

## CAPÍTULO 4. DISEÑO CONCEPTUAL.

En la fase anterior, la de *recopilación de requisitos de usuario*, se pretende identificar de forma completa los problemas que la aplicación (el sistema) debe resolver, así como las características que dicha aplicación debería incluir, tanto desde el punto de vista estático (necesidades de información) como dinámico (operaciones sobre la información). Así pues, se realizan las actividades necesarias para recoger lo que los usuarios esperan obtener del sistema y los usos que éstos pretenden dar al sistema. Dichas actividades suelen incluir las siguientes:

- Identificar las áreas de aplicación principales y los grupos de usuarios que utilizarán el SBD
- Recopilar la documentación existente relativa a las aplicaciones actuales (si las hay), así como de otros tipos de documentos, como manuales de reglas del negocio, informes, normativas, etc.
- Realizar entrevistas a los usuarios, para establecer sus prioridades, preferencias, la importancia que les dan a las aplicaciones y a los datos, etc.
- Estudiar el entorno de operación del sistema actual (el que se va a reemplazar por el sistema nuevo que se pretende desarrollar, o el que va a interactuar con en sistema nuevo) y de los planes de aprovechamiento de la información. Esto incluye identificar las funciones que realiza el sistema actual (tipo de operación o función, frecuencia con que se llevan a cabo, tiempos de respuesta, restricciones, etc.), y el volumen de información tratada (flujos de información) dentro del sistema.

El resultado de estas actividades suele producir especificaciones de requisitos informales (escritas en lenguaje natural, normalmente) que suelen ser ambiguas y estar poco o nada estructuradas.

El propósito de esta etapa de *diseño conceptual* es representar estos requisitos informales mediante una descripción formal y completa, pero independiente de los criterios de representación utilizados por los sistemas gestores de bases de datos. El resultado de esta fase es el *esquema conceptual*, también conocido como *modelo conceptual de datos*.

### 4.1 ETAPAS DEL DISEÑO CONCEPTUAL.

Dos son las etapas en las que se puede considerar dividido el diseño conceptual: el análisis de requisitos y la conceptualización (o diseño del esquema conceptual).

#### Análisis de requisitos

En esta etapa de la fase de diseño conceptual de BD, se pretende analizar de forma más minuciosa y sistemática (siguiendo estándares concretos) las especificaciones de requisitos, para identificar (extraer) más claramente los requisitos relacionados con la información manipulada por el sistema.

Es necesario, pues, a) eliminar las ambigüedades (descripciones imprecisas o incorrectas de la realidad) de los requisitos recabados, b) de completarlos (los usuarios transmiten sus necesidades, objetivos, etc. pero pueden describirlos de manera vaga o con lagunas, que habrá que completar), c) dotarlos de una estructura, y d) entender realmente el significado de todos los términos.

El resultado debe ser una especificación clara y precisa de los requisitos (de información, en nuestro caso). En el libro [dMP93] se habla de que esta etapa produce un *esquema percibido* o *descriptivo* en lenguaje natural, que representa los requisitos de datos del sistema.

A lo largo de este apartado utilizaremos un ejemplo al que aplicaremos una serie de reglas prácticas para llevar a cabo esta tarea. Imaginemos que necesitamos diseñar una base de datos para una universidad y que, a partir de entrevistas con el personal de la universidad, hemos recopilado las especificaciones de requisitos de datos mostradas a continuación:

1 En una base de datos de una universidad, se representan	10 impartición de las clases (cada curso se imparte a lo sumo
2 datos sobre estudiantes y profesores. Para los estudiantes,	11 una vez en un día). Para estudiantes graduados se
3 se representa el apellido, edad, sexo, ciudad y provincia de	12 representa el nombre del tutor y el número total de
4 nacimiento, ciudad y provincia de residencia de sus	13 créditos en el último año. Para estudiantes de doctorado,
5 familias, lugares y provincias donde vivieron antes	14 se representa el título y área de investigación de su tesis.
6 (con el lapso que vivieron en cada uno), cursos que han	15 Para los instructores, se representa el apellido, la edad, el
7 aprobado, con nombre, código, profesor, nota y fecha.	16 lugar y provincia de nacimiento, nombre del departamento
8 Asimismo, se representan los cursos a los que asisten en	17 al que pertenecen, número de teléfono, título, situación y
9 la actualidad y para cada uno, día, lugar y horas de	18 temas de investigación.

Las siguientes son siete reglas prácticas que permiten descubrir y eliminar las posibles ambigüedades e imprecisiones:

#### **R1. Elegir el nivel apropiado de abstracción para los términos.**

Los términos abstractos se usan con frecuencia en enunciados de la vida real, en casos en que los términos específicos serían más apropiados para clarificar la situación. Las categorías generales son comunes en el lenguaje natural porque producen una comunicación rápida y eficaz en la que, comúnmente, la ambigüedad se resuelve por el contexto. Sin embargo, en el diseño conceptual se debe utilizar términos en un nivel correcto de abstracción, evitando los términos demasiado generales o demasiado específicos, especialmente si el diseñador no es un experto en el dominio de la aplicación.

En el ejemplo, aparecen los siguientes términos abstractos: *lugares*, *lapso* y *situación*; los términos apropiados correspondientes son: *ciudades*, *número de años*, *estado civil*.

#### **R2. Evitar el uso de “casos” en lugar de conceptos generales.**

Esta regla evita la fuente opuesta de ambigüedades: los usuarios de los sistemas de información adoptan, a veces, términos más específicos de lo necesario.

Por ejemplo, en una empresa de electrónica, un encargado puede decir: «*necesito conocer, a diario, la cantidad en existencia de chips*». El término *chip* no describe un concepto, sino más bien un caso del concepto correcto, es decir, *componente*. Por tanto, el término elegido debería ser *componente*.

#### **R3. Evitar las expresiones vagas o indirectas.**

En el lenguaje natural se usa con frecuencia la repetición deliberada y las expresiones indirectas.

Por ejemplo, se puede decir «*la persona sentada en la taquilla...*», en vez de «*el taquillero...*». La segunda oración indica un tipo específico de entidades TAQUILLERO como especialización de otro, PERSONA; mientras que la primera frase se refiere al tipo de entidades PERSONA e indica una interrelación del mismo con otro tipo de entidades, TAQUILLA. Así pues, la segunda oración permite una clasificación más clara de los conceptos.

Al usar este tipo de rodeos se incurre en el riesgo de expresar el significado de los conceptos en términos de *referencias implícitas* a otros conceptos (TAQUILLERO), en lugar de *referencias explícitas* a los conceptos mismos (PERSONA-Encargado\_De-TAQUILLA).

#### **R4. Elegir un estilo estandarizado de enunciado.**

En la “libre conversación” se usan muchos estilos sintácticos para lograr una comunicación más eficaz. Esta variedad de estilos debe evitarse en los textos que definen los requisitos; el uso de *categorías sintácticas simples* permite un modelado directo (y único) de los requisitos.

Idealmente, se deberían producir enunciados que respondan a algún estilo estándar; por ejemplo, las descripciones de los datos deberían ser claras y sencillas, de la forma:

<concepto> <verbo> <especificación>.

Las especificaciones que describen restricciones podrían tener la forma

*para <concepto> se cumple <propiedades>.*

Los enunciados que describen operaciones deberían utilizar, en la medida de lo posible, estructuras sintácticas no ambiguas, similares a las de los lenguajes de programación, como

*si <condición> entonces <acción> si no <acción> o cuando <condición> hacer <acción>.*

La aplicación completa de esta regla no es siempre posible o conveniente; el diseñador debe seleccionar un estilo apropiado que sea un término medio entre la estandarización y la expresividad.

#### **R5. Identificar sinónimos y homónimos.**

Los requisitos suelen resultar de las contribuciones de varios usuarios. Distintas personas pueden dar un mismo significado a diferentes palabras (sinónimos) o diferente significado a las mismas palabras (homónimos).

