

# Una evaluación del del modelo entidad relación extendido

**Hossein Saiedian**

Departamento de Ciencias de la Computación, Universidad de Nebraska de Omaha. NE 68182-0500, USA

Recibida el 29 de mayo 1995; revisada 11 de diciembre 1996; aceptada 18 de diciembre de 1996

(Traducido por Eduardo Jara G.)

---

## **Resumen**

El Modelo Entidad Relación (ER) permite desarrollar un diseño de base de datos en un esquema de alto nivel conceptual sin considerar los problemas de bajo nivel como la eficiencia, el modelo implícito del administrador de base de datos o las estructuras físicas de los datos. El Modelo Entidad Relación se hizo muy popular para el diseño de base de datos y es usado extensivamente. Para aumentar su poder de expresión, muchos investigadores han introducido o propuesto ciertas extensiones a este modelo. Algunas de estas extensiones son importantes, mientras que otras agregan poco poder de expresión, pero proveen características auxiliares. Puesto que el Modelo Entidad Relación es ampliamente usado, es importante conocer qué extensiones han sido propuestas para este modelo y qué ofrecen estas extensiones a los usuarios. El objetivo de este artículo es estudiar las extensiones al Modelo Entidad Relación más importantes y evaluar sus méritos. Resaltamos que detrás de las diferencias sintácticas de algunas extensiones hay un enriquecimiento de la semántica de las relaciones entre las entidades. También resaltamos la cercanía entre el Modelo Entidad Relación y el modelamiento de datos orientado objeto. © 1997 Elsevier Science B.V. © 1997 Elsevier Science B. V.

**Palabras Clave:** modelamiento conceptual; Modelo Entidad Relación Extendido; modelo de objeto.

---

## **1. Introducción**

En 1976 Peter Chen Publicó el Modelo Entidad Relación (MER) original que proveía un enfoque visual fácil de usar del diseño lógico de la base de datos [1]. El modelo es comprensivo, elude la complicaciones de almacenamiento y consideraciones de eficiencia, las cuales son reservadas para el diseño físico de la base de datos. En las dos décadas siguientes, muchos otros han adoptado el modelo original y lo han usado estusiastamente con cambios pequeños. Además, algunos autores han extendido el modelo incrementando sus capacidades haciéndolas más apropiadas a sus requerimientos particulares. La más comprehensiva extensión incluye iconos dinámicos como una adaptación al modelamiento de base de datos orientada a objetos.

En el prefacio de su artículo, Chen decía “El enfoque entidad-relación provee una metodología comprehensiva y fácil de entender para un diseño lógico de la base de datos independiente del almacenamiento o consideraciones de eficiencia” [1]. El problema resuelto por él con el MER es la complejidad del diseño lógico de una base de datos. El proceso convencional de diseño de una base de datos consiste en el mapeo de la información del mundo real directamente a un esquema del usuario, esquema que es específico para cierto tipo de sistema de administración de base de datos (DBMS). El accionar del diseñador está restringido a tipos limitados de estructuras de datos, consideraciones de rutas de acceso y eficiencias de recuperación y actualización, y debe

producir un esquema de usuario tomando en cuenta todas esas consideraciones. El resultado puede ser un esquema de usuario difícil de entender y modificar. El enfoque MER simplifica este proceso introduciendo unos diseños intermedios llamados vistas de negocio o esquemas de negocio. El esquema de negocio, que se expresa como un diagrama entidad relación, es un diseño conceptual de la base de datos y es una representación pura del mundo real y además es independiente del almacenamiento y las consideraciones de eficiencia. Este esquema de negocio puede más tarde ser trasladado hacia un esquema de usuario específico de un DBMS. Estas dos fases simplifican el proceso y lo hacen más organizado. El esquema de negocio es fácil de diseñar, y, en caso de una transición desde un tipo de DBMS a otro puede ser remapeado a un esquema de usuario adecuado al nuevo DBMS.

El MER es extensamente usado durante el análisis de requerimientos y para un modelamiento conceptual de la base de datos. Debido a su simplicidad, es más fácil de entender por individuos no técnicos. Pruebas en el ambiente de un mundo real han mostrado que es una efectiva herramienta de comunicación entre diseñadores de base de datos y usuarios finales.

### *1.1. Componentes del modelo entidad relación*

Los componentes principales del MER son los *tipos-entidad*, *tipos-relación* y los *atributos*. Una entidad es definida como una “cosa” la cual puede ser identificada como única. Esta puede ser una persona, un ítem, o un concepto sobre el cual una organización desea guardar datos. Las entidades que comparten propiedades similares pueden ser clasificadas en tipos-entidad, tales como EMPLEADO y DEPARTAMENTO. Las entidades pueden tener ciertas relaciones unas con otras, estas relaciones pueden ser clasificadas en tipos-relación. Por ejemplo, ADMINISTRA es un tipo-relación entre los tipos-entidad EMPLEADO y DEPARTAMENTO. La relación puede ser uno-a-uno, como CASADO lo es entre dos entidades PERSONA, uno-a-muchos como ENSEÑA entre PROFESOR y varias entidades CURSO, o muchos-a-muchos como TRABAJA-EN entre muchas entidades EMPLEADO y muchas entidades PROYECTO. En el modelo de Chen, las entidades y relaciones tienen propiedades, llamadas atributos. Por ejemplo, EDAD es un atributo de la entidad EMPLEADO y HORAS-TRABAJADAS es un atributo de la relación TRABAJA-EN entre EMPLEADO y PROYECTO. Un atributo puede tomar valores de un cierto tipo de datos. Un atributo multivaluado puede tener más de un valor. Por ejemplo, el GRADO-ACADÉMICO de una entidad PROFESOR.

Cada entidad debe tener un único identificador que la distinga de las otras entidades del mismo tipo. Este podría ser un atributo ya en uso, como el nombre de empleado, o podría ser un atributo agregado por su unicidad, como el número de identificación nacional del empleado. Chen compara el identificador de la entidad con el concepto de *clave primaria* en las bases de datos convencionales. Las relaciones son identificadas por el uso de los identificadores de todas las entidades involucradas en la relación. En el caso de una relación que involucra entidades del mismo tipo, se deben asignar “roles” como, por ejemplo, ESPOSO y ESPOSA en la relación MATRIMONIO. Una entidad puede depender de entidades de otros tipo-entidad para su existencia. En este caso Chen habla de entidad

“débil”. Un ejemplo es CARGA-FAMILIAR (de un empleado), una carga familiar de un empleado no sería de gran interés si el empleado renuncia a la compañía. Una entidad tiene un “dependencia-ID” con respecto a otra entidad, si no tiene su identificador y sólo puede ser identificada unívocamente por su relación con la otra entidad. Por ejemplo, una ciudad solo puede ser unívocamente identificada dentro de un país o estado particular.

### *1.2. Diagramas Entidad Relación*

En el MER original, un tipo-entidad es representado por un rectángulo con el nombre del tipo-entidad dentro de él. Un tipo-relación es representado por un diamante, con el nombre de la relación dentro. Tipos-entidad relacionados están conectados al diamante por líneas rectas. Cada línea es marcada con un “1”, “N” o “M” para indicar relaciones del tipo 1:1, 1:N o M:N. Un tipo-entidad débil es encerrado dentro de un rectángulo de doble línea, se coloca una “E” en el diamante del tipo-relación y una flecha apunta hacia el tipo-entidad débil. El rectángulo de doble línea también es usado para un tipo-entidad dependiente-ID, con un “ID” en el diamante del tipo-relación y una flecha hacia la entidad dependiente. Todo esto está en el “dominio conceptual superior” del diagrama. (ver fig. 1). Los atributos y sus tipos de datos son mostrados en el “dominio conceptual inferior”. (ver fig. 2). Un tipo de dato de un atributo es representado por un círculo con el nombre del tipo de dato adentro, conectado por una flecha a su tipo-entidad. El nombre del atributo se añade a la flecha a menos que el nombre sea el mismo que el nombre del tipo de dato. Un ejemplo de nombres diferentes es un tipo de dato FECHA el cual es usado para el atributo de FECHA-DE-NACIMIENTO de una entidad EMPLEADO. Los atributos multivalorados son indicados poniendo “1:N” cerca de la flecha de conexión.

### *1.3. Modelamiento de datos con énfasis en las relaciones*

El MER original fue propuesto para proveer una visión unificada de los datos[1]. Como indicaron Hull y King [2], a principios de los 70’ dos líneas de investigación opuestas se iniciaron en base de datos. El modelo relacional revolucionó el campo al separar la representación lógica de los datos de la implementación física. Los modelos semánticos fueron introducidos en primer lugar como herramientas para el diseño de esquemas. El énfasis de los modelos semánticos iniciales era modelar cuidadosamente las relaciones que frecuentemente aparecen en las aplicaciones típicas de bases de datos. Consecuentemente, los modelos semánticos son más complejos que el modelo relacional y alientan una visión más navegacional de las relaciones entre los datos.

El MER fue el primer modelo semántico centrado alrededor de las relaciones más que en los atributos. EL MER ve el mundo como un conjunto de entidades y relaciones entre estas entidades. La filosofía subyacente es que un atributo es sólo un simple hecho acerca de una entidad, mientras que una relación puede modelar la construcción de entidades más complejas a partir de otras entidades.

En los años recientes el modelamiento de datos orientado a objetos se ha hecho muy popular, pero el modelamiento entidad relación es todavía muy popular y ha descubierto muchas nuevas aplicaciones. La continuación de la conferencia internacional sobre MER

así como muchos libros y artículos dedicados al MER en los años recientes da testimonio de la importancia del enfoque ER. Entonces, es importante un examen de la evolución del MER para comprender por qué es tan popular. En este artículo examinaremos y compararemos varias extensiones del modelo entidad relación, resaltando como las extensiones sintácticas han sido hechas para ampliar el poder de representación y de modelamiento del MER original.

