

FUNDAMENTOS Y DISEÑO DE BASES DE DATOS (2010/2011)

ÍNDICE

TEMA 1. INTRODUCCIÓN A LA TECNOLOGÍA DE BASES DE DATOS	4
1.1. EL PAPEL DE LAS BASES DE DATOS EN LOS SISTEMAS DE INFORMACIÓN	4
CONCEPTO DE SISTEMA DE INFORMACIÓN	4
COMPOSICIÓN DE UN SISTEMA DE INFORMACIÓN	4
CLASIFICACIÓN DE LOS SISTEMAS DE INFORMACIÓN RESPECTO A LA ESTRUCTURA DE LOS DATOS	5
TIPOS DE SISTEMAS DE INFORMACIÓN EMPRESARIALES	5
1.2. CONCEPTOS Y OBJETIVOS DE LAS BASES DE DATOS	6
EVOLUCIÓN DE LOS SISTEMAS DE GESTIÓN DE DATOS: DE LOS SISTEMAS ORIENTADOS A PROCESOS A LOS ORIENTADOS A DATOS.	6
VENTAJAS DE CONSTRUIR UNA BUENA BASE DE DATOS	7
CARACTERÍSTICAS DE LAS BASES DE DATOS	8
NIVELES DE ABSTRACCIÓN DE UNA BASE DE DATOS	8
1.3. SISTEMA GESTOR DE BASES DE DATOS (SGBD)	9
EL SGBD COMO INTERFAZ ENTRE USUARIO Y BASE DE DATOS	9
QUÉ ES UN SGBD	10
FUNCIONES DE UN SGBD: DE DEFINICIÓN O DESCRIPCIÓN, DE MANIPULACIÓN Y UTILIZACIÓN O CONTROL.	10
Función de definición o descripción.	11
Función de manipulación	11
Función de utilización o control	12
LENGUAJES DE LOS SGBD	12
TEMA 2. METODOLOGÍA DE DESARROLLO PARA BASES DE DATOS	14
2.1. DEFINICIÓN DE METODOLOGÍA	14
2.2. DEFINICIÓN DE MODELADO	14
2.3. MODELOS DE DATOS	14
2.4. APLICACIÓN DE UNA METODOLOGÍA	15
2.5. PRINCIPALES FASES DE LA METODOLOGÍA DE DESARROLLO DE BASES DE DATOS	16
TEMA 3. MODELO ENTIDAD INTERRELACIÓN (ER)	18
3.1. ELEMENTOS BÁSICOS DEL MODELO E/R	19
ENTIDAD	19
INTERRELACIONES	19
ATRIBUTOS	21
DOMINIOS	21
REPRESENTACIÓN DEL ATRIBUTO/DOMINIO	21
ESTÁTICA DEL MODELO E/R	22
3.2. EXTENSIONES DEL MODELO E/R	22
Dependencia en existencia:	22
Dependencia en identificación:	23
Atributos multivaluados y compuestos	25
Generalización y herencia	25
¡Lo que tienes que tener claro!	26
Ocurrencia	26