En general, el riesgo de los homónimos es mayor cuando el vocabulario de términos es pequeño, mientras que el riesgo de sinónimos es mayor cuando el vocabulario de términos es rico. Es más, si usuarios distintos adoptan vocabularios en diferentes niveles de abstracción, el riesgo de que aparezcan sinónimos es mayor. En el ejemplo, los tres términos diferentes: *profesor*, *tutor* e *instructor* (líneas 2, 12 y 15) se refieren al mismo concepto (son sinónimos); *lugares/lugar* se usa tres veces (líneas 5, 9 y 16), dos de ellas (líneas 5 y 9, o líneas 9 y 16) con diferentes significados (homónimo).

Estas situaciones deben ser clarificadas: para los sinónimos debe elegirse un único término, y para los homónimos debe utilizarse términos diferentes.

#### **R6. Hacer explícitas las referencias entre términos.**

No especificar las referencias entre los términos puede hacer que ambiguos algunos conceptos.

En el ejemplo no está claro si el número de teléfono es una propiedad de los profesores o de los departamentos. Nótese que los conceptos referidos pueden estar explícitamente mencionados en los requisitos (*profesores* y *departamentos*) u omitidos por completo (esto es cierto para *día* que se puede interpretar como *día de la semana* o *día del mes*; los términos *semana* y *mes* no aparecen en los requisitos).

**R7. Utilizar un glosario.** La creación de un glosario de términos es una buena forma (aunque requiere bastante tiempo) de entender el significado de los términos y eliminar las ambigüedades de los requisitos. Después de crear un glosario completo, en las descripciones de los requisitos sólo se deberían utilizar los términos del glosario. Para cada término, el glosario debe incluir: 1) su nombre; 2) una definición corta (de 5 a 20 palabras) que sea aceptable para todos los usuarios del término; 3) posibles sinónimos, es decir, términos que tengan igual significado para los usuarios (los sinónimos expresan el *área de equivalencia* del término); y 4) posibles *palabras clave*, es decir otros términos del glosario que estén lógicamente cercanos al término definido (las palabras clave expresan el *área de influencia* del término).

La aplicación de estas reglas usualmente produce requisitos más estructurados que al inicio de la actividad de diseño.

La siguiente tabla muestra todas las fuentes de ambigüedad y sus correcciones.

LÍNEA	TÉRMINO	NUEVO TÉRMINO	RAZONES PARA LA CORRECCIÓN
5	lugares	ciudades	<i>lugar</i> es una palabra genérica
6	lapso	número de años	<i>lapso</i> es una palabra genérica
9	actualidad	año actual	<i>actualidad</i> es ambigua
9	día	día de la semana	más específico
9	lugares	aulas	homónimo de <i>lugares</i> en la línea 5
10	clases	cursos	sinónimo de <i>cursos</i> en la línea 8
12	tutor	profesor	sinónimo de <i>profesor</i> en la línea 2
15	maestro	profesor	sinónimo de <i>profesor</i> en la línea 2
16	lugar	ciudad	lo mismo que en la línea 5
17	teléfono	teléfono del departamento	más específico
18	situación	estado civil	<i>situación</i> es un término ambiguo
18	temas	áreas de investigación	sinónimo de <i>área de investigación</i> en la línea 15

Alcanzado este punto, se reescriben las especificaciones utilizando las correcciones propuestas y además, es muy útil analizar el texto y descomponerlo en conjuntos de enunciados, tales que cada conjunto se refiera al mismo concepto. Esta actividad produce modificaciones locales del texto o el desplazamiento de fragmentos del mismo, y ayuda a estructurar los requisitos.

Si los enunciados relativos al mismo concepto están agrupados, es más fácil tener en cuenta todos los detalles acerca de ese concepto durante el diseño.

Las especificaciones de requisitos del ejemplo, escritos de nuevo, serían los siguientes:

---

#### Enunciados generales

En una base de datos de una universidad se representan datos sobre estudiantes y profesores.

---

#### Enunciados sobre estudiantes

Para los estudiantes, se representa el apellido, edad, sexo, ciudad y provincia de nacimiento, ciudad y provincia de residencia de sus familias, ciudades y provincias donde han vivido antes (con el número de años que vivieron en cada una), cursos que han aprobado, con nombre, código, profesor, nota y fecha.

---

#### Enunciados sobre cursos

Asimismo se representan los cursos a los que asisten en el año actual y, para cada uno, día de la semana, aulas y horas de impartición de los cursos (cada curso se imparte a lo sumo una vez en un día).

---

#### Enunciados sobre tipos específicos de estudiantes

Para estudiantes graduados se representa el nombre del profesor tutor en el último año. Para estudiantes de doctorado se representa el título y área de investigación de sus tesis.

---

#### Enunciados sobre profesores

Para profesores se representa el apellido, edad, ciudad y provincia de nacimiento, nombre del departamento al que pertenecen, número de teléfono del departamento, título, estado civil y área de investigación.

---

Y este sería el *esquema percibido* o *descriptivo* en lenguaje natural, que representa los requisitos de datos del sistema que hemos utilizado como ejemplo.

## Conceptualización

También suele denominarse *Diseño del Esquema Conceptual*.

En esta etapa se realiza un refinamiento y estructuración sucesivos del esquema percibido, para obtener el esquema conceptual de la BD. Para describir el EC suele utilizarse un modelo de datos de alto nivel, de forma que el EC sea independiente de características de implementación, y permita ver la información con todo su contenido semántico.

El diseñador, por tanto, debe intentar representar el *contenido de información* de la base de datos, sin considerar ni cómo será implementada esta información en el sistema real, ni la eficiencia de los programas que usarán esta información.

El esquema conceptual resume los requisitos de datos, y es una representación de alto nivel de abstracción de la base de datos que resulta muy útil para propósitos de documentación.

¿Por qué el EC debe ser independiente de la implementación (del SGBD)? Pues porque...

- La meta del diseño del EC es el entendimiento completo de la estructura, semántica (significado), interrelaciones y restricciones de la BD. Es mejor si se consigue esto independientemente de un SGBD particular, pues las peculiaridades de cada SGBD específico no deben influir sobre el diseño del EC.
- El EC es una descripción del contenido de la BD. La elección del SGBD y las decisiones de diseño posteriores pueden cambiar, pero eso no alterará el EC si es independiente del SGBD.
- Es fundamental emplear un modelo de datos de alto nivel para describir el EC, puesto que es más general y más expresivo que un modelo de datos de un SGBD particular.
- El EC puede servir como vehículo de comunicación entre usuarios, diseñadores y analistas. Los modelos de datos de alto nivel incluyen conceptos más fáciles de entender que los modelos de datos de nivel más bajo (específicos de SGBD).

En adelante emplearemos la terminología del Modelo Entidad/Interrelación Extendido (MEER).

En el **paso del esquema descriptivo a un primer esquema conceptual**, se habla de dos enfoques: el lingüístico y el de categorización de los objetos.

El **enfoque lingüístico** permite llegar a un primer EC a partir de conceptos lingüísticos, siguiendo estas pautas:

- Un **substantivo** (nombre común) que actúa como sujeto o complemento directo en una frase es, en general, un **tipo de entidad**, aunque podría ser un **atributo**. Por ejemplo, en la frase “*los socios piden prestados libros*”, existen dos posibles tipos de entidad: SOCIO (substantivo que actúa como sujeto) y LIBRO (que actúa como complemento directo).
- Los **nombres propios** suelen indicar **ocurrencias de un tipo de entidad**, por ejemplo “*Ende, M.*” indica una ocurrencia del tipo de entidad AUTOR.
- Un **verbo transitivo** o una **frase verbal** suele ser un **tipo de interrelación**. En la frase anterior “*piden prestados*” indica una interrelación entre los tipos de entidad LIBRO y SOCIO.
- Una **preposición** o **frase preposicional** entre dos **nombres** suele ser un **tipo de interrelación**, o también puede establecer la **asociación entre una entidad** y alguno de sus **atributos**. Por ejemplo, “*la editorial del libro...*” puede indicar la interrelación entre los tipos de entidad EDITORIAL y LIBRO, o bien puede asociar el atributo *editorial* al tipo de entidad LIBRO.