#### *1.4. Organización*

El poder de modelamiento del MER depende fuertemente de los diagramas entidad relación. Es por esto que en este artículo nuestra discusión sobre la evolución del modelamiento ER estará fuertemente centrado en la evolución de los diagramas ER. No puede lograrse un buen entendimiento de la semántica involucrada en el MER sin un cuidadoso estudio de la representación sintáctica en los diagramas ER. Detrás de un simple cambio sintáctico puede haber una importante mejora semántica, un cambio de significado, o algunos otros cambios cruciales en el modelamiento de datos. Un buen ejemplo es el significado de la relación ternaria. Sintácticamente, una relación ternaria es muy similar a una relación binaria; sin embargo, hasta muy recientemente no existía [3] un análisis detallado de las combinaciones de cardinalidades binarias/ternarias. Al parecer diferencias sintácticas menores pueden ser la punta de grandes icebergs semánticos; de ahí que no deberían ser pasadas por alto. Cuando sea apropiado, usaremos un esquema universitario (o sus variaciones) como ejemplo para comparar las diferentes extensiones. Sin embargo, algunas veces tendremos que usar diferentes ejemplos para una mejor ilustración del tópico tratado.

El resto del artículo está organizado como sigue. En la sección 2 entregamos una breve descripción de los mejoramientos al MER original. Este incluye una inspección de la terminología añadida y las variaciones de los diagramas. En la sección 3 describimos varios MER extendidos e incluimos diagramas que incluyen sus rasgos más importantes. La sección 4 provee un resumen similar de los modelos ER orientados a objetos. En la sección 5 se entrega un breve bosquejo de diagramas de esquemas orientados a objeto. Junto con la presentación de los modelos extendidos vamos evaluando el mejoramiento al modelo original; y, finalmente, en la sección 6 cerramos el artículo delineando algunas conclusiones.

Asumimos que el lector está familiarizado con el concepto de sistemas de base de datos. También es deseable algún conocimiento de conceptos de orientación a objetos.

## **2. Expansiones del modelo entidad relación original**

Autores más recientes han añadido explicaciones y usado diferentes términos para describir el modelo original de Chen. Algunos de éstos se muestran a continuación.

### *2.1. Objetos y clases*

Batini et. al. [4] hablan de tipos-entidad y tipos-relaciones como clases de objetos. De acuerdo a ellos, un tipo entidad representa una clase de objetos del mundo real así como un

tipo-relación representa un agregado de uno o más tipos de entidades. Ellos introducen el término “rings” (anillos) para describir “una relación binaria que conecta una entidad consigo misma”, la cual es llamada relación recursiva por otros autores. Teorey et al. [5] se refieren a “tres clases de objetos: entidades, atributos y relaciones” de Chen. Ellos explican, que “las entidades eran los principales objetos acerca de los cuales se recolectaba información y usualmente se referían a una persona, lugar, cosa o evento de interés informacional. Los atributos eran usados para detallar las entidades otorgándoles propiedades descriptivas como un nombre, color y peso”. En referencia al “MER básico” de Chen, Markowitz y Shoshani[6] dicen, “los objetos atómicos son llamados entidades. Las asociaciones entre entidades son representadas por las relaciones”. Y agregan que “los objetos son calificados por atributos y son clasificados dentro de grupos de objetos”. Esto ayuda a entender el modelo básico en términos de objetos y clases ya que esta terminología será usada para explicar conceptos avanzados.

## *2.2. Las restricciones de las relaciones*

En el modelo original de Chen, la conectividad de una relación especifica el mapeo de asociaciones entre ocurrencias de entidades. Los valores de la conexión son o “uno” o “muchos”. Teorey et al.[5] definen la cardinalidad como el número real asociado con el término “muchos”. Como Chen, ellos usan una notación 1:1, 1:N, M:N en su modelo. Batini et al.[4] usan una notación más específica que muestra el número mínimo de cardinalidad, llamado “min-card”, y el máximo llamado “max-card”. Estos son expresados en el diagrama como (min-card, max-card). Por ejemplo, (1, 5) significa que una entidad debe participar en un mínimo de 1 y en un máximo de 5 instancias de la relación en cualquier momento. El mínimo de 1 también implica que la participación de la entidad en la relación es obligatoria. Otro ejemplo es (0, 10) en donde 0 significa que la entidad no requiere participar en una relación. El valor de  $n$  para max-card significa “sin límite”. Esta convención es usada también por Navathe et al.[7]. Czejdo et al.[8] usan una notación similar, excepto que usan un asterisco en lugar de  $n$  para representar que el número máximo no tiene límite.

## *2.3. Atributos compuestos*

Como es indicado por Hull y King[2], la forma de representar un atributo compuesto en el modelo original es usando un tipo-entidad y un tipo-relación. Su ejemplo es DIRECCIÓN, que es una entidad que tiene como atributos CALLE, CIUDAD y CÓDIGO POSTAL. DIRECCIÓN esta asociada con otro tipo-entidad, PERSONA, por el tipo-relación VIVE-EN. Batini et al. [4] agregan un atributo compuesto para su modelo. Ellos también usan DIRECCIÓN como ejemplo y muestran en su diagrama el atributo compuesto en un óvalo. Cada atributo simple (CALLE, CIUDAD, etc.) es mostrado por un pequeño círculo (con nombre) conectado por una línea corta al óvalo, el que está conectado por una línea al tipo-entidad PERSONA y el par (min-card, max-card) es mostrado para el atributo compuesto y no para cada atributo simple. Elmasri y Navathe [9] también permiten un atributo compuesto en su modelo básico, mostrando una fila de atributos, cada uno representado por un óvalo con el nombre adentro.

## 2.4. Identificadores

Aunque Chen examinó los identificadores no los incluyó en su diagrama ER. Autores posteriores generalmente los incorporan, usualmente subrayando el nombre o nombres de los atributos si éstos están incluidos en el diagrama.

Batini et al.[4] toman una aproximación ligeramente diferente en el sentido que tanto atributos como otros tipos-entidad pueden ser identificadores. Ellos definen un identificador para un tipo-entidad E como un conjunto:

$$I = \{A_1, \dots, A_n; E_1, \dots, E_m\}$$

Donde  $n \geq 0$ ,  $m \geq 0$ ,  $n+m \geq 1$

Donde  $A_1, \dots, A_n$  son atributos y  $E_1, \dots, E_m$  son otros tipos-entidad como E. Un identificador es clasificado como simple si  $n+m = 1$  o como compuesto si  $n+m > 1$ . Un identificador es interno si  $m=0$ , o externo si  $n=0$ . Un identificador es mixto si  $n>0$  y  $m>0$ . Ellos observaron que los tipos-entidad que tienen identificadores internos algunas veces son llamados tipos-entidad “fuertes” y los que sólo tienen identificadores externos son llamados tipo-entidad “débiles”. En su diagrama, usan un pequeño círculo abierto para la mayoría de los atributos. Si un atributo es simple e interno, el círculo es ennegrecido. Un identificador interno compuesto es mostrado conectando un círculo ennegrecido adicional a las líneas de los atributos que forman el identificador. Los identificadores mixtos también tienen líneas extendidas a las líneas de conexiones a los tipos-entidad externas. Debido a esto no se necesita el rectángulo de doble línea de un tipo-entidad débil usado por Chen. Teorey et al.[5] ven los atributos como de dos tipos, “identificadores” y “descriptores”. Los identificadores son únicos entre las ocurrencias de un tipo-entidad, mientras que los descriptores describen una ocurrencia de entidad.

## 2.5. Diagramas para los modelos básicos

Todos los autores analizados en este artículo usan un rectángulo para representar un tipo-entidad y un diamante para representar los tipos-relación en sus diagramas ER. Algunos, pero no todos, incluyen atributos en el diagrama, y los que lo hacen usan un óvalo con el nombre dentro (los identificadores subrayados), un círculo pequeño con el nombre al lado (los identificadores tienen un círculo ennegrecido) o una línea con el nombre al lado (el identificador no se indica). Para la cardinalidad se usa la notación de 1:1, 1:N, M:N de Chen o la notación (min-card, max-card) discutida en la sección 2.2. Elmasri y Navathe [9] usan una línea doble de conexión entre un tipo-entidad y un tipo-relación para mostrar una participación obligatoria. Teorey et al. [5] ennegrecen la mitad del diamante del tipo-relación que mira hacia un tipo-entidad cuya conectividad es “muchos” (si es “uno” no se ennegrece), y ponen un círculo pequeño en la línea cerca del diamante si la relación es opcional (si es obligatoria se usa una línea sin círculo).

Wertz [10] discute otros estilos de diagramación. Uno de estos, atribuido a Clive Finkelstein, no usa el diamante y pone un símbolo de pata de gallo en la línea de conexión en su extremo de “muchos” y también usa una barra vertical (a través de la línea de

conexión) para indicar una participación obligatoria y un círculo abierto para una participación opcional. Otro enfoque, atribuido a Charles Bachman, tampoco usa el diamante y usa una flecha en el extremo “muchos” del tipo-relación. Para indicar una participación opcional usa un pequeño círculo abierto en la unión del rectángulo del tipo-entidad y la línea de conexión; y un círculo negro para una participación obligatoria. En cuanto el modelo de Chen, Wertz[10] también dice, “variantes menores incluyen el uso de una flecha para indicar el lado “uno” de la relación y una flecha doble para indicar el lado “muchos”; un punto para indicar el lado “muchos” de una relación y una caja doble para indicar una entidad débil”.

## 2.6. Resumen

Varias revisiones sintácticas discutidas en esta sección ofrecen mayores conveniencias para los diseñadores de bases de datos. Las expansiones discutidas en esta sección se refieren tanto a las entidades como a las relaciones. Aunque estas revisiones o expansiones no enriquecen significativamente al poder de modelamiento del MER, algunas de ellas nos proveen nuevas formas de interpretar el modelo original. Por ejemplo, el ver el tipo-entidad como una clase provee una interpretación de este modelo, y así se enriquece la semántica.

## 3. Modelos entidad relación extendidos

Varios modelos ER extendidos (EER) han aparecido en la literatura reciente. En general, su contribución es añadir la abstracción de generalización al modelo original de Chen y variaciones parecidas a las ya discutidas en la sección 2.

