enlaces con estructura y semánticas comunes. Estas suelen aparecer como verbos en la definición del problema. Las asociaciones describen un conjunto de objetos potenciales. Por ejemplo: una persona *Trabaja para* una

compañía. Todos los enlaces de cada asociación conectan objetos procedentes de las mismas clases [Rumbaugh 96 et. al].

### 2.5.3 Agregación

Una *agregación* es la relación "parte-todo" o "una-parte-de" en la cual los objetos que representan los *componentes* de algo se asocian a un objeto que representa el *ensamblaje* completo.

La *agregación* es una forma de asociación fuertemente acoplada con una semántica especial, tal como el cierre transitivo y la propagación de valores de atributos. La agregación suele encontrarse frecuentemente en problemas de listas de embarque o desglose de piezas [Rumbaugh 96 et. al].

Las agregaciones se dibujan parecidas a los enlaces de las asociaciones nada más que se le agrega un pequeño rombo que indica el extremo de ensamblaje en la relación.

### 2.5.4 Clasificación

Se refiere a los objetos de la misma estructura de datos (atributos) y comportamiento (operaciones) que se aglutinan para formar una clase [Rumbaugh 96 et. al].

## 2.6 Conclusiones

Ahora sabemos que tipos de modelos de datos existen, como son el modelo relacional, el orientado a objeto y el objeto relacional y como cada uno de ellos maneja la información así como ventajas y desventajas que nos ofrecen cada uno de ellos. Fueron también presentados algunos de los conceptos orientados a

## **Modelado de Datos**

---

objeto que serán tratados a lo largo de éste trabajo. Para darnos una idea mas concreta de lo que se refiere orientado a objeto, en el siguiente capítulo encontraremos definiciones y características de la Tecnología orientada a objeto.