El enfoque de categorización de objetos, indica lo siguiente:

- Si un concepto tiene más **propiedades** además de su nombre, y/o describe un **tipo de objetos de datos con existencia autónoma**, y/o se usa como **operando** en una sentencia de **selección** (obtención de datos), **borrado o inserción**, es conveniente utilizar una **entidad** para representarlo. Por ejemplo, en la biblioteca los libros poseen propiedades como el título, idioma, número de copias, etc. LIBRO es una entidad por tener dichas propiedades, y porque su existencia es independiente de otros conceptos. Por otro lado, cuando un socio deja de serlo se elimina su ficha de la base de datos, así que SOCIO es una entidad, por ser un operando en una sentencia de borrado.
- Si un concepto tiene una estructura simple, sin propiedades relevantes asociadas, y/o describe un objeto de datos al que se le asigna un **valor**, y/o se utiliza como **operando** en una **operación aritmética, booleana o de string de caracteres**, es conveniente representar dicho concepto con un **atributo** de otro concepto al cual se refiere. Por ejemplo, el título de un libro no tiene propiedades interesantes a parte de su propio valor, y además, se puede consultar si existe un libro cuyo título es *Momo* (operación *booleana* en la que el título aparece como operando), por todo ello *título* es un atributo de LIBRO.
- Si los requisitos contienen un concepto que proporciona un **vínculo lógico** entre dos o más entidades (pero sin tener las características de una entidad), y/o hace posible la **selección de una entidad** a través de una referencia a un atributo de otra entidad, entonces es conveniente representar dicho concepto mediante una **interrelación**. Por ejemplo, podemos seleccionar los libros que ha escrito cierto autor cuyo nombre es conocido; de este modo surge una interrelación ESCRIBIR, ya que permite seleccionar entidades (LIBRO) mediante una referencia a un atributo (*nombre*) de otra entidad (AUTOR).
- Si uno o varios conceptos son **casos particulares** de otro, es conveniente representarlos con una jerarquía de **generalización/especialización**.

En definitiva, se trata de reglas basadas en el papel o *rol* que un determinado objeto de datos desempeña en el procesamiento de información.

Por tanto, y combinando ambos enfoques, si se dice que...

"*La biblioteca presta un conjunto de libros de los que interesa recoger, además del título, el idioma y la fecha de publicación...*", según el primer enfoque, se puede considerar que LIBRO, al ser un nombre o sustantivo que actúa como complemento directo, es un tipo de entidad o bien un atributo; según el segundo enfoque, se puede concluir definitivamente que es un tipo de entidad, ya que LIBRO es un *concepto* que tiene más propiedades que su nombre (idioma, fecha, etc.), con existencia autónoma, y que se utilizará en sentencias de selección, borrado e inserción (al consultar sus datos, eliminar o introducir libros en la base de datos).

"*Cristina Aliaga ha tomado prestados los libros Momo y La Celestina*", estos nombres propios indican ocurrencias de entidades (el primero de la entidad SOCIO, y los otros dos de la entidad LIBRO).

"*Un libro puede haber sido traducido a varios idiomas...El idioma de un libro...*", el *idioma* es un *concepto* al que se le asigna un valor (como *inglés, italiano, alemán*, etc.) sin más propiedades que su valor, y que actúa como operando en una operación *booleana* o de *string* de caracteres. Por ejemplo, se podrá consultar los libros escritos en inglés, o si un determinado libro está en italiano, etc. En el enfoque lingüístico, la proposición *de* asocia el atributo *idioma* al tipo de entidad LIBRO.

"*Los autores escriben libros...*", según el primer enfoque, se puede establecer que ESCRIBIR es una interrelación entre LIBRO y AUTOR, ya que es un verbo transitivo. También se puede considerar, según el segundo enfoque, como un *concepto* que representa el vínculo entre LIBRO y AUTOR, y que hace posible la selección de una entidad (LIBRO) a través de una referencia a un atributo (*nombre*) de otra entidad (AUTOR), por ejemplo al consultar los libros escritos por "*Wilde, O.*".

Por otra parte, en la especificación de los requisitos suelen aparecer dos verbos que resultan de especial interés: *ser* y *tener*.

1) “**ES**” permite crear **jerarquías de generalización/especialización entre entidades**.

Un ejemplo sería este: “... *tanto un libro como un artículo de nuestra base de datos **son** documentos ...*”, se puede establecer una jerarquía en la que el supertipo es el tipo de entidad DOCUMENTO y los subtipos son LIBRO y ARTICULO.

2) “**TIENE**”, puede denotar:

- a) **Interrelación** entre entidades. Por ejemplo, la frase “...*los libros pueden **tener** más de un autor...*” establece la interrelación entre los tipos de entidad LIBRO y AUTOR. En este caso, *tener* actúa de forma totalmente análoga a un verbo transitivo.
- b) “**Ocurrencia de**”. Por ejemplo, se puede decir “...*un libro **tiene** varios ejemplares...*” en el sentido de que un ejemplar es un ocurrencia de un libro. En este caso, el identificador (principal) de la entidad que es ocurrencia (EJEMPLAR) se suele formar con el identificador del tipo de entidad “principal” (LIBRO) junto con un atributo “discriminante” de la ocurrencia. Por ejemplo, si se identifica a los libros con un código de cinco dígitos (99999), a los ejemplares correspondientes se les concatenaría un discriminante de dos dígitos (clave parcial), con lo que quedarían identificados como 99999-01 (el primer ejemplar), 99999-02 (el segundo ejemplar del libro 99999), etc. (dependencia en identificación, y por tanto en existencia).
- c) **Asociación de entidades con sus atributos**. Por ejemplo “...*los libros **tienen** un título, un año de publicación y un idioma determinado...*”, asocia el tipo de entidad LIBRO con los atributos *título, idioma, año*.

(Ojo en 1) y 2b) es importante **revisar las interrelaciones** en las que participan los tipos de entidad afectados, pues puede que ya no estén en el nivel adecuado (caso de “...*los socios toman prestados **libros**...*” que ahora debería pasar a “...*los socios toman prestados **ejemplares** de libros...*”). También revisar que los **atributos** queden en el tipo de entidad adecuado. En 1) las interrelaciones y atributos comunes a los subtipos deben quedar en el supertipo.)

---

Otro aspecto a tener en cuenta, es que del uso del singular o plural es posible deducir ciertos tipos y grados de las interrelaciones, así como cardinalidades de las entidades; Por ejemplo, de la frase “...*un libro es escrito por varios autores...*” y “...*un autor puede escribir varios libros...*” se puede deducir que la interrelación ESCRIBIR es de grado 2 y de tipo M:N (muchos a muchos). Además, de la frase “... *un libro puede ser escrito por un anónimo...*” es posible deducir que la cardinalidad mínima de la entidad AUTOR en la interrelación es 0.

---

Modelar la realidad (los conceptos y objetos de la realidad) es un problema complicado, normalmente abierto a varias soluciones alternativas, puesto que la misma realidad se puede modelar de formas diversas.

Hasta ahora hemos visto unos criterios generales que pueden servir como guía en este proceso de categorización de los objetos de datos del UoD, para obtener un primer esquema conceptual (expresado con los conceptos del MEER), a partir del análisis lexicográfico (lingüístico, lexicológico, filológico...) de los requisitos (esquema percibido). Aún así, siguen existiendo problemas como la distinción entre *atributo* y (*tipo de*) *entidad*, entre *generalización* y *atributo*, o entre *atributo compuesto* y *conjunto de atributos simples*.

### ¿Tipo de entidad o Atributo simple?

Se debe elegir un tipo de **entidad** cuando se entiende que es posible asociar características (atributos, interrelaciones, generalizaciones) con el concepto, ya sea ahora o más adelante en el proceso de diseño.

- Si el concepto tiene asociados otros atributos. Por ejemplo, en un principio pudo considerarse la editorial como un atributo (*editorial*) de LIBRO; posteriormente, al detectar que tiene a su vez otros atributos como *dirección, año de fundación, nacionalidad*, etc., se creará el tipo de entidad EDITORIAL.
- Si el concepto está relacionado con otros tipos de entidad. Por ejemplo, si la editorial se considerara como un atributo de LIBRO, no se podría reflejar las posibles interrelaciones existentes entre las editoriales y los autores (por ejemplo, para especificar que la editorial “Alfaguara” *ha contratado* a “Pérez Reverte, A.” para la publicación de una serie de 7 novelas).
- Los atributos multivalorados deberían ser modelados como tipos de entidad. Por ejemplo, si se considerara *idioma* como atributo de la entidad LIBRO, el hecho de que un mismo libro puede encontrarse en varios idiomas (**ojo: esto no se cumple para los ejemplares**) indica que es un atributo multivalorado, por lo que sería conveniente considerar el tipo de entidad IDIOMA interrelacionado con LIBRO.