3.3.CONSTRUIR UN DIAGRAMA E/R	27
IDENTIFICADOR DE LOS CONSTRUCTORES	27
ELECCIÓN ENTRE VARIOS CONSTRUCTORES	27
Entidades/atributos	27
Entidades/atributos multivaluados	27
Atributos/Interrelaciones	28
PASOS EN LA REALIZACIÓN DEL DIAGRAMA E/R:	28
 TEMA 4. MODELO RELACIONAL	 30
 4.1. ESTRUCTURA DEL MODELO RELACIONAL	 30
DOMINIOS Y ATRIBUTOS	31
INTENCIÓN DE UNA RELACIÓN	31
EXTENSIÓN DE UNA RELACIÓN	31
CLASES DE RELACIÓN	31
CLAVES	32
4.2. RESTRICCIONES	32
RESTRICCIONES INHERENTES	32
RESTRICCIONES SEMÁNTICAS	33
REPRESENTACIÓN DE LA INTEGRIDAD REFERENCIAL	33
TERMINOLOGÍA DE LA INTEGRIDAD REFERENCIAL	34
INTEGRIDAD REFERENCIAL: CONSECUENCIAS DE LAS OPERACIONES DE BORRADO/MODIFICACIÓN	34
RESTRICCIONES SEMÁNTICAS	35
Rechazo	35
Disparadores (Triggers)	35
4.3 GRADO RELACIONAL	36
 TEMA 5. TRANSFORMACIÓN DEL MODELO E/R AL MODELO RELACIONAL	 37
 5.1. TRANSFORMACIÓN DE ENTIDADES, DOMINIOS Y ATRIBUTOS	 37
5.2. TRANSFORMACIÓN DE INTERRELACIONES N:M	38
TRANSFORMACIÓN DE RELACIONES N:M	38
TRANSFORMACIÓN DE RELACIONES 1:N	38
5.3 ESTUDIO DE LA CARDINALIDAD MÍNIMA	39
INTERRELACIONES 1:N – LADO 1	39
INTERRELACIONES 1:N – LADO N	39
5.4 TRANSFORMACIÓN DE INTERRELACIONES 1:1	39
5.5 TRANSFORMACIÓN DE INTERRELACIONES REFLEXIVAS	40
5.6 TRANSFORMACIÓN DE INTERRELACIONES DE GRADO MAYOR QUE 2	41
5.7 TRANSFORMACIÓN DE DEPENDENCIAS	41
5.8 TRANSFORMACIÓN DE JERARQUÍAS	42
ESTRATEGIA B	43
ESTRATEGIA C	43

TEMA 1. INTRODUCCIÓN A LA TECNOLOGÍA DE BASES DE DATOS

1.1. EL PAPEL DE LAS BASES DE DATOS EN LOS SISTEMAS DE INFORMACIÓN

CONCEPTO DE SISTEMA DE INFORMACIÓN

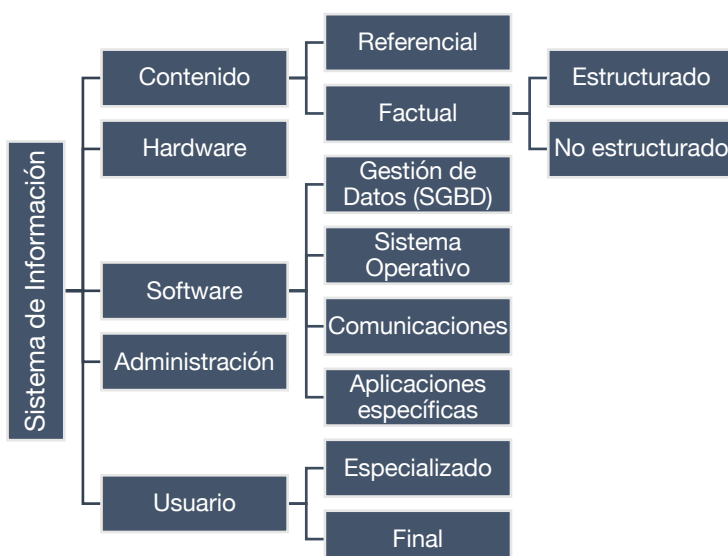
Un Sistema de Información es un conjunto de elementos ordenadamente relacionados entre sí que aporta a la organización información relevante y necesaria para su correcto funcionamiento; un SI eficaz debe recoger, procesar y almacenar datos de procedencia interna y externa de forma correcta y coherente.

Las características de un SI pueden agruparse en:

- **Tecnológicas:** son las que afectan al rendimiento y seguridad del sistema desde el punto de vista del equipo informático.
- **Funcionales y semánticas:** hacen referencia a lo que el sistema hace, a la corrección de los procesos que sigue y a su adaptabilidad a un entorno de datos cambiante.
- **Económicas:** ponen el énfasis en el coste del sistema y en la eficacia con que responde a los objetivos que debe cumplir el sistema.
- **Sociales:** son las características que afectan y tienen un impacto sobre el entorno social (interno o externo) en el que se desenvuelve el sistema.