### 3.1. El Modelo EER de Teorey, Yang y Fry

Teorey et al.[5] dicen que “la introducción de la abstracción de categorías en el MER resulta en dos tipos adicionales de objetos: jerarquías de subconjuntos y jerarquías de generalización”. Ellos dicen que las jerarquías de subconjuntos especifican subconjuntos con posibles intersecciones y la jerarquías de generalización especifican subconjuntos estrictamente disjuntos. Un tipo-entidad  $E_1$  es un subconjunto de otro tipos-entidad  $E_2$  si cada ocurrencia de  $E_1$  es también ocurrencia de  $E_2$ . En una jerarquía de generalización, cuando un tipo-entidad  $E$  es una generalización de tipos-entidad  $E_1, E_2, \dots, E_n$ , cada ocurrencia del tipo-entidad  $E$  también es una ocurrencia de uno y sólo uno de los tipos-entidad  $E_1, E_2, \dots, E_n$ . Por ejemplo, en el diagrama EER de la fig.3., FACULTATIVO y ESTUDIANTE son subconjuntos de PERSONA (asumiendo que una persona puede ser un miembro de la facultad y al mismo tiempo un estudiante), mientras que CURSO es una generalización de CURSO-CON-CRÉDITOS y CURSO-SIN-CRÉDITOS. Una jerarquía de generalización es llamada jerarquía “es-un” exclusiva. Cada tipo-entidad subconjunto es mostrado en el diagrama EER con una flecha gruesa de línea doble apuntada al tipo-entidad del cual es subconjunto. La jerarquía de generalización es mostrada con una flecha gruesa de línea doble desde cada subconjunto a un hexágono que contiene un nombre que caracteriza a todos los subconjuntos (TIPO-CURSO en el ejemplo anterior), y otra flecha gruesa desde el hexágono al tipo-entidad genérico. En los pasos para usar su modelo, Teorey et al. instruyen al usuario a poner el identificador y descriptores genéricos en el

tipo-entidad genérico y poner el identificador y descriptores específicos en los tipos-entidad subconjuntos. Así, en nuestro diagrama EER de ejemplo, RUT y NOMBRE son mostrados como atributos de PERSONA mientras que FACULTATIVO y ESTUDIANTE tienen sus propios atributos junto al identificador RUT.

### 3.2. El modelo EER de Markowitz y Shoshani

Markowitz y Shoshani [6] ofrecen una extensión del MER similar a la de Teorey et al.[5] pero que tiene más características y definiciones. Ellos describen la generalización como “un mecanismo de abstracción que permite ver un conjunto de conjuntos de entidades (e.g. SECRETARIA, FACULTATIVO) como un único conjunto de entidad genérico (e.g. EMPLEADO)”. Note que los autores usan el término “conjunto-entidad” y “conjunto-relación” para denotar tipos-entidad y tipos-relación respectivamente. Por motivos de claridad usaremos los términos “tipo-entidad” y “tipo-relación”. Markowitz y Shoshani dicen que la generalización define una relación transitiva, por eso si el tipo-entidad  $E_1$  es una generalización directa del tipo-entidad  $E_2$ , y  $E_2$  es una generalización directa del tipo-entidad  $E_3$ , entonces  $E_1$  es una generalización transitiva de  $E_3$ . La especialización es el inverso de la generalización. Un tipo-entidad especializado hereda los atributos de todos los tipos-entidad genéricos directos y transitivos, incluyendo el identificador del tipo-entidad. Estos atributos son llamados atributos heredados. Los tipos de entidad que no son débiles, y no son una especialización de otro tipo-entidad, son llamados tipo-entidad “independientes”. En el diagrama EER de ejemplo (fig.4.) PERSONA es una generalización de FACULTATIVO y ESTUDIANTE, y ESTUDIANTE es una generalización de ESTUDIANTE-GRADUADO. De este modo PERSONA es una generalización transitiva de ESTUDIANTE-GRADUADO, que de este modo hereda todos los atributos desde ESTUDIANTE y PERSONA, y tiene además su atributo propio GRADO. Markowitz y Shoshani también permiten agregación “total”, por eso los tipos-relación pueden asociar tipos-entidad y tipos-relación y no sólo tipos-entidad. Por ejemplo, en el diagrama EER de la fig.4., la agregación total permite la asociación entre el tipo-relación ENSEÑA y el tipo-relación OFRECE. Markowitz y Shoshani definen sus diagramas como grafos dirigidos. Las flechas muestran dirección y representan la interacción de elementos y dependencias existentes. El grafo debe ser acíclico pues los ciclos implican tipos-entidad redundantes.

### 3.3. El modelo ECER

El modelo entidad-relación conceptual extendido (ECER) desarrollado por Czejdo et al.[8] usa interfaces gráficas para formular consultas y actualizar. Este concepto retiene las ventajas del diagrama ER normalmente usados sólo para diseño de bases de datos y las aplica a las interfaces de usuario simples del tipo “apuntar y clicar”. Son incluidos varios operadores para actualizar, recuperar, calcular, etc.

Czejdo et al. definen tres tipos de generalización/especificación. El tipo 1 involucra exactamente dos tipos-entidad, donde uno es un subconjunto del otro. Note que en el texto original los autores usan el término “conjunto entidad” para denotar a un tipo-entidad. Por motivos de claridad, nosotros nos quedaremos con “tipo-entidad”. Un tipo 1 de



generalización/especialización es mostrado en el diagrama con símbolo de subconjunto sobre la línea de conexión de los dos tipos-entidad. En el diagrama ECER de ejemplo de la fig.5., ESTUDIANTE-GRADUADO es de ese tipo de especialización de ESTUDIANTE. El tipo 2 involucra un tipo-entidad generalizado A y varios tipos-entidad especializados  $B_1, B_2, \dots, B_n$ , donde  $A=B_1 \cup B_2, \dots \cup B_n$ . Esto es representado en el diagrama conectando cada tipo-entidad a un pequeño círculo con una U dentro. Por ejemplo, en la fig. 5, PERSONA es la unión de FACULTATIVO y ESTUDIANTE, así cada persona es un miembro de la facultad, un estudiante o ambos. El tipo 3 es como el tipo 2 excepto que los tipos especializados son disjuntos y están conectados en el diagrama a un círculo que contiene el símbolo más. En el ejemplo CURSO-CON-CRÉDITO y CURSO-SIN-CRÉDITO son este tipo de especialización de CURSO.

### 3.4. El modelo EER de Hohenstein y Gogolla

El modelo de Hohenstein y Gogolla [11] es definido en conexión con una proposición de cálculo para el modelo. Su modelo está basado en el de Chen y es similar al de Teorey et al.[5] en el sentido que incluye la generalización. Hohenstein y Gogolla presentan la generalización y especialización como “construcciones de tipos” con tipos de entrada y tipos de salida. Los tipos de entrada “ya definido o básicos” son usados en la construcción de tipos de salida, esto significa que “todas las entidades de los tipos de entrada son agrupadas y distribuidas entre los tipos de salida”. Esto es mostrado en el diagrama conectando todos los tipos de entrada en la base de un triángulo, y conectando la cima del triángulo con los tipos de salida. Dentro del triángulo se coloca un símbolo de subconjunto o un símbolo de igualdad. Por ejemplo, en el diagrama EER de la fig.6., FACULTATIVO y ESTUDIANTE son tipos de entrada usados en la construcción del tipo de salida PERSONA. El modelo también permite a los atributos tener valores de tipos de datos complejos y enumerados. Los usuarios pueden definir operaciones para manipular los ítems de datos complejos. En el diagrama, un tipo de dato de atributo es indicado junto con el nombre del atributo. Los atributos pueden ser multivaluados y el conjunto de valores para un atributo multivaluado puede ser clasificado dentro de tres categorías diferentes: “conjuntos”, “bolsas” y “listas”. Los elementos en un conjunto ocurren una sola vez, en una bolsa los elementos pueden ocurrir más de una vez, mientras que los elementos en una lista son guardados en forma enumerada. Además de la generalización y especialización, este modelo también incluye el concepto de tipo-entidad de “estructura compleja”. Una entidad de un tipo-entidad de estructura compleja está compuesto por otras entidades, llamadas “componentes”. En nuestro diagrama de ejemplo, DEPARTAMENTO es un tipo-entidad de estructura compleja porque incluye dos componentes: PRESIDENTE y COMITÉ-DE-GRADO (una lista). Estos dos componentes son de otro tipo-entidad: FACULTATIVO.

### 3.5. El modelo ECR

Navathe et al.[7] proponen un modelo de entidad-categoría-relación (ECR) el cual usan para una integración de vistas. Una categoría es “un subconjunto de entidades de un tipo-entidad”. Un conjunto de entidades es llamada clase de objetos, sea éste tipo-entidad o una categoría. Las categorías usualmente comparten la mayoría de los atributos, pero no necesariamente todos. Por ejemplo, FACULTATIVO y ESTUDIANTE son categorías del

tipo-entidad PERSONA. Ambos comparten los atributos RUT y NOMBRE, pero tienen sus atributos propios RANGO y GPA respectivamente. El modelo ECR usa notaciones en el diagrama similares a las de Czejdo et al.[8] (ver fig.5.). Las categorías son mostradas en el diagrama como hexágonos, conectados al tipo-entidad con un símbolo de subconjunto dibujado sobre la línea. Los atributos compartidos son conectados al tipo-entidad y los atributos de categoría no compartidos son conectados al hexágono. Las categorías son el foco de la integración de vistas. Las categorías son definidas para cada fuente de datos para la integración y son comparadas durante el proceso.

### *3.6. Resumen*

La discusión sobre mejoramiento en el poder para el modelamiento ha acompañado al enfoque ER por dos décadas. Smith y Smith [12] presentan los conceptos de generalización y agregación. El modelo de datos semántico de Hammer y McLeod[13] introdujo los conceptos de clase y subclase, así como otros conceptos avanzados de modelamiento.

Varias extensiones del MER original, como los que hemos resumido en las secciones previas, trabajan la semántica en varios grados. Detrás de las diferencias sintácticas de varias extensiones está el enriquecimiento semántico acerca de las relaciones entre las entidades. Por ejemplo, muchos de los cambios sintácticos propuestos giran alrededor de la generalización/especialización, una clara indicación de las mejoras semánticas. Los modelos resumidos en esta sección son independientes, pero también pueden ser vistos como pasos hacia los modelos de datos orientados a objeto o modelos ER orientados a objetos, como analizaremos en la próxima sección.

## **4. Modelos ER orientados a objetos**

Con los continuos adelantos en las bases de datos orientadas a objeto, ha aumentado el interés por incluir características del paradigma orientado a objeto para ajustar el modelo ER al diseño de bases de datos orientadas a objeto. Estas características incluyen generalización y herencia, el principio ocultamiento de la información, tipos de datos abstractos y traspaso de mensajes. Los siguientes modelos incorporan conceptos de orientación a objeto en el modelo ER.