Se debe elegir un **atributo** cuando el concepto posea una estructura atómica simple y no le sea aplicable ninguna propiedad de interés. Por ejemplo, para una aplicación de un concesionario dedicado a la venta de coches, el concepto “color” suele ser un atributo del tipo de entidad COCHE; sin embargo, si la aplicación se refiere a una fábrica de coches, y en particular al proceso de coloreado, entonces puede convertirse en una entidad COLOR, con los atributos *nombre, código\_de\_color, número\_necesario\_de\_pinturas, anticorrosivo*, etc.

### □ ¿Generalización/Especialización o Atributo simple?

Debe usarse una **generalización/especialización** cuando ocurre que (ahora o quizá más adelante en el proceso de diseño) alguna característica (atributo, interrelación) se asocie sólo con alguno de los tipos de entidad del nivel más bajo (subtipos). De lo contrario, debe elegirse un **atributo**.

Por ejemplo, podría utilizarse *sexo* como atributo de PERSONA, pero si suponemos que estamos modelando la información relacionada con un hospital y que es necesario conocer el número de embarazos de las pacientes por tener incidencia en los posibles tratamientos a que éstas pueden someterse, en este caso resultará más conveniente no utilizar el atributo *sexo*, sino crear los tipos de entidad HOMBRE y MUJER como subtipos de PERSONA, y *número\_de\_embarazos* será un atributo de MUJER.

Otro ejemplo sería utilizar un atributo *pensionista* que indicara si una persona se ha jubilado ya o no; si existiera una interrelación en la que únicamente participaran las personas pensionistas, como por ejemplo la que asocia a cada jubilado con el centro de la 3ª edad al que pertenece, debería crearse PENSIONISTA como subtipo de PERSONA.

Por otro lado, una especialización de PERSONA basada en el *color\_del\_cabello* no suele ser útil, porque pocas veces se asocian rasgos específicos a los rubios o a los canosos; sin embargo, esta generalización podría servir para una base de datos de estilistas, donde el tratamiento del cabello depende de los colores del mismo.



## □ ¿Atributo Compuesto o Conjunto de Atributos Simples?

Se debe elegir un **atributo compuesto** cuando resulta natural asignarle un nombre a una agrupación de otros atributos. Es ideal para situaciones en la que se hace referencia al atributo compuesto como unidad, y también se hace referencia a sus componentes de forma individual. Por ejemplo, *dirección* es una buena abstracción de los atributos *calle, ciudad, provincia, código\_postal*. Se debe elegir un **conjunto de atributos simples** cuando representan propiedades independientes y apenas se necesita acceder a todos ellos vistos como una unidad. Por último, si sólo se hace referencia al conjunto de atributos como unidad, deben representarse mediante un **atributo único simple**.

## 4.2 DISEÑO DEL ESQUEMA CONCEPTUAL

### 4.2.1 Enfoques para el diseño de esquemas conceptuales

#### ***Centralizado***

Los requisitos de las diferentes aplicaciones y grupos de usuarios se combinan en un único conjunto (catálogo) de requisitos, antes de iniciarse el diseño.

Se diseña un único esquema conceptual, que corresponde al conjunto global de los requisitos.

Cuando hay muchos usuarios, combinar todos sus requisitos puede ser una tarea difícil y lenta. Este enfoque se basa en que una autoridad centralizada, el (equipo del) diseñador de la base de datos, decide cómo combinar los requisitos de los diferentes usuarios y aplicaciones, y diseña el esquema conceptual de la BD. Una vez diseñado y terminado el EC, el DBA especificará los esquemas externos necesarios para cada grupo de usuarios y aplicación.

#### ***Integración de vistas***

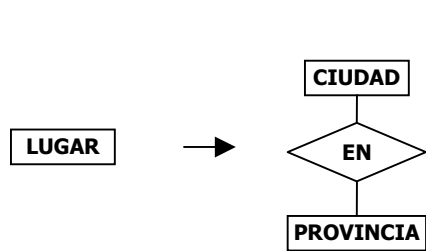
No se combinan los requisitos, sino que se diseña un esquema para cada grupo de usuarios que corresponde sólo a sus requisitos propios (será la vista de los datos que tiene cada grupo). Es necesaria una etapa de integración de vistas, en la que combinar los esquemas para obtener un esquema conceptual global de la BD. Las vistas individuales pueden utilizarse después como base para los esquemas externos.

La diferencia entre ambos enfoques está en qué, cómo, quién y cuándo se realiza la combinación.

- En el centralizado, lo que se combina son los requisitos recopilados, lo realiza manualmente el (equipo del) diseñador de la base de datos, antes de diseñar el esquema global.
- En el de integración de vistas, lo que se combina son los esquemas conceptuales diseñados previamente (para cada grupo de usuarios o aplicación a partir de sus requisitos). Es el (equipo del) diseñador de la base de datos quien integra dichos esquemas (vistas) para formar el EC global. Puede hacerlo manualmente, aunque si la BD es grande (con decenas de grupos de usuarios, muchas entidades complejas con relaciones complejas entre ellas, etc.) será necesario seguir un método y usar herramientas automatizadas que le ayuden en la integración. Más adelante profundizaremos en este enfoque.

Dado un conjunto de requisitos, ya sea la especificación de requisitos global (1er enfoque) o una parte de la especificación, correspondiente a un grupo de usuarios (2º enfoque), se debe crear (diseñar) un esquema conceptual EC que los describa y satisfaga.

Existen varias **estrategias** para diseñar esquemas conceptuales y la mayoría siguen un **proceso incremental**, es decir, se parte de una versión preliminar del esquema (derivada de los requisitos tras su análisis lexicográfico o categorización de objetos) y se modifica, se refina o se desarrolla, realizando las **transformaciones** adecuadas en el esquema, hasta llegar a la versión definitiva del EC.

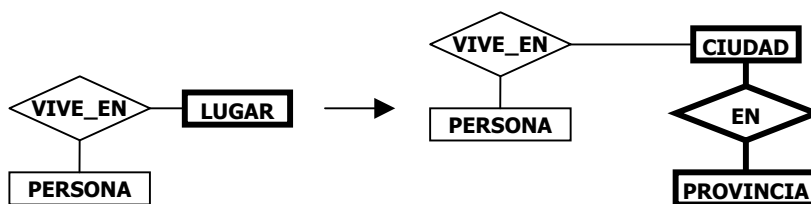


En el ejemplo mostrado en la figura de la izquierda, el segundo esquema se obtiene a partir del primero mediante la aplicación de una transformación.

El propósito de la transformación es refinar el concepto abstracto LUGAR en términos de conceptos más específicos: los dos tipos de entidad CIUDAD y PROVINCIA y la interrelación binaria EN que se define entre ellas.

Las transformaciones de esquemas tienen tres características bien definidas:

1. **Cada transformación de esquema tiene un esquema inicial, esto es, el esquema al que se aplica la transformación, y un esquema resultante, el efecto de aplicar la transformación.** En la figura, el esquema inicial consiste en una entidad y el esquema resultante en un par de entidades conectadas por una interrelación.
2. **Cada transformación de esquema crea una correspondencia entre nombres de conceptos del esquema inicial y nombres de conceptos del esquema resultante.** En la figura anterior, la correspondencia es entre el nombre LUGAR y los nuevos nombres CIUDAD, EN y PROVINCIA.
3. **Los conceptos del esquema resultante deben heredar todas las conexiones lógicas definidas para los conceptos del esquema inicial.** Los elementos dentro del esquema inicial suelen estar conectados con otros elementos; por ejemplo un tipo de entidad puede estar conectada con interrelaciones, atributos, o con otros tipos de entidad en jerarquías de generalización/especialización. Cuando una entidad se transforma en un conjunto de conceptos, esos conceptos deben heredar todas las conexiones que tenía dicha entidad en el esquema anterior. En la figura siguiente, el vínculo lógico entre PERSONA y LUGAR, dado por la interrelación VIVE\_EN, es heredado por la entidad CIUDAD.