COMPOSICIÓN DE UN SISTEMA DE INFORMACIÓN

- **Contenido:** conjunto de datos con su correspondiente descripción.
 - **Referencial:** indexa las referencias bibliográficas a los documentos que contienen los datos, pero no tienen los datos.
 - **Factual:** integra los datos en la estructura interna y puede buscarse de forma directa sin tener que acudir a otras fuentes de información. En estos sistemas la información puede estar estructurada o desestructurada, en las primeros los campos tienen un formato fijo y la entrada de los mismos está restringida a él; en los sistemas facutales desestructurados, la información contenida admite diferentes formatos (documentos, voz, imagen...).



▪ **Hardware:** Son los equipos físicos que contienen la información del sistema.

▪ **Software:** Incluye los programas elegidos para implementar el sistema y la base de datos, la documentación de uso, etc.; dentro de esta categoría es donde se encuentra el interés y desarrollo de los SGBD.

▪ **Administración:** son los sistemas implementados de optimización por parte de la organización para garantizar que los datos contenidos en sus SI sean de calidad y así poder tener una buena conversión de los datos a información.

- **Usuarios:** Son los principales interesados en la base de datos, podemos diferenciar entre: el especializado o desarrollador, que es aquel usuario que se dedica a crear, implementar y mantener aplicaciones y programas a medida para satisfacer las demandas de información de los usuarios finales. Un usuario ordinario será quien demande estos programas y los usará como herramienta de apoyo a su trabajo.

CLASIFICACIÓN DE LOS SISTEMAS DE INFORMACIÓN RESPECTO A LA ESTRUCTURA DE LOS DATOS

Los Sistemas de Información tratan de simplificar y buscar soluciones a los problemas que un usuario puede tener a la hora de trabajar con bloques de información; si nos centramos en estructura que tienen los Sistemas de Información con dos grandes grupos:

- **Sistemas de Recuperación de Información (SRI):** se dedican al tratamiento de datos no formateados (documentos) y a su extracción para conseguir un correcto análisis de los mismos.
- **Sistemas de Gestión de Bases de Datos (SGBD):** se dedican al tratamiento de los datos ya formateados (aquellos que ya han pasado por un proceso de limpieza y adaptación a la base de datos).

En la actualidad (2011), existe una clara tendencia hacia la convergencia de estos dos tipos de tratamiento y hay un solapamiento de los procesos en el que es un mismo sistema el que realiza ambas operaciones.

TIPOS DE SISTEMAS DE INFORMACIÓN EMPRESARIALES

Los Sistemas de Información son un recurso cada vez más imprescindibles para las organizaciones y empresas, que comienzan a manejar un gran volumen de datos y a necesitar la implementación de sistemas de tratamiento de la misma. Si nos enfocamos en la utilidad que un SI puede tener a nivel empresarial, vemos que hay dos puntos clave en los que una buena gestión de datos puede hacer mejorar a la empresa:

- **SI para la gestión:** el tratamiento administrativo de los datos operacionales es imprescindible para el control no solo interno de la empresa, también con su relación con el entorno, trabajadores y clientes; y se vuelven necesarios para conseguir una rutina de la organización dinámica.

Los Sistemas de Información para la Gestión suelen atender a las tareas rutinarias de la empresa, es decir, afectan a aquellas operaciones que componen el funcionamiento normal de la empresa. En este tipo de sistemas, se reproducen de forma automática todas o parte de las operaciones que tradicionalmente se realizaban de forma manual.

- **SI para la toma de decisiones:** buscan sistematizar las funciones directivas y los datos a los que la dirección recurre para tomar decisiones; así mismo, es ineludible que estos hayan seguido un proceso correcto de curado, tratamiento y exposición, por tanto, es imprescindible que los datos utilizados para la toma de decisiones sean de calidad.