### *4.1. El modelo OOER.*

Navathe y Pillalamarri[14] usan un enfoque orientado a objeto para su modelo de datos. Ya que los modelos semánticos logran sus metas usando abstracción, la que definen como “una primitiva semánticamente irreductible en el modelo que permite modelar un tipo fundamental de relación en una forma que esconde los detalles y las diferencias, y se concentra en las propiedades comunes de un conjunto de objeto”, presentan su modelo en términos de 5 abstracciones comúnmente usadas. Estas son agregación (ES-PARTE-DE), generalización (ES-UN), clasificación (ES-UNA-CLASE-DE), asociación (ESTA-ASOCIADO-CON), e identificación (ES-IDENTIFICADO-POR).

Navathe y Pillalamarri examinaron el modelo ER básico a causa de su extendida popularidad, pero lo encontraron deficiente por su fracaso para modelar explícitamente generalización y clasificación, y por su capacidad limitada para describir la interacción entre constructores. Esto llevó al modelo de entidad-relación orientado a objeto (OOER) que retiene muchos rasgos del modelo de Chen, pero extiende sus capacidades. La abstracción de agregación involucra la entidad (también conocida como un objeto), que Navathe y Pillalamarri definen como algo que existe en el mundo real y tiene atributos que lo describen. Los valores de atributo de un tipo similar se agrupan en clases de atributos o dominios. Las entidades que tienen atributos similares se clasifican en clases de entidad o en tipos de entidad, y una entidad puede definirse como una agregación de atributos o una agregación de otras entidades (lo que distingue el modelo OOER del modelo ER). Se pueden definir muchas operaciones para las clase de entidad: actualización, relacionales (select, join y project), comparación, y aritméticas.

Con la abstracción de generalización dos o más clases de objetos pueden ser generalizados para formar una clase de objetos más elevada. Especialización es la noción inversa donde nuevas clases de objeto pueden ser definidas como subclases de una o más clases de objeto. Por ejemplo en el diagrama de OOER de la Fig. 7 EMPLEADO es una generalización de GERENTE, INGENIERO y SECRETARIA, mientras que CLIENTE-EXTRANJERO es una especialización de CLIENTE. El constructor usado es la clase de objeto de tipo superclase o subclase (que son términos relativos ya que la misma clase puede formarse por generalización o especialización). En el caso de la herencia múltiple, el usuario debe especificar la precedencia. Además la relación de generalización de clases de objeto se pueden definir con restricciones de Exclusión, Intersección o Igualdad de conjuntos. Respectivamente, éstas indican: no hay instancias comunes, algunas instancias comunes, o una relación 1: 1 entre los objetos que participan en una generalización. Todas las operaciones que se han definido para las clases de entidad son también aplicables a las subclases.

La abstracción de clasificación se basa en la noción de conjunto. Este concepto fuerza la distinción entre instancias individuales y una clase de estas instancias. Si ciertas propiedades se aplican a todo un conjunto, entonces se pueden modelar como atributos de la clase. Una agrupación de instancias puede formar una clase de entidad, una clase de relación, o una clase de superclase/subclase. Adicionalmente, una metaclass es una agrupación en que todos los objetos son clases, en otras palabras es un conjunto de conjuntos de instancias. Como se indica en la Fig. 7, las instancias de la metaclass PERSONA son las clases de entidad CLIENTE y la clase de objeto generalizada EMPLEADO. Se pueden definir operaciones agregadas para los atributos de la clase, como contar, sumar, etc.

La abstracción de asociación en el modelo OOER es la relación que representa la interacción entre clases de objeto como una clase de objeto más elevada. Exhibe una dependencia de existencia con los objetos que participan en la asociación.

Identificación es la abstracción usada para identificar unívocamente las estructura de objeto que se han formado. Cada clase de entidad tiene por lo menos un atributo que se

declara como su identificador (clave primaria) y se usa para acceder a un objeto. Las instancias de una clase de objeto generalizada pueden ser identificadas por las instancias correspondientes de la clase que la definen o por cualquier atributo importante de la propia clase generalizada.

Navathe y Pillalamarri[14] evalúan su modelo OOER en cuanto a su nivel de orientación a objeto. Concluyen que tiene una orientación a objetos estructural y operacional pero no conductual. La orientación a objetos estructural se refiere a que permite la definición de estructuras de datos para representar objetos complejos. La orientación a objetos operacional se refiere a que permite la definición de operadores genéricos para manejar objetos complejos como un todo, en lugar de descomponerlos en operaciones primitivas de objetos más simples. El modelo OOER se define de forma tal que la estructura y operaciones son encapsuladas en cada clase de objeto. Sin embargo, no existen los detalles sintácticos para especificar tipos abstractos de datos, encapsulamiento de los datos, paso de mensajes y sobrecarga de operadores, ni tampoco existe la orientación a objetos conductual.

El modelo OOER usa notaciones de diagramas ER convencionales para denotar clases de entidad, atributos y clases de relación, con la incorporación de flechas desde las clases de entidad relacionadas hacia el diamante de la clase de relación. La generalización y especialización se representa en el diagrama con un polígono de seis lados. La diferencia se muestra con la dirección de la flecha que los une, que es hacia el polígono en la generalización y desde el polígono para la especialización. En el modelo OOER se muestra una metaclassa con un polígono de cinco lados. Los objetos que forman una clase están conectados por una flecha que apunta hacia el polígono, y los objetos que son atributos de la clase están conectados sin flechas.

#### *4.2 El modelo BIER.*

En su enfoque entidad-relación con comportamiento integrado (BIER) para el diseño de bases de datos orientadas a objetos Kappel y Schrefl [15] establecen un modelo estático y un modelo conductual. El modelo estático representa las propiedades estructurales de los objetos del mundo real y el modelo conductual explica eventos del mundo real en términos de conductas de objetos. Mientras el modelo conductual se basa en redes de Petri, el modelo estático usa un enfoque de diagrama ER extendido. En el modelo estático la estructuras del mundo real son representadas como tipos de objeto (similares a los tipos-entidad en el modelo ER). Las características de los objetos son representadas como atributos. El valor de un atributo puede ser de un tipo de datos o un tipo de objetos. Este concepto es utilizado para representar las relaciones entre los objetos. También se incluyen generalización y herencia.

El modelo define los conceptos de objetos primitivos y objetos complejos. Los objetos primitivos son independientes en el sentido que no se definen a partir de otros objetos, mientras que los objetos complejos se definen a partir de objetos de niveles más bajos. Además, se pueden clasificar los objetos complejos como objetos “grupales” y objetos “agregados”. Un objeto grupal se define a partir de objetos de niveles más bajos (llamados miembros) de una misma clase; esta abstracción se denomina agrupación. Un objeto

agregado se define a partir de objetos de niveles más bajos (llamados componentes) de clase distintas; esta abstracción se denomina agregación. La agregación y la agrupación emplean el concepto de relación. Los objetos de niveles más bajos se llaman “independientes” si no sólo participan en la relación que es representada por el objeto del nivel más alto en el que están agrupados. Deben existir antes que el objeto de nivel más alto sea creado, y continúan existiendo si el objeto de nivel mas alto es eliminado.

Los objetos dependientes son privados de un objeto de mayor nivel y son creados y anulados junto con dicho objeto. En nuestro diagrama de ejemplo (Fig. 8), DEPARTAMENTO es un tipo de objeto agregado con tipos componente independientes EMPLEADO y PROYECTO. Esta agregación puede ser usada para representar el tipo-relación TRABAJA-PARA entre DEPARTAMENTO y EMPLEADO, y el tipo-relación CONTROLA entre DEPARTAMENTO y PROYECTO. El atributo TRABAJA-EN del objeto EMPLEADO puede tener un valor del tipo objeto PROYECTO. EMPLEADO a su vez es un tipo de objeto grupal con tipo miembro dependiente NIÑOS.

### *4.3 El modelo OOER.*

Como el modelo BIER[15], el modelo entidad-relación orientado a objetos (OOERM) desarrollado por Gorman y Choobinech[16] también incluye orientación a objetos conductual. Gorman y Choobinech presentan la visión orientada a objeto como una extensión natural de la visión entidad-atributo-relación: "Cuando pensamos en un objeto o entidad (e.g. una bicicleta), consideramos su atributos, como el color. Naturalmente también consideramos lo que normalmente podemos hacer con ese objeto (montarla, estacionarla, etc.)." En términos de definición, describen un objeto como “un 'paquete' de información y una descripción de su manipulación.” Esto corresponde a un conjunto de variables relacionadas y a un conjunto de operaciones sobre esas variables en términos de programación procedural. Un mensaje se define como la especificación de una manipulación sobre un objeto, esta especificación consta del receptor (objeto a ser manipulado), el seleccionador (nombre simbólico de la manipulación), y argumentos opcionales (otros posibles objetos que toman parte en el proceso). Un mensaje recuerda a la llamada a un procedimiento. Un método es una descripción de un único tipo de manipulación.

En programación orientada a objeto, una clase o un tipo de objeto corresponde a un tipo-entidad en el modelo ER, y el conjunto de variables de objeto y métodos corresponden a atributos. La visibilidad de un objeto, que se refiere a qué objetos pueden acceder y qué objetos pueden ser accesados, es similar al concepto de relación en el modelo ER . El paso de mensajes usa la visibilidad del objeto, deja que el objeto que envía acceda a las variables del objeto receptor.

El esquema OOERM consta de varios diagramas, llamados diagrama entidad-relación orientados a objeto, y se usan para representar tanto las propiedades estructurales como conductuales de la base de datos. Un OOERD estructural se dibuja para definir la clasificación y jerarquía de objetos, incluyendo las relaciones entre objetos. Además, para modelar cada proceso de aplicación se dibuja un OOERD por separado. En un OOERD las

clases de entidad y relaciones usan los rectángulos y diamantes normales, mientras que los atributos son representados por nodos circulares con el nombre del atributo al lado (Fig. 9). Una jerarquía de generalización es representada por una flecha gruesa doble, que apunta a la clase de nivel más alto. Si la línea de la flecha no está ennegrecida representa una subclase que es un subconjunto de la clase del nivel más alto sin propiedades adicionales. Si la línea de la flecha está ennegrecida representa una subclase que es una especialización del nivel más alto y tiene propiedades adicionales. Como muestra la Fig. 9. AUTOR es un subconjunto de PERSONA mientras que ESTUDIANTE es una especialización de PERSONA.