Para realizar este tipo de transformaciones se utiliza un conjunto de *primitivas de refinamiento o transformación*; cada primitiva se aplica a un esquema inicial y produce un esquema resultante.

#### 4.2.2 Primitivas del diseño conceptual

Las primitivas se clasifican en dos grupos: descendentes y ascendentes. Las primitivas descendentes corresponden a refinamientos puros, es decir, refinamientos que se aplican a un concepto y producen una descripción más detallada de ese concepto. En cambio, las primitivas ascendentes introducen conceptos nuevos y propiedades que no aparecían en versiones anteriores del esquema.

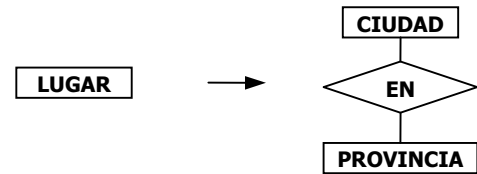
## Primitivas de transformación descendentes (*top-down*).

Propiedades:

- Tienen una estructura simple: se aplican sobre un concepto único (un elemento del esquema inicial) y el resultado se compone de un pequeño conjunto de conceptos (en el esquema resultante).
- Todos los nombres se refinan dando lugar a nuevos nombres, que describen el concepto original con mayor detalle (a un nivel de abstracción menor).
- Los vínculos lógicos del concepto inicial son *heredados* por un sólo concepto del esquema resultante.

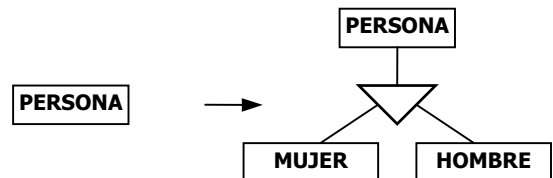
### D1: entidad → entidades interrelacionadas.

Refina una entidad para producir una interrelación entre dos o más entidades. Se utiliza cuando se detecta que la entidad describe dos conceptos lógicamente conectados entre sí.



### D2: entidad → generalización/especialización.

Refina una entidad para producir una jerarquía de generalización/especialización.



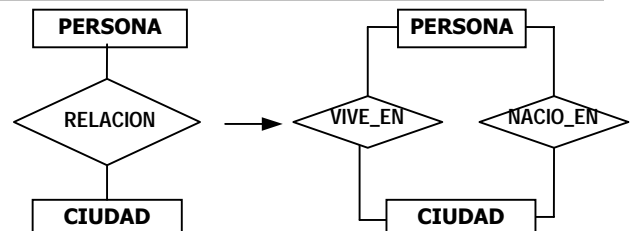
### D3: entidad → entidades no interrelacionadas.

Divide una entidad en un conjunto de entidades independientes. Permite la introducción de nuevos tipos de entidad, sin establecer interrelaciones ni jerarquías entre ellos.



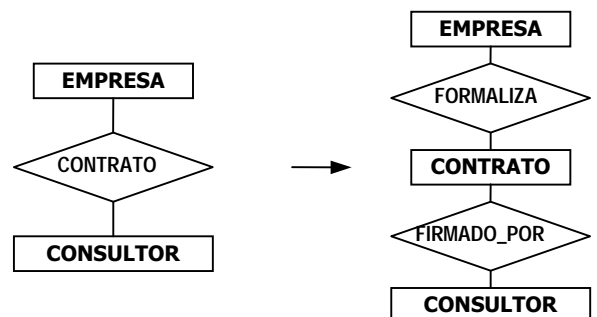
### D4: interrelación → interrelaciones paralelas.

Refina una interrelación para producir dos o más interrelaciones entre las mismas entidades. Se emplea cuando se detecta que la interrelación describe dos o más conceptos distintos ligando las mismas entidades.



### D5: interrelación → entidad con interrelaciones.

Refina una interrelación para producir una ruta de entidades e interrelaciones. Se utiliza cuando se detecta que una interrelación entre dos conceptos describe un concepto con existencia autónoma /y/o que tiene bastantes conceptos propios) debe expresarse mediante un tercer concepto que estaba oculto en la representación anterior.



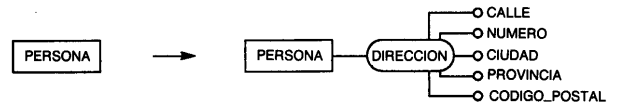
### D6: desarrollo de atributos.

Refina una entidad o una interrelación, introduciendo sus atributos (conjunto de atributos simples).



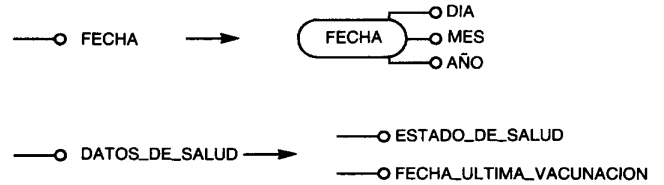
### D7: desarrollo de atributos compuestos.

Refina una entidad o una interrelación, introduciendo un atributo compuesto.



### D8: refinamiento de atributos.

Refina un atributo simple para producir un atributo compuesto, o bien un grupo de atributos independientes.



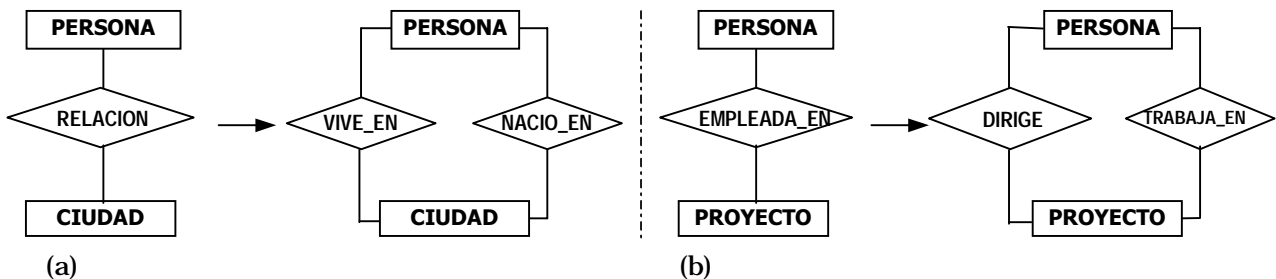
Las primitivas anteriores tienen esquemas resultantes simples y pueden considerarse como las herramientas más sencillas para refinar un esquema. Deben utilizarse lo máximo posible para conseguir que el proceso de diseño sea estructurado, comprensible y fiable. Existen otras transformaciones más complejas que también pueden clasificarse como descendentes; un ejemplo sería refinar una entidad para dar un conjunto de entidades e interrelaciones.

**Al aplicar las primitivas de refinamiento se debe respetar, implícitamente, ciertas restricciones.** Por ejemplo en la primitiva D5, existe una relación clara entre las cardinalidades mínima y máxima de la interrelación en el esquema inicial y en el esquema resultante.

Por ejemplo, si el esquema inicial posee una interrelación 1:1, el esquema resultante no puede incluir dos interrelaciones N:M (muchos a muchos).

Estas restricciones dependen también del significado de los conceptos sometidos a refinamiento.

Considérense por ejemplo, dos aplicaciones diferentes de la primitiva D4, mostradas en la figura siguiente:



En la primera situación (a), la interrelación RELACIONADA\_CON es un modelado preliminar de dos interrelaciones distintas e independientes, producidas por el refinamiento (NACIO\_EN y VIVE\_EN); en este caso no existe restricción sobre las cardinalidades mínima y máxima de estas dos interrelaciones respecto de la inicial: una persona está relacionada con una o muchas ciudades y una ciudad puede estar relacionada con ninguna o muchas personas. Una persona nació en una y sólo una ciudad y vive (está empadronado) en una y sólo una ciudad. Una ciudad puede ser ciudad natal de ninguna o muchas personas y ser el domicilio de ninguna o muchas personas.