Los Sistemas de Información para la toma de decisiones están enfocados a cómo sirven y a convertirse en herramientas de apoyo para que los directivos apliquen de forma estratégica la información y el conocimiento que obtienen a través del análisis de datos.

Actualmente (2011), se tiende a mantener un único repositorio para la información que se encarga de alimentar a los dos tipos de sistemas. Incluso existen SGBD que pueden atender las demandas de los dos tipos de S.

1.2. CONCEPTOS Y OBJETIVOS DE LAS BASES DE DATOS

EVOLUCIÓN DE LOS SISTEMAS DE GESTIÓN DE DATOS: DE LOS SISTEMAS ORIENTADOS A PROCESOS A LOS ORIENTADOS A DATOS.

Los sistemas tradicionales enfocaban el trabajo hacia los procesos que se realizaban dentro de la organización, haciendo que fuesen los procesos los que tuviesen un tratamiento y refuerzo mayor al que la organización aplicaba a los datos. En este **Sistema orientado al Proceso** los datos se encuentran repetidos en diversos ficheros tratados de forma independiente por cada aplicación, lo que presenta una clara desventaja por la duplicidad y baja productividad de los datos; además, de la pérdida de relaciones ricas de los datos.

En los sistemas orientados hacia el proceso, las aplicaciones realizan los procesos de análisis de forma independiente y los datos no se cruzan ni cotejan entre ellas lo que aumenta el riesgo de tener menor calidad y fiabilidad en los datos si la actualización de estos no se hace correctamente.

Si una empresa tiene sus SI orientados a los procesos, deberá tener en cuenta los inconvenientes más importantes y buscar una forma conveniente de reducir el impacto negativo que pueden tener para la calidad de la información:

- Desperdicio de la memoria secundaria de los equipos al tener redundancia de datos.
- Las labores de indexación, introducción y actualización de datos requieren más tiempo.
- Los problemas de inconsistencia aumentan exponencialmente cuantas más recurrencias de bases de datos existan y reiteraciones en los datos, sobre todo, si estos no se actualizan en todas las bases de datos de forma correcta (ej. despistes y olvidos).
- Son Sistemas poco adaptables a nuevos entornos y requerimientos.
- Hay una gran dependencia físico-lógica en el sentido de que al modificar los datos normalmente se deben modificar los programas.

Ejemplo

La dirección de la empresa Promoción S.A. decide realizar una selección de personal entre sus empleados con efectos de promoción interna. Para ello decide crear una aplicación de “Selección” y un fichero con los siguientes campos: DNI, nombre, dirección, puesto, fecha de contratación, formación, especialidad, nivel de conocimiento...

En una reunión posterior decide informatizar la “Nomina”.

En todos los casos que puedan valorar, el tratamiento inicial de la información se ha planteado a corto plazo y orientado a solucionar un problema puntual, por lo que no tiene en cuenta la utilización potencial del sistema en el futuro; esto hace que la herramienta de “Selección” creada no cumpla con las expectativas o no tenga todas las funciones necesarias para contener el sistema que requeriría la gestión automática de la nómina de los trabajadores.

La empresa, en esencia, puede optar por tres soluciones: Crear una nueva aplicación de nómina independiente, con sus referencias independientes a las de selección; puede modificar el fichero de selección, o puede crear una nueva aplicación para las nóminas pero que la información esté enlazada al fichero de selección.