Los métodos son mostrados en un OOERD como óvalos con el nombre del método dentro. Están unidos a su clase de entidad por una línea sólida, similar a un atributo. El paso del mensaje es mostrado por una flecha segmentada que va desde la clase emisor a la clase receptor. La posición del mensaje en una sucesión de mensajes es mostrada por un entero entre paréntesis. Los parámetros opcionales se muestran sobre la flecha y los resultados opcionales se muestran bajo la flecha.

Se usan flechas segmentadas que se separan para indicar la bifurcación condicional y se marca cada rama con "T" para verdadero o "F" por falso. A los métodos, paso del mensajes e indicadores de la bifurcación se les denomina iconos dinámicos. Todas las clases de entidad tienen cinco operaciones predefinidas como propiedades implícitas. Estas operaciones son "creación," "añadir a subclase", "remover de subclase", "borrar" y "seleccionar". Tienen la misma definición estándar para todas las clases. Los mensajes que invocan estas operaciones predefinidas pueden ser especificados interactivamente por el usuario o usados dentro de los métodos. Se usa una cláusula "where" (similar a SQL) para identificar las instancias de clase específicas para las que se aplican las operaciones. Las operaciones predefinidas se representan en el OOERD con un óvalo de método sobrepuesto sobre la clase de entidad a que se aplica. La letra que representa la operación predefinida se pone en la parte derecha del óvalo y la secuencia ordinal de la operación se pone entre paréntesis en la parte izquierda del óvalo. Otro rasgo de este modelo es la derivación de atributo. En este caso el valor de un atributo de un objeto puede ser derivado como el resultado de un mensaje recibido por ese objeto.

Un proceso puede tener una secuencia de muchas operaciones. Los modelos OOERDs están pensados para modelar procesos individuales de aplicación. El diagrama provee una vista pictórica de los cambios de estado que resultan de la ejecución del proceso. Por eso, los autores recomiendan que se dibuje un OOERD separado por cada proceso de aplicación y que cada OOERD debe contener sólo los detalles estructurales y operacionales pertinentes a ese proceso. Por ejemplo, el OOERD por el proceso de préstamo de libro (fig. 10) muestra sólo las clases pertinentes y sus operaciones.

#### *4.4. Comentarios.*

En los 80s hemos visto un cambio en el énfasis del modelamiento de datos: los intereses generales relacionados al modelamiento conceptual (inspirado en parte por la inteligencia artificial)[17] han convergido a modelos de datos orientados a objeto. Según Hull y

King[2], esencialmente, los modelos semánticos encapsulan aspectos estructurales de los objetos, mientras que los lenguajes orientados a objeto encapsulan aspectos conductuales de éstos.

Los modelos de datos orientados a objeto han incorporado algunas consideraciones importantes trabajados por modelos conceptuales (tal como la agregación) al igual que aspectos trabajados por modelos lógicos (tal como el no uso de la 1ª forma normal en algunos modelos relacionales extendidos y navegación en el modelo de red). Como consecuencia, el límite entre los modelos lógico y conceptual es borroso. Un modelo de datos orientados a objeto puede ser visto como un modelo tanto a nivel conceptual como a nivel lógico.

Como señaló Loomis[18], un modelo de objeto incluye relaciones entre objetos. Estas relaciones, que son asociaciones entre objetos, son como las relaciones entre entidades en el modelo de datos ER. Un modelo de objeto también puede incluir relaciones de contención y estructuras recursivas. Un DBMS de objetos construye estas relaciones en la base de datos de objetos y puede usarlos directamente en tiempo de ejecución cuando retorna los objetos a las aplicaciones. Las relaciones en los modelos de datos ER son completamente abstractas, mientras que los modelos de datos orientados a objeto llevan la semántica de las relaciones a la implementación.

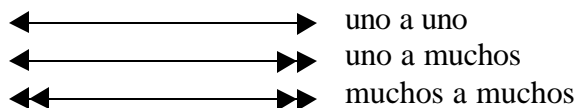
Otro aspecto que se debe señalar es la parte conductual de una base de datos orientado a objetos pura. Chen[ 19] argumenta que el modelo ER tiene una parte de conductual, pero sin dar mayor explicación. Asegura que "...en el mundo ER los 'datos' y los 'procesos' están en igualdad, mientras que en el mundo de la orientación a objeto los procesos están encapsulados con los datos. Si pensamos profundamente, la encapsulación de los procesos con los datos es un asunto de 'empaquetamiento' y nada tiene que ver con el argumento de si la orientación a objetos tiene mayor comportamiento dinámico que el modelo ER (u otros paradigmas). La forma de empaquetamiento de procesos de la orientación a objeto trabajará bien en ciertas aplicaciones y en otras no." Sin embargo, Chen no provee una elaboración mayor de sus argumentos, algunos de los modelos resumidos en esta sección han indicado cómo la orientación a objeto y el MER pueden acercarse uno o a otro.

## **5. Esquemas de Diagramas orientados a Objetos.**

Finalmente, echaremos un vistazo a algunos asuntos relacionados que van más allá del modelamiento ER. En particular, daremos un breve resumen de los diagramas de esquemas orientados a objeto usados en ODMG 93[20].

Se han usado representaciones gráficas en modelos de datos orientados a objetos para representar esquemas de base de datos. Nos referiremos a ellos como *diagramas de esquemas orientados a objeto*. Aquí examinamos la representación gráfica usada en el estándar de base de datos de objeto ODMG 93 (*Object Database Standard*), que acorta la brecha entre el modelamiento ER y el modelamiento orientado a objeto. ODMG 93 ha adoptado notaciones básicas de los diagramas ER con extensiones importantes. La representación gráfica usa las notaciones siguientes:

1. los tipos de objeto se muestran como rectángulos;
2. los tipo-relación se muestran como líneas etiquetadas;
3. la cardinalidad permitida para el tipo-relación se indica por las flechas en el fin de las líneas:



4. flechas grandes grises apuntan a un supertipo desde un subtipo.

En esta representación gráfica, los tipos de objeto son representados exactamente de la misma manera como el tipo-entidad en los modelos ER. Esta es una indicación clara de que los tipos de objeto se extendidos a partir de los tipos-entidad. Los tipos-relación han hecho cambios significativos. La representación tradicional del diamante en un diagrama ER se reemplaza por líneas etiquetadas. Las etiquetas son los nombres de los tipos-relación; sin embargo, las relaciones en ODMG-93 tienen direcciones. Una relación en diagramas ER es ahora reemplazada por un *par* de relaciones (llamadas *relaciones inversas*) con direcciones opuestas. Por ejemplo, la conexión entre dos tipos del objeto curso y estudiante puede ser representada por relación toma de estudiante a curso y su relación inversa es-tomado-por de curso a estudiante. Una definición ODL (Objet Definition Language) para la interfaz del curso debería incluir lo siguiente:

```

interface Curso{
  . . .
  relationship Set <Estudiante> es_tomado_por
                inverse Estudiante: toma;
  . . .
}

```

La Fig.11 es un ejemplo de un diagrama de esquema orientado a objeto (los atributos no están incluidos).

La representación mejorada de las relaciones en los diagramas ODMG-93; junto con la inclusión explícita de cardinalidad como mencionamos antes (tal como uno-a-muchos o muchos-a-muchos) ha traído un enriquecimiento significativo de la semántica. La falta de direcciones en el tipo-relación en el diagrama ER original ha sido siempre una fuente potencial de confusión semántica; el problema ahora está resuelto. Más importante, la inclusión explícita de las direcciones puede facilitar la construcción de caminos de navegación para contestar preguntas. La construcción de caminos para la navegación es crucial en las bases de datos orientadas a objeto.

Un ejemplo de pregunta en una sintaxis parecida a SQL es:

```

select enseñanza (estudiante: x.nombre, profesor: z.nombre)
from x in Estudiantes, y in x.toma, z in y.dictado_por

```



```
where z.rango = "professor_asociado"
```

Note que las direcciones de relación son importantes para especificar los caminos en la pregunta: la relación `toma` sigue la dirección desde `estudiante` a `profesor`, mientras que la relación `dictado_por` sigue la dirección de `profesor` a `estudiante`.

## 6. Discusión

En este artículo comenzamos viendo ER como el primer modelo de datos centrado en las relaciones y examinamos varias extensiones. Aunque extensiones simples al modelo original han agregado mejoras menores, los modelos ER extendidos han mejorado significativamente el poder de modelamiento, estableciendo un puente hacia el modelo de datos orientados a objetos. Recalamos que detrás de la extensión sintáctica existe un enriquecimiento semántico.

El modelo entidad-relación es una herramienta muy poderosa y útil para muchos propósitos de modelamiento. Es muy popular y todos los autores hablan favorablemente de sus capacidades. Sus principios son relativamente simples, pero efectivos, lo que parece explicar su amplia aceptación. El diagrama literalmente dibuja un cuadro del mundo real bajo modelamiento, y este cuadro puede ser entendido incluso por el usuario final menos sofisticado.

Las mejoras y variaciones discutidas la sección 2 agregan relativamente poco a la potencialidad del modelo. Muchas de ellas son diferencias que reflejan preferencias personales, tal como las maneras de dibujar los diagramas. De hecho, algunos de ellas pueden dificultar el uso del modelo básico porque podrían causar confusión. Ayudaría tener una norma para los diagramas ER de modo que un solo conjunto de símbolos fuera usado por todos los diseñadores de base de datos. Esto sería particularmente útil para usuarios finales que tienen que interactuar con muchos diseñadores pero que, lógicamente, no quieren tener que aprender diversas notaciones.

Algunas de las diferencias mostradas en la Sección 2 se relacionan con la terminología o la presentación. El modelo ER original[1] era muy sencillo y práctico. Autores posteriores intentaron dar disertaciones más eruditas y gastaron más tiempo en definiciones y terminología. Autores más recientes han tendido a usar términos como "clase" y "objeto" que son más consistentes con los conceptos del modelo orientado a objeto. Un concepto de la sección 2 que tiene una diferencia notable es la cardinalidad. El modelo de Chen se basó en la notación 1: 1, 1: N, M: N, pero la notación (min.card, max.card) agrega la capacidad de poner límites inferiores y superiores en la participación de una entidad en una relación. Esto se puede usar tanto para especificar valores de cardinalidad como para indicar restricciones en la participación. Por contraste, el concepto de atributo compuesto se puede perder al momento en que es mapeado a una relación. Sin embargo, aún es útil para entender la relación entre atributos durante el modelamiento conceptual y el análisis.