En la segunda situación (b), las interrelaciones DIRIGE Y TRABAJA\_EN constituyen una partición de la interrelación EMPLEADA\_EN, de forma que el director de cada proyecto está entre las personas que trabajan en el mismo. Si se establece que una persona puede estar empleada en uno y sólo uno de los proyectos y que en un proyecto trabajarán una o muchas personas, entonces una persona podrá trabajar en un único proyecto, y dirigir cero o un único proyecto; un proyecto estará dirigido por una única persona y en él trabajarán una o muchas personas. Es decir si PERSONA - EMPLEADA\_EN - PROYECTO es (1,1):(1,n), entonces PERSONA - DIRIGE - PROYECTO debe ser (0,1):(1,1) y PERSONA - TRABAJA\_EN - PROYECTO debe ser (1,1):(1,n).

## Primitivas de transformación ascendentes (*bottom-up*)

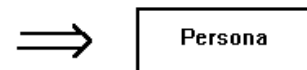
Las primitivas ascendentes introducen nuevos conceptos y propiedades que no aparecían en versiones anteriores del esquema, o modifican conceptos existentes. Las primitivas ascendentes se usan en el diseño de un esquema cuando se descubren rasgos del dominio de aplicación que no fueron captados en ningún nivel de abstracción en versiones anteriores del esquema.

Las primitivas ascendentes también se aplican cuando se fusionan esquemas diferentes para formar un esquema global más amplio, conocido como *integración* (véase más adelante “integración de vistas”).

Las primitivas ascendentes más utilizadas en el curso de un diseño son las siguientes:

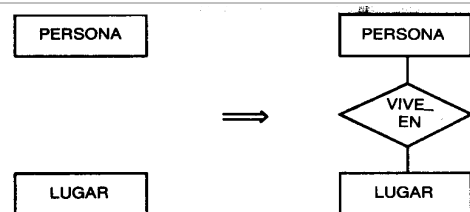
### A1 : Generación de tipo de entidad.

Genera una nueva entidad. Se usa cuando el diseñador descubre un nuevo concepto con propiedades específicas, que no aparecía en el esquema anterior.



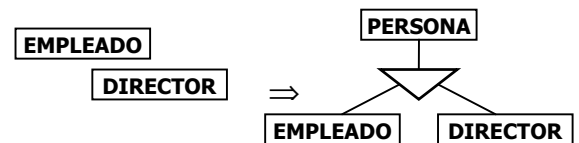
### A2: Generación de interrelación.

Genera una nueva interrelación entre entidades definidas anteriormente.



### A3 : Generación de generalización.

Crea una nueva entidad que se toma como una generalización de entidades definidas previamente.



### A4: Agregación de atributos.

Se utiliza cuando se detecta que un tipo de entidad o de interrelación puede verse como un conjunto o agregación de atributos.



### A5: Agregación de atributo compuesto.

Crea un atributo compuesto como la agregación de un conjunto de atributos simples de una entidad o interrelación ya definida.



Es necesario destacar que la aplicación de la primitiva ascendente A3 hace necesario comprobar si, debido a la nueva generalización, las propiedades de las entidades iniciales deben pasar a la entidad de nivel superior. En la figura se introduce una nueva entidad PERSONA y una relación supertipo/subtipo entre PERSONA y EMPLEADO y DIRECTOR; como consecuencia, atributos tales como *nombre* y *apellido* y las interrelaciones NACIO\_EN y VIVE\_EN deben ser asignados a la entidad genérica, puesto que son comunes a las entidades que ahora actúan como subtipos. Vemos que la aplicación de primitivas ascendentes puede obligar a los diseñadores a investigar las consecuencias de la transformación en el resto del esquema (es decir, a revisar las interrelaciones y atributos de las entidades afectadas).

### 4.2.3 Estrategias para el diseño de esquemas conceptuales

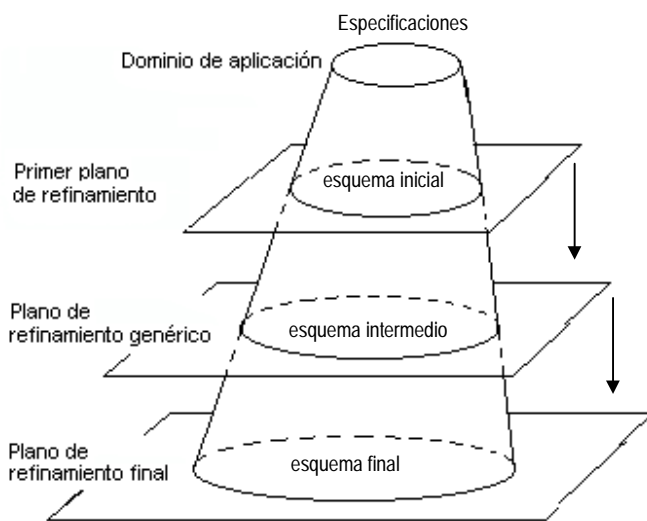
Se distinguen cuatro estrategias para el diseño de esquemas conceptuales de base de datos: *descendente*, *ascendente*, *centrífuga* y *mixta*. Cada una de ellas se caracteriza por el uso de determinados tipos de primitivas. A lo largo de este apartado, seguiremos un sencillo ejemplo al que se aplicaremos estas estrategias; supongamos que necesitamos diseñar el esquema de una base de datos demográfica que representa propiedades de las personas y lugares geográficos, cuyos requisitos son los siguientes:

En la base de datos de un censo se consideran las siguientes propiedades de las personas: nombre, apellido, sexo, edad, lugar de nacimiento, lugar de residencia, años de residencia, situación militar de los hombres, **apellido de soltera de las mujeres (\*obviamente, hablamos de un censo no español\*)**.

Los lugares pueden ser estados extranjeros o ciudades nacionales. Cada uno tiene un nombre y número de habitantes (que representa la población total en el caso de los estados extranjeros) y los nombres de las regiones o ciudades.

#### 1. Estrategia descendente (*top-down*)

En la estrategia descendente se obtiene un esquema mediante refinamientos sucesivos del esquema conceptual inicial; el esquema se expande gradualmente, aplicando primitivas de refinamiento descendente; cada primitiva introduce nuevos detalles en el esquema. El proceso termina cuando están representados todos los requisitos de información.



La figura de la izquierda muestra una representación abstracta en forma de cono del proceso de diseño: en cada transformación descendente el diseñador se mueve de un plano (nivel) de diseño a otro (modificando el esquema mediante la aplicación de primitivas), mientras que la parte del dominio de aplicación, representada por el cono, se mantiene inalterable: cada nivel contiene un esquema que describe la misma información, pero con un grado de detalle distinto.

Es importante señalar que, en una estrategia descendente «pura», **todos los conceptos que se representan en el esquema final deben estar presentes en cada plano (nivel) de refinamiento**.

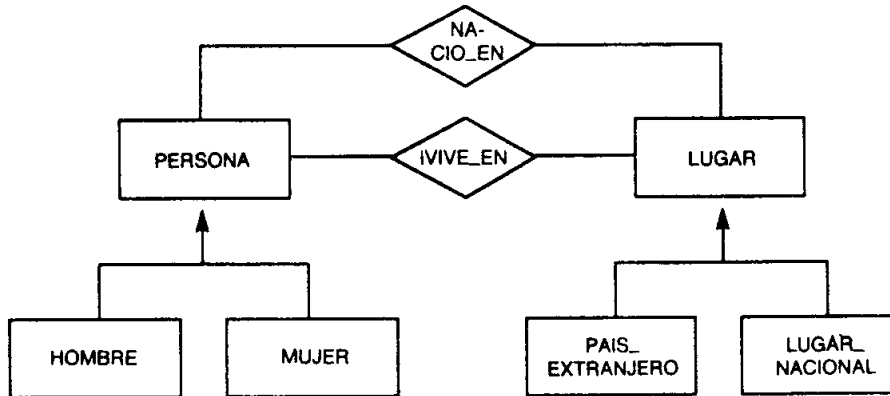
Para abordar el diseño de la base de datos demográfica vamos a utilizar un enfoque descendente puro. El primer esquema representa sólo un concepto, la entidad `DATOS_DEMOGRAFICOS`, que modela toda la información de forma muy abstracta.