Los principales problemas que puede encontrarse en estos casos son:

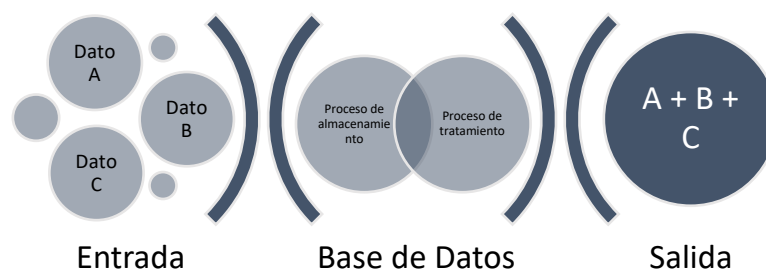
- La dificultad de mantenimiento de la estructura
- La excesiva dependencia de un soporte físico
- La dependencia de los datos respecto a los programas
- Dificultad para montar SI orientados a la toma de decisiones, dado que los ficheros solo están orientados a cumplir con una función operativa.

El mantenimiento de los SI orientados a Operaciones es mucho más complicado porque la información está reiterada y cualquier modificación de los datos exige un trabajo de repetición en todos los lugares donde aparece el dato específico. No mantener estos ficheros actualizados llevaría a una incoherencia de la información y un descontrol, pues probablemente no sabrías cuál es el correcto y suele estar enlazado con una falta de control por parte de la empresa de los momentos de actualización de datos.

Además, al estar íntimamente ligados a las aplicaciones, cualquier cambio en la estructura física obliga a modificar las aplicaciones para que se adapten a estos cambios. Estos Sistemas tienen su utilidad centrada casi exclusivamente en cubrir las necesidades de la organización en contextos muy específicos y no buscan un análisis de la información rico; por lo que es un sistema en desuso y fuera de ese entorno, inoperantes.

Los **Sistemas orientados a los datos** surgen como una necesidad para resolver los problemas intrínsecos del sistema tradicional. En estos la información se trabaja y optimiza para evitar la duplicidad de datos y, por lo tanto, se crea una base de datos que engulle al sistema de ficheros y los automatiza en una única colección de datos que se relacionan eficazmente y son entendidos por todos los procesos que lo necesitan con el menor número de interacciones posible.

La solución a los problemas del enfoque tradicional conduce a un cambio radical en el enfoque del SI. Ahora los datos no se diseñan pensando en la aplicación concreta, sino que tratan de satisfacer unas necesidades de información a nivel de organización.



VENTAJAS DE CONSTRUIR UNA BUENA BASE DE DATOS

- **Independencia de los datos respecto a los tratamientos y viceversa:** Supone que un cambio en los tratamientos no obligue a cambiar el diseño de la Base de Datos. Por otro lado, la alteración en la forma de guardar los datos o los cambios de acceso a los mismos no debe suponer cambios en el tratamiento. En la práctica, esta independencia nunca es absoluta, pero los SGBD intentan mejorar continuamente este aspecto.
- **Coherencia de los resultados:** Como la información se recoge y almacena una sola vez, las aplicaciones utilizan el mismo conjunto de datos por lo que cualquier cambio es registrado en un único sitio y no es necesario ocuparse de modificar las diferentes versiones del dato.
- **Mayor disponibilidad de la información en entornos multiusuario:** Como el conjunto de datos es común para todos, no hay problemas de “propiedad” de los datos. Todo el mundo puede ver la misma información.
- **Reducción del espacio de almacenamiento:** La desaparición de las redundancias lleva consigo la disminución del espacio necesario de almacenamiento.
- **Información integrada con los datos:** En el enfoque clásico, los datos se encuentran definidos semánticamente en los programas que los tratan. Existe, por tanto, una disociación de datos y

contenido. Los sistemas orientados a datos tratan de paliar este problema incluyendo la descripción de los datos junto con su contenido en la propia base de datos.

- **Mayor eficiencia en la recogida de datos:** Al no existir redundancia lógica (puede que haya redundancia física por motivos de eficiencia), los datos se recogen una sola vez y, por tanto, los controles sobre su calidad se realizan también solo una vez.
- **Reducción del tiempo en los procesos de entrada, actualización y salida de datos.**

CARACTERÍSTICAS DE LAS BASES DE DATOS

Una de las primeras características que encontramos cuando pensamos en una base de datos es en la que se inspira su creación por la necesidad de un correcto trato de datos. Por tanto, podemos decir que una base de datos es una colección o depósito de datos que están integrados, almacenados en un soporte secundario (no volátil), que tiene las redundancias controladas y es compartida y utilizada por diferentes usuarios y aplicaciones independientes.