La contribución mayor de los modelos extendidos en la sección 3 es la generalización. A pesar que hay algunas diferencias en definiciones y diagramas, el punto es que se reconoce una jerarquía de clases y se usa el concepto de la herencia ya sea para compartir o separar

atributos específicos. En la sección 4, examinamos el modelo OOER de Navathe y Pillalamarri[14] que aporta más a la discusión de la terminología, pero es similar a los discutidos en la sección 3. Una excepción notable es el concepto de clase representado por un polígono de cinco lados, particularmente cuando se usa como una metaclass. Estos autores presentan una disertación notable en términos de orientación a objeto.

El Modelo BIER de Kappel y Schrefl[15] y el modelo OOER de Gorman y Choobinch[16] representan un adelanto considerable, ya que ambos pueden ser vistos como verdaderos modelos de datos orientados a objetos. La incorporación de métodos, mensajes y operaciones hacen de ellos una mejor opción para el diseño de base de datos orientadas a objeto. A medida que las bases de datos orientadas a objeto se hagan más populares estos modelos y sus inevitables clones hallarán indudablemente un público más amplio. Finalmente los diagramas de esquema orientados a objeto que se usan en ODMG 93 revelan el predominio de notaciones ER y la relación conceptual entre estos dos enfoques de modelamiento de datos (a saber, entidad-relación y orientación a objeto).

## **7. Conclusiones**

En las dos décadas pasadas hemos sido testigos de una herramienta muy poderosa, el modelo entidad-relación creció hasta ser una técnica ampliamente reconocida. Las mejoras a este modelo, particularmente la generalización y un enfoque orientado a objeto, han añadido capacidad y poder al modelo original. Muchos artículos escritos en las dos décadas pasadas han ayudado al discutir y explicar los conceptos, y también han ofreciendo ejemplos del uso del modelo. Muchas de las extensiones han agregado poco al poder expresivo del modelo ER, pero las más recientes, sobre todo las relacionadas a conceptos de objetos han añadido algunas capacidades.

## **Agradecimientos**

Al autor le gustaría dar reconocimientos a los esfuerzos y contribuciones de Munib Siddiqi durante las partes iniciales de este proyecto. Los comentarios de Z. Chen han sido muy útiles.

## **Referencias**

- [1] P. Chen, The entity-relationship model: Towards a unified view of data, ACM Trans. Sistemas de bases de datos 1 (1) (1976) 9-36
- [2] R. Hull, R. King, Semantic database modelling: Survey, applications, and research issues, ACM Computing Surveys 19 (3) (1987) 201-260
- [3] T. Jones, I. Song, Analysis of binary/ternary cardinality combinations in entity-relationship modeling, Data and Knowledge Engineering 19 (1) (1996) 39-64.
- [4] C. Batini, S. Ceri, S. Navathe, Conceptual Database Design, Benjamin/Cummings, Redwood City, CA, 1992.
- [5] T. Teorey, D. Yang, y J. Fry. A logical design methodology for relational databases using the extended E-R model, ACM Computing Surveys, 18(2):197-222, 1986.

- [6] V. Markowitz, A. Shoshani, Representing extended entity-relationship structures in relational databases: A modular approach, *ACM Trans. on database System* 17 (3) (1992) 423-464
- [7] S. Navathe, R. Elmasri, J. Larson, Integrating user views in database design, *IEEE Computer* 19 (1) (1986) 50-62
- [8] B. Czejdo, R. Elmasri, M. Rusinkiewicz, D. Embley, A graphical data manipulation language for an extended entity-relationship model, *IEEE Computer* 23 (3) (1990) 26-35
- [9] R. Elmasri, S. Navathe, *Fundamentals of Database Systems*, Benjamin/Cummings, Redwood city, CA, 1994.
- [10] C. Wertz, *Relational Database Design*, CRC Press, Greenwich, CT, 1993.
- [11] U. Hohenstein, M. Gogolla, A calculus for an extended entity-relationship model incorporating arbitrary data operations and aggregate functions, in: C. Batini (Ed.), *Entity-Relationship Approach: A Bridge to the User*, Proc. 7<sup>a</sup> Int. Conf. On entity relationship approach, 1988, Rome, Italy, North Holland, 1988.
- [12] J. Smith, D. Smith, Database abstractions: Aggregation and generalization, *ACM Trans. Database Systems* 2 (2) (1997) 105-33.
- [13] M. Hammer, D. McLeod, Database description with SDM: A semantic data model, *ACM Trans. Database Systems* 6 (3) (1980) 351-386
- [14] S. Navathe, M. Pallalamari, Toward making the ER approach object-oriented, in: C. Batini (Ed Entity-Relationship Approach: A Bridge to the User, Proc. 7<sup>a</sup> Int. Conf. On entity relationship approach, 1988, Rome, Italy, North Holland, 1988.
- [15] G. Kappel, M. Schref, A behavior-integrated entity-relationship approach for the design of object-oriented databases, in: C. Batini (Ed Entity-Relationship Approach: A Bridge to the User, Proc. 7<sup>a</sup> Int. Conf. On entity relationship approach, 1988, Rome, Italy, North Holland, 1988.
- [16] K. Gorman, J. Choobineh, The object-oriented entity-relationship (OOERM), *J. Management Inf. Sys* / (3) (1991) 41-65
- [17] M. Brodie, J. Mylopoulos, J. Schmith, *On conceptual Modelling: Perspectives from Artificial Intelligence*, springer-Verlag, New York, NY, 1984.
- [18] M. Loomis, *Object Databases: The Essentials*, Addison-Wesley, Reading, MA, 1995.
- [19] P. Chen, Entity-relationships vs. Object-orientation, en: G. Pernul, A.M. Tjoa (Eds.), Proc. 11<sup>o</sup> Int. Conf. On entity relationship approach, octubre 1992, pp.1-2.
- [20] R. Cattell (Ed.), *Object-Database Standard: ODMG-93*, Morgan Kaufmann, San Mateo, CA, 1994.

FIGURAS

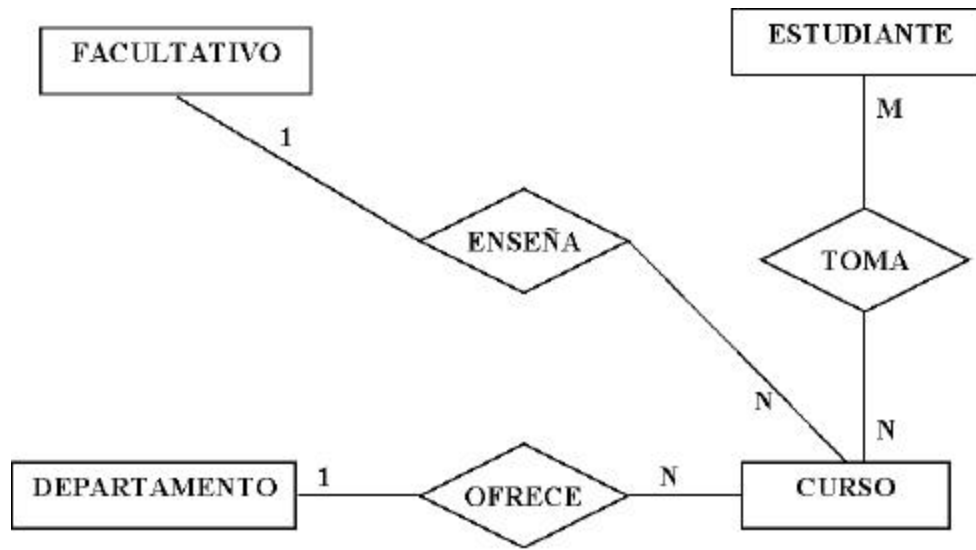


Fig. 1. Un diagrama ER para una base de datos universitaria

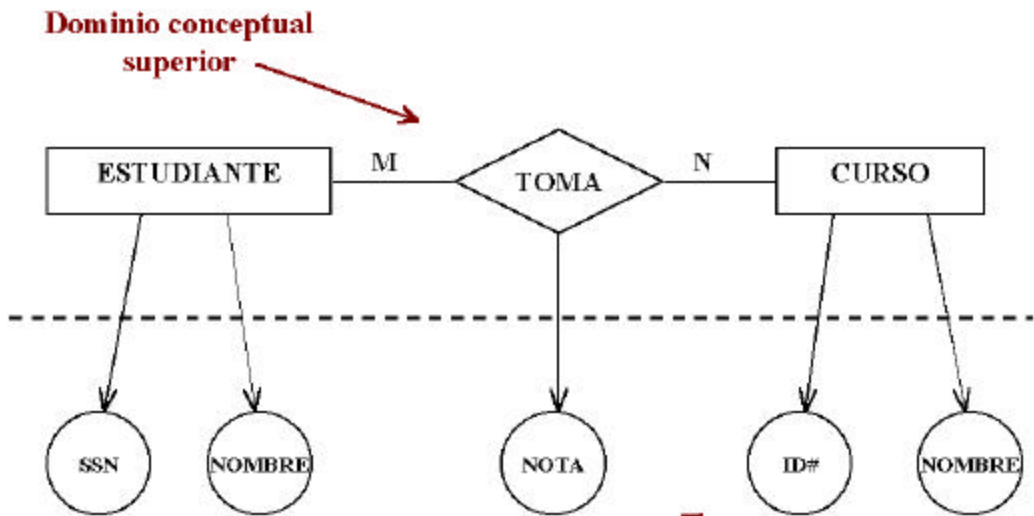
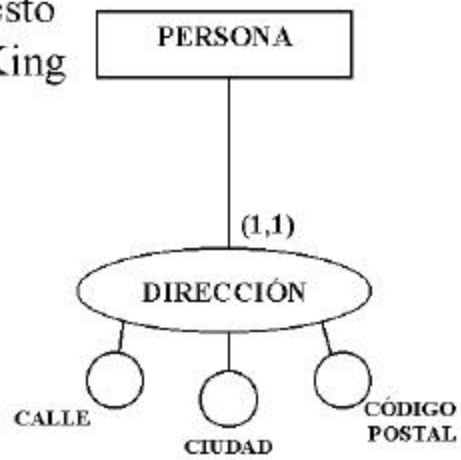