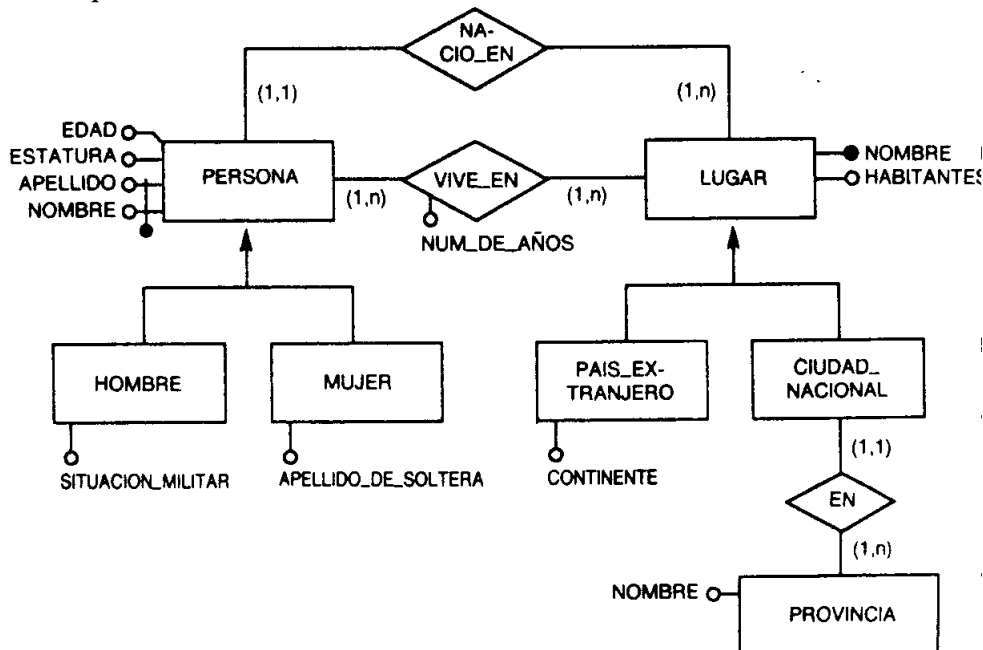
El segundo esquema refina esa entidad para producir dos entidades, `DATOS_DE_PERSONAS` y `DATOS_DE_LUGARES`, y una interrelación entre ellas (aplicación de la primitiva de transformación D1). Esto representa explícitamente el hecho de que los datos se refieren a dos tipos de objetos de la realidad: personas y lugares.



En el tercer esquema se realizan tres refinamientos distintos: a) se distinguen dos interrelaciones diferentes entre personas y lugares (D4), b) se introducen dos tipos distintos de personas (D2), y c) se distinguen dos tipos diferentes de lugares (D2).



El cuarto esquema introduce los detalles finales: se especifican los atributos (D6), se refina la entidad LUGAR\_NACIONAL (D1), y se especifican las cardinalidades y los atributos identificadores de cada entidad. Este es el esquema final.



Al emplear refinamientos descendentes, el diseñador puede analizar un concepto cada vez, ignorando otros detalles, con lo que se simplifica el proceso iterativo. Este enfoque es **aplicable cuando el diseñador es capaz de construir en su mente una vista de alto nivel, que abarca todos los requisitos** considerados (esto es especialmente difícil en el caso de bases de datos muy grandes).





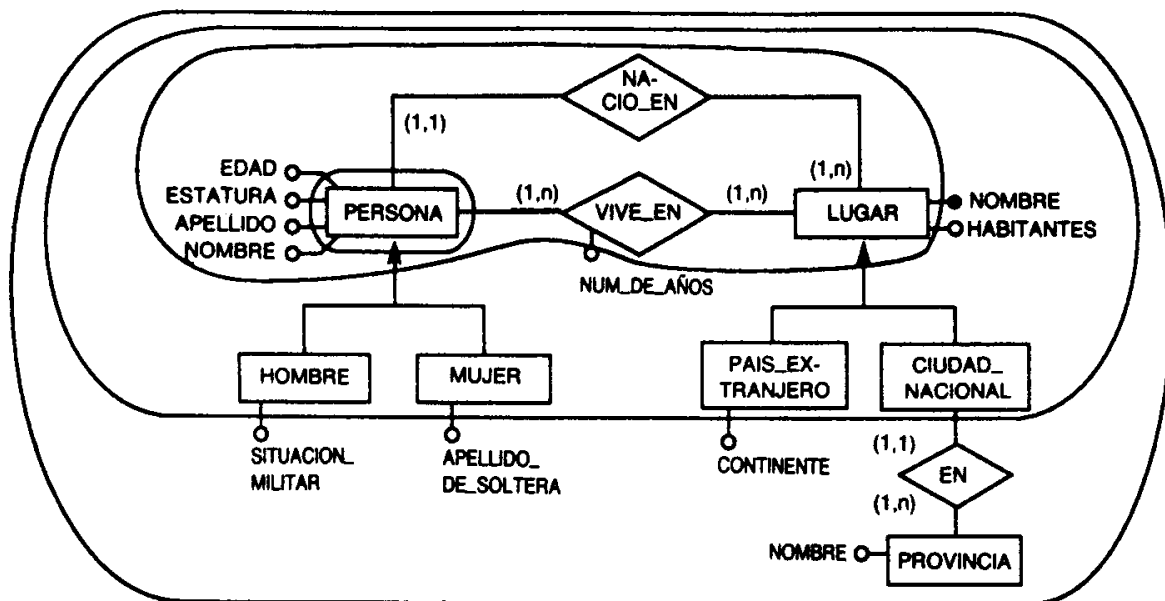
### 3. Estrategia centrífuga (*inside-out*).

La estrategia centrífuga puede considerarse un caso especial de estrategia ascendente y también suele denominarse estrategia “de dentro a afuera”. Primero se fijan los conceptos más evidentes, y luego se procede con un movimiento similar al de una mancha de aceite: agregando los conceptos más cercanos al concepto inicial, y *navegando* después hacia los más distantes.

En la siguiente figura se muestra cómo se aplica este enfoque a la base de datos demográfica. PERSONA es el concepto más importante de este ejemplo; si nos desplazamos de la entidad PERSONA hacia fuera, primero introducimos las entidades directamente relacionadas con las personas mediante interrelaciones y generalizaciones; de esta forma, descubrimos la entidad LUGAR, la interrelación VIVE\_EN, las entidades HOMBRE y MUJER, y la generalización que les relaciona con PERSONA.

Ahora se puede buscar conceptos cercanos a las entidades recién descubiertas. Esto conduce a representar: 1) la nueva interrelación NACIO\_EN, entre PERSONA y LUGAR; 2) las nuevas entidades PAIS\_EXTRANJERO y CIUDAD\_NACIONAL, ambas especializaciones de LUGAR; y 3) dos nuevos atributos de las entidades HOMBRE y MUJER. Finalmente, se puede introducir la entidad PROVINCIA y la interrelación que la une a CIUDAD\_NACIONAL. Luego, se completa el esquema con las cardinalidades y los identificadores.

En la figura, las áreas delineadas representan las diversas capas introducidas progresivamente en el esquema.



Una ventaja de la estrategia centrífuga es que no necesita de ninguna integración entre esquemas.

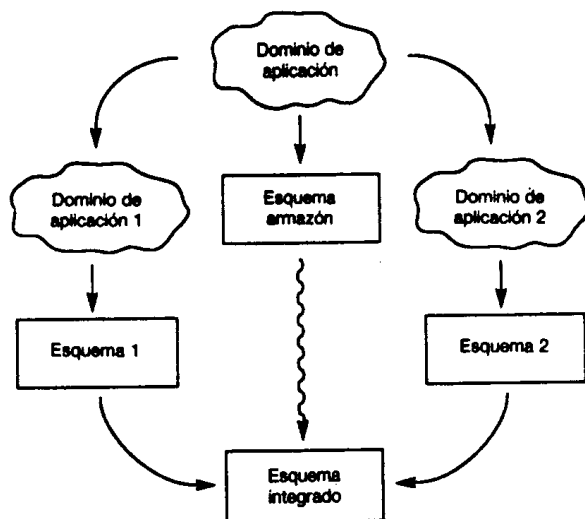
En esta estrategia el orden de los refinamientos es disciplinado, al igual que en el enfoque descendente. Sin embargo, los niveles de abstracción de los conceptos introducidos en versiones sucesivas del esquema son similares; de este modo se pierde la ventaja de trabajar en un nivel de abstracción para luego, una vez completado el diseño en el mismo, pasar a otro inferior.