La definición de los datos (Estructura de DB) única y almacenada junto con los datos se ha de apoyar en un modelo de datos previamente ajustado a las necesidades de gestión, organización y al tipo de datos que requiere y trabaja la empresa.

Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad (integridad, disponibilidad y confidencialidad) del conjunto de datos.

Los datos deben estar estructurados y relacionados de forma que recojan la mayor cantidad semántica posible del mundo que tratan de representar. Se tiende a almacenar juntos los datos, sus relaciones y sus restricciones semánticas.

La redundancia debe ser solo física; no debería existir redundancia lógica. Esto quiere decir que es posible que una información aparezca duplicada a nivel físico, puede estar en varios discos o porciones de disco, pero no deberá crearse un modelo lógico de los datos que incluya información duplicada.

Las bases de datos deben atender a múltiples usuarios y ser accesible por diferentes aplicaciones.

La descripción de los datos (lo que se denomina estructura o esquema de la base de datos) debe guardarse con los mismos datos y ser única. En los sistemas de ficheros, los datos estaban en un sitio y su descripción se encontraba en los programas.

El propio SBD incluye procedimientos para recuperar y actualizar la información.

NIVELES DE ABSTRACCIÓN DE UNA BASE DE DATOS

En los Sistemas de Información tradicionales existían dos estructuras diferenciadas:

- Lógica: correspondiendo a la visión del usuario.
- Física: siendo la forma en que se encuentran los datos almacenados.

Con la aparición de las bases de datos aparece un nuevo nivel: el conceptual. Este nivel supone una división del antiguo nivel lógico en dos: el lógico externo (o de usuario) y el lógico global (el que hemos nombrado como conceptual).

La división en niveles de abstracción de una base de datos trata de recoger las distintas perspectivas o visiones que tienen de la base de datos los diferentes usuarios y la propia máquina. A los usuarios en

general les corresponde el nivel externo; este nivel da lugar a un esquema externo (estructura de los datos desde el punto de vista de cada usuario concreto). Desde el punto de vista de la empresa, la información tiene una estructura global que correspondería a nivel conceptual (a la concepción, estructura de los datos desde el punto de vista del diseñador de la información de la empresa atendiendo a las necesidades del nivel externo). Por último, el software de la máquina, que es el centro del SGBD, tiene también una forma particular de ver la información que se denomina estructura interna que correspondería al nivel interno. La estructuración en tres niveles trata de conseguir la independencia entre datos y aplicaciones.

Estructura lógica de usuario: **esquema externo**.

- Corresponde a la visión que tiene de la base de datos cada usuario particular.
- En ella se reflejan todas las relaciones y datos que tengan interés para cada usuario particular, existen tantos esquemas externos como aplicaciones, teniendo en cuenta que el mismo esquema externo lo pueden usar varias aplicaciones.

Estructura lógica global: **esquema conceptual**.

- Corresponde al enfoque del conjunto de la empresa
- Influye la descripción de todos los datos, así como las interrelaciones entre ellos y sus restricciones de integridad.
- Se interpone entre los niveles anteriores (lógica de usuario e interno) y pretende ser una representación global de los datos, siendo independientes tanto del equipo como de cada usuario particular.

Estructura física: **esquema interno**.

- El nivel interno corresponde a la visión de los datos desde la perspectiva del software de base de datos. Por este motivo es muy diferente según el gestor de base de datos con el que se trabaje, aunque los conceptos manejados son muy similares en todos ellos.
- Hace referencia a la forma en que se organizan los datos en el almacenamiento físico.
- Depende mucho de cada SGBD, aunque suele referirse a:
 - Estrategia de almacenamiento: asignación del espacio de almacenamiento para los datos.
 - Caminos de acceso: forma de acceder a la información (claves, índices, punteros...)