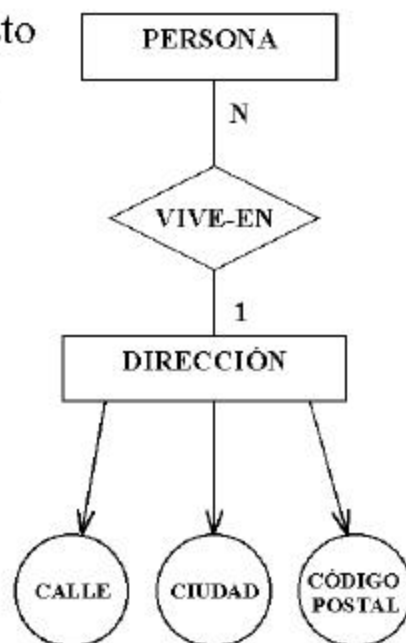
Fig. 2. Tipos de atributos y valores para estudiante, toma y curso

**Dominio conceptual inferior**

Atributo compuesto  
notación Hull y King



Atributo compuesto  
notación Chen



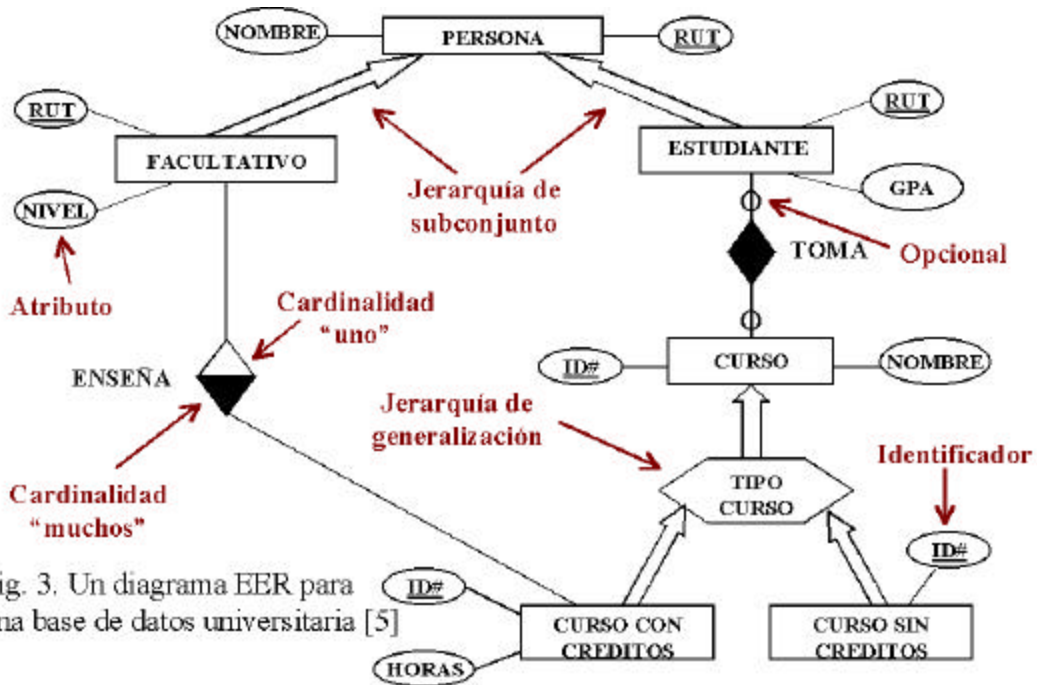


Fig. 3. Un diagrama EER para una base de datos universitaria [5]

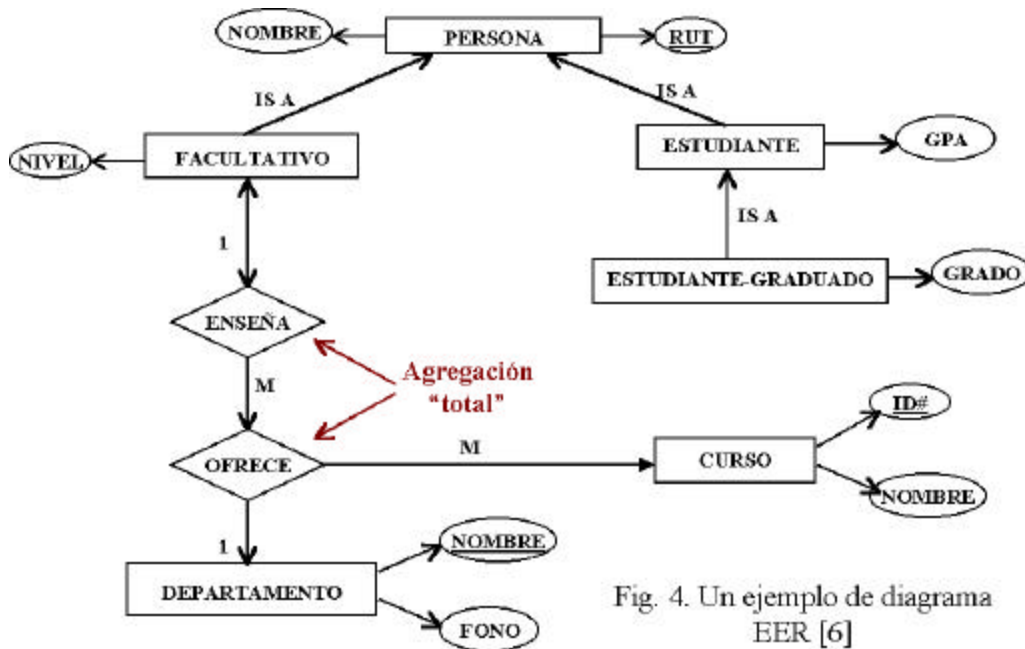


Fig. 4. Un ejemplo de diagrama EER [6]

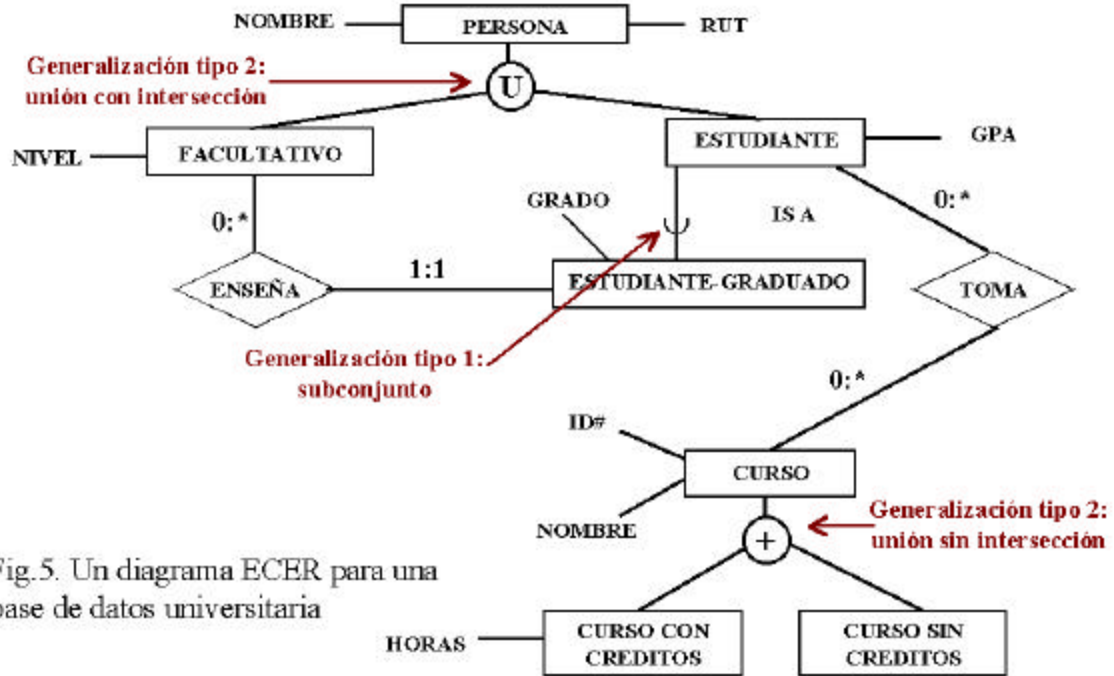


Fig. 5. Un diagrama ECER para una base de datos universitaria

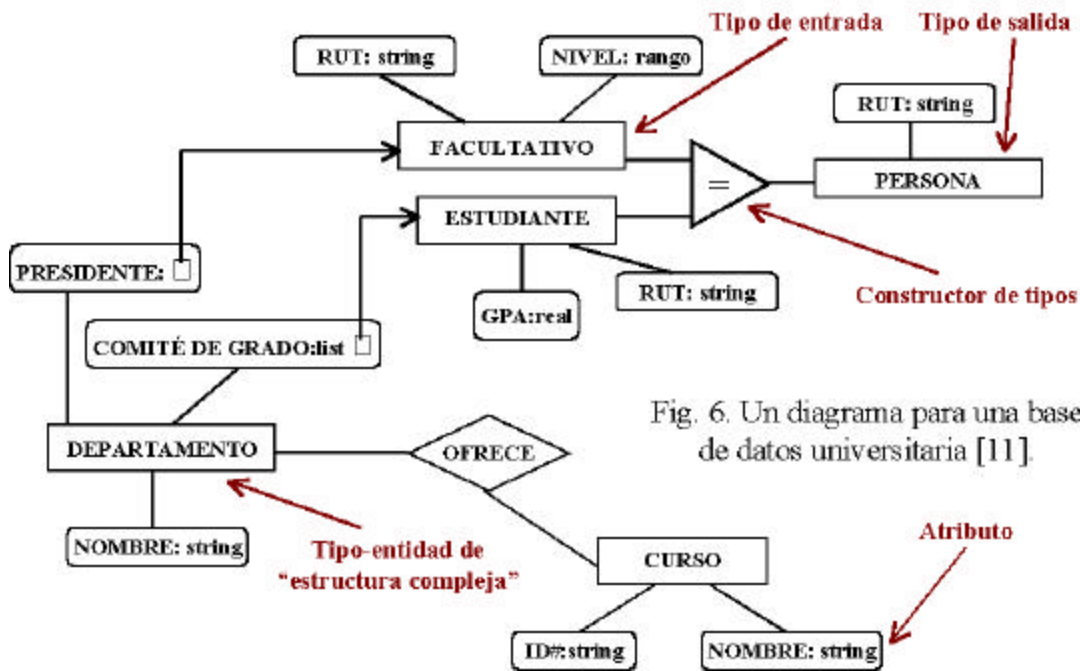


Fig. 6. Un diagrama para una base de datos universitaria [11].