Por otro lado, tiene la desventaja de que es necesario revisar una y otra vez toda la especificación de requisitos, para buscar conceptos todavía no representados y describirlos con detalle en el esquema.

#### 4. Estrategia mixta (*mixed*)

La estrategia mixta aprovecha tanto la estrategia descendente como la ascendente, pues i) los requisitos se descomponen según una estrategia ascendente, ii) se diseña un subesquema correspondiente a cada partición, siguiendo una estrategia ascendente o descendente, y finalmente se combinan los diferentes subesquemas.

La idea principal de este enfoque se muestra en la figura siguiente: cuando el dominio de aplicación es muy complejo, el diseñador divide los requisitos en subconjuntos (aunque no tanto como en la estrategia ascendente), que se consideran por separado.



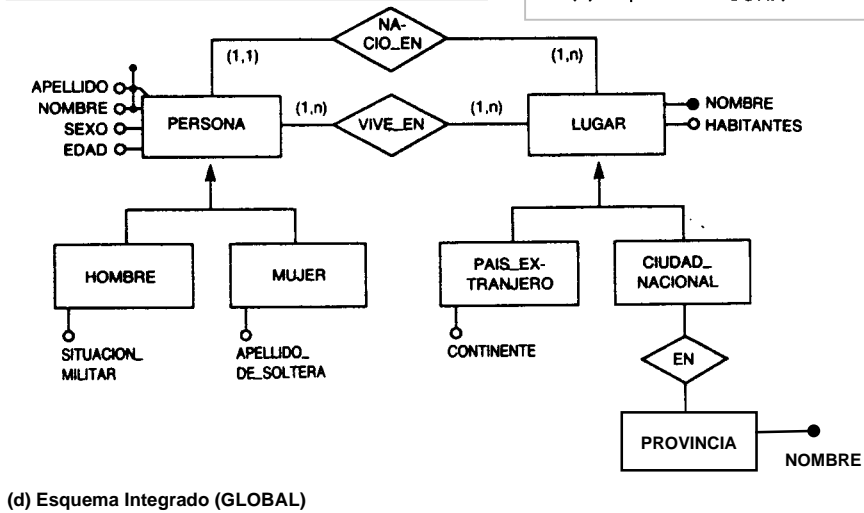
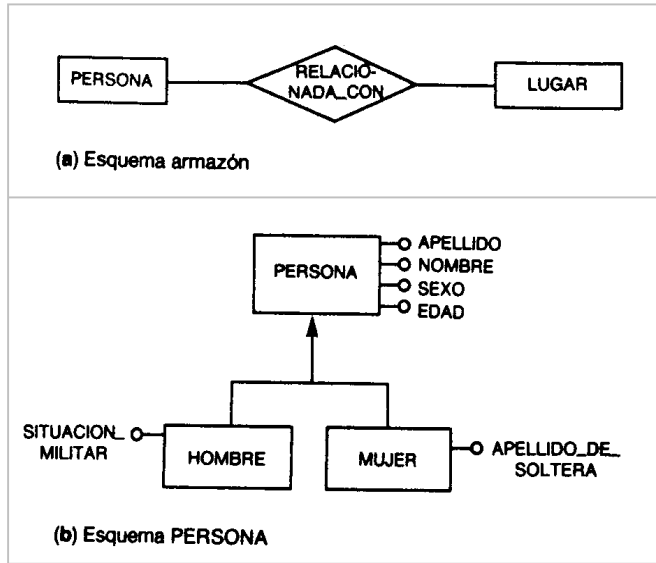
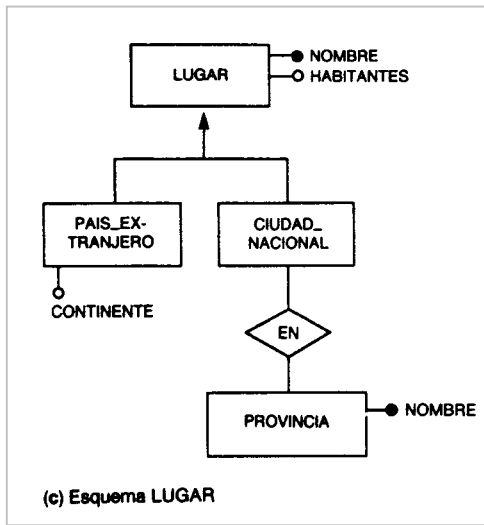
Al mismo tiempo que lleva a cabo la descomposición de las especificación de requisitos, el diseñador define un *esquema armazón o esqueleto*, que contiene los conceptos más importantes del dominio de aplicación y cómo se relacionan entre sí: proporciona una visión unificada y sintética del diseño completo, fijando los vínculos entre las particiones. El tiempo adicional requerido para realizar este paso es bien recompensado después, ya que la existencia del esquema armazón facilita la integración de los diferentes esquemas producidos.

Veamos ahora cómo abordar el diseño de la base de datos demográfica siguiendo un enfoque mixto. Los requisitos se consideran divididos en dos partes, referentes a personas y lugares.

Antes de empezar el diseño por separado de los dos esquemas, se pueden identificar con facilidad los conceptos *persona* y *lugar* como aspirantes a tipos de entidades del esquema armazón, unidos por la interrelación RELACIONADO\_CON.

Luego, se aborda el diseño de los dos subesquemas para las particiones identificadas (información referente a personas y a lugares).

Una vez terminado el diseño de ambos subesquemas, se conectan al esquema armazón (combinación). Para terminar, se refina la interrelación RELACIONADO\_CON para dar las interrelaciones NACIO\_EN y VIVE\_EN y se añaden las cardinalidades y los identificadores de las entidades.



## Comentarios acerca de las diferentes estrategias

Este apartado ha presentado cuatro estrategias alternativas y complementarias para el diseño progresivo de esquemas conceptuales.

La primera pregunta que surge es la siguiente: **A partir de ciertos requisitos de datos ¿las cuatro estrategias conducen siempre al mismo esquema final?**

La respuesta es claramente negativa, ya que cada estrategia sigue una filosofía de diseño específica. De hecho, incluso adoptando la misma estrategia, es posible modelar los mismos requisitos de maneras muy diferentes. Como consecuencia, es importante efectuar periódicamente una revisión de la calidad del esquema, e incluso por parte de varios diseñadores diferentes.

Una segunda pregunta sería **¿cómo saber qué estrategias se deben aplicar de forma práctica en una determinada organización y entorno?** El diseñador debe emplear la estrategia de diseño más conveniente con respecto a la organización y entorno específico. Por ejemplo, la estrategia descendente suele ser la más conveniente en el caso de organizaciones altamente estructuradas, donde los ejecutivos de alto nivel tienen una visión completa del dominio de aplicación, en un nivel de abstracción elevado. Por otro lado, una estrategia ascendente puede ser conveniente en organizaciones informales y con bajo nivel de estructuración, donde es más fácil modelar la información en detalle y luego integrarla, que crear una abstracción inicial de toda la base de datos y luego refinarla.

Por otro lado, una estrategia mixta es conveniente cuando es necesario comenzar el proceso de diseño antes de que todos los requisitos de datos estén disponibles, o bien si los detalles de la información que el diseñador ya conoce puede variar.

Se recomienda utilizar la estrategia descendente siempre que sea posible. Si no es así, se debería adoptar una mezcla de estrategias: las que resulten más naturales en el contexto específico en el que se trabajará la aplicación que se pretende desarrollar. Un diseñador experimentado puede usar una mezcla de las estrategias; por ejemplo, el diseñador puede elegir proceder de forma descendente, porque de esta forma el proceso de diseño es bastante estructurado y disciplinado; pero sin embargo, el diseñador podría omitir algunos conceptos, que se añadirían después (normalmente aplicando primitivas ascendentes) en una sesión de diseño posterior, en la que abordara el diseño con mayor detalle. En otras palabras, un diseño de ese estilo se conduciría principalmente de forma descendente, con algunas aplicaciones excepcionales de primitivas ascendentes.

En la mayor parte de los casos prácticos, según se indica en [ACPT99], la estrategia mixta es la única que puede ser realmente adoptada por los equipos de diseño de esquemas conceptuales de sistemas de información.