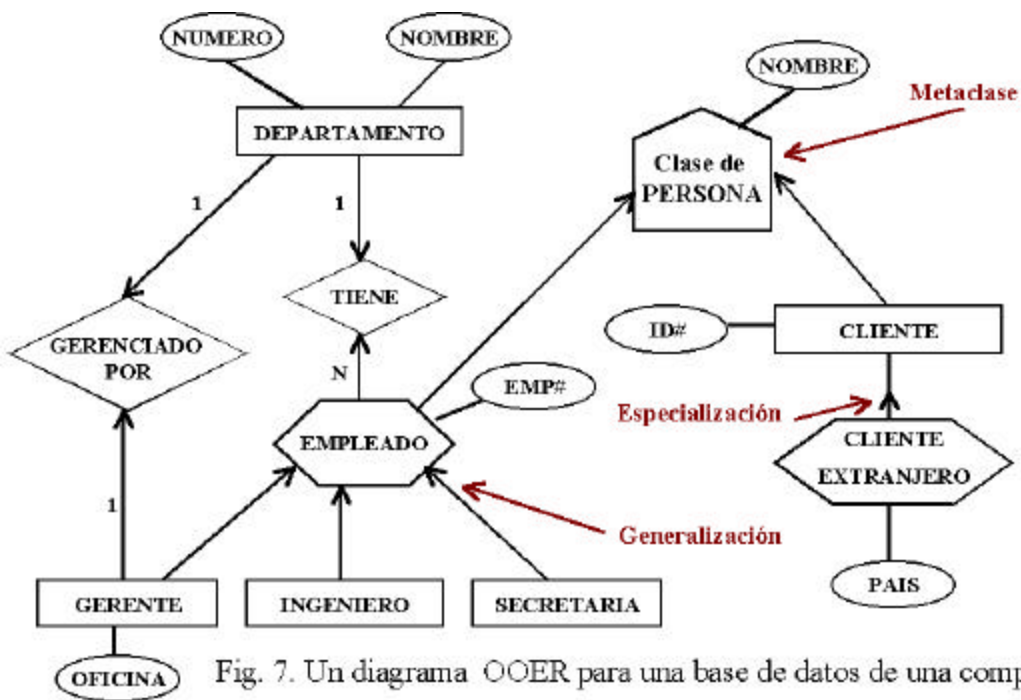


Fig. 7. Un diagrama OOER para una base de datos de una compañía.

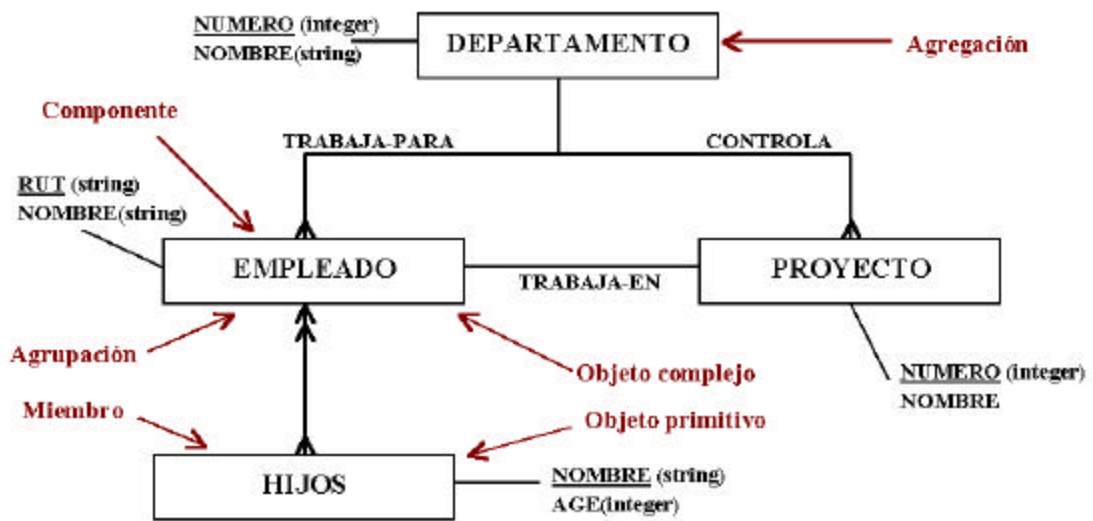


Fig. 8. Un modelo estático BIER para la base de datos de una compañía.



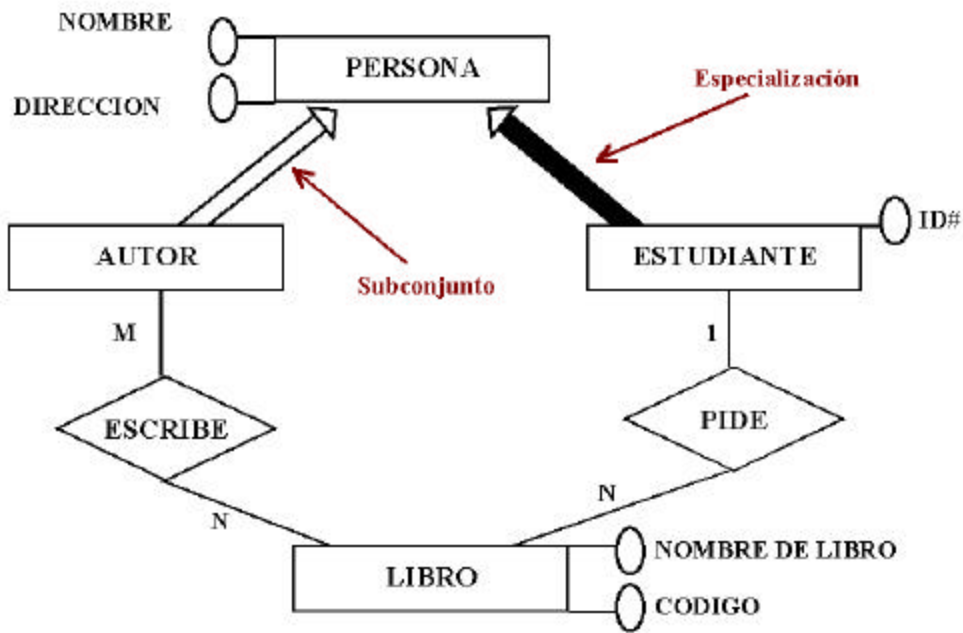


Fig. 9. Un ejemplo de OOERD estructural.

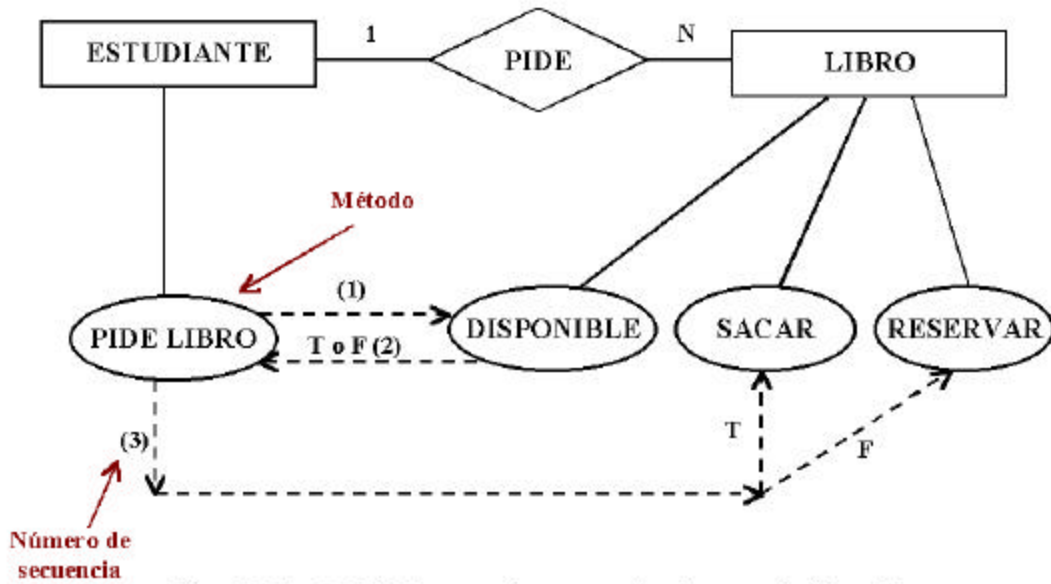


Fig. 10 Un OOERD para el proceso "préstamo de libros"

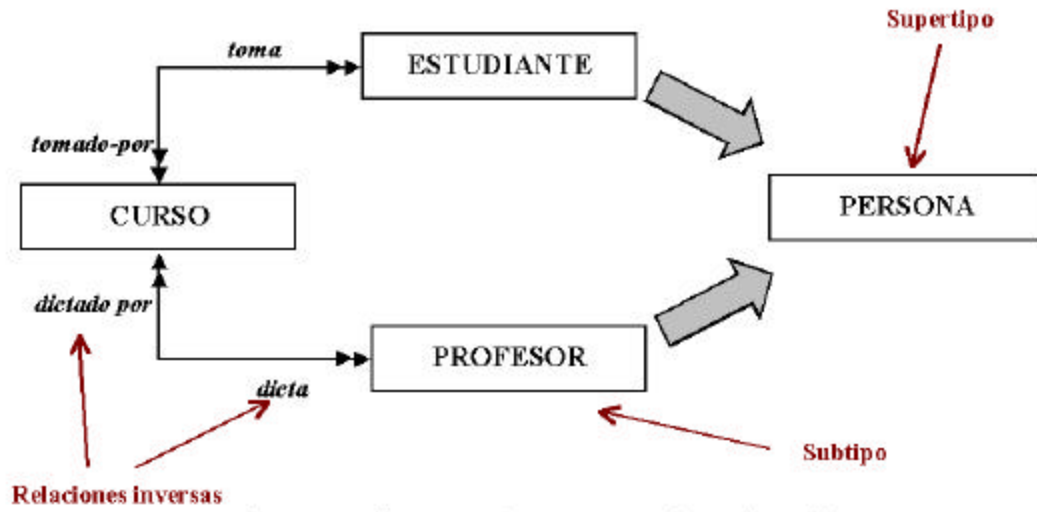


Fig.11 Un diagrama de esquema orientado a objeto