1.3. SISTEMA GESTOR DE BASES DE DATOS (SGBD)

EL SGBD COMO INTERFAZ ENTRE USUARIO Y BASE DE DATOS

El SI se compone de 3 subsistemas: **operacional, táctico y estratégico**. Por tanto, lo deseable es que la base de datos se diseñe con la idea de integrar la información de estos 3 subsistemas y que atienda a las necesidades de los usuarios en los 3 niveles. El SGBD aparece entonces como el elemento que permite a los usuarios trabajar con la base de datos.

Para comenzar con una base de datos hay que entender quiénes son los que interactuarán con ella, estos son los usuarios:

- **Especializado:** son los informáticos encargados de la creación y mantenimiento de la base de datos, los programas para usuarios finales. Encontramos 3 perfiles mayoritariamente:
 - Diseñadores: se encargan de estructurar la base de datos (lógica y física). Deciden qué es lo que se guarda en ella, por lo que deben hablar con todos los usuarios que interactuarán con ella para averiguar qué información se debe almacenar y, lo que es más importante, decidir cuál es la mejor estructura para reflejar esa información. En ocasiones se diferencia entre diseñadores lógicos (más cercanos al punto de vista del usuario final) y diseñadores físicos (más cercanos al punto de vista de la máquina).
 - Administradores: vigilancia y gestión de datos. La figura del administrador coincide muchas veces con la del diseñador físico. Son los encargados de gestionar los datos (controlando el acceso a los mismos y garantizando su disponibilidad) y también de realizar las actividades necesarias para que el SGBD funcione en todo momento de la forma más efectiva posible.
 - Analistas, programadores y desarrolladores: procedimiento y programas para facilitar el trabajo a los usuarios finales. En este grupo se incluyen los jefes de proyecto, analistas, analistas-programadores y programadores; a fin de cuentas, todo el personal informático encargado de realizar la aplicación que facilitará el flujo eficiente y correcto de la información.
- **Final:** se consideran como tales aquellos que utilizan la base de datos como herramienta para el desarrollo de su trabajo. Suelen estar interesados en el contenido más que en la estructura. Se pueden agrupar en dos categorías genéricas:
 - Habituales: consultas y actualizaciones de la base con programas predefinidos. Trabajan de forma regular con la base de datos, por lo que suelen utilizar aplicaciones informáticas diseñadas especialmente para ayudarlos en su trabajo.
 - Esporádicos: Herramientas sencillas porque el uso no es habitual. No trabajan de forma habitual con la base de datos, pero suelen tener unas exigencias muy concretas por lo que requieren de un tipo de programa muy específico. En ocasiones utilizan herramientas sencillas propias del SGBD (por ejemplo, SQL).

QUÉ ES UN SGBD

Conjunto de programas, procedimientos y lenguajes que suministran a los distintos tipos de usuarios los medios necesarios para describir y manipular los datos almacenados en la base, garantizando su seguridad.

Si se tiene en cuenta que en una base de datos existe una gran variedad de usuarios y aplicaciones es imprescindible dotar al sistema de la suficiente flexibilidad para atenderlos a todos.

Se suele denominar Sistema de Base de Datos al conjunto formado por la base de datos, el sistema de gestión de datos y a cualquier otro procedimiento humano o automático que garantice la disponibilidad de la información.

FUNCIONES DE UN SGBD: DE DEFINICIÓN O DESCRIPCIÓN, DE MANIPULACIÓN Y UTILIZACIÓN O CONTROL.

Las operaciones habituales sobre una base de datos son:

Sobre el conjunto de datos	Creación	Las operaciones habituales que se realizan sobre una base de datos pertenecen a dos categorías:	
	Reestructuración		
	Consulta a la totalidad		
Sobre registros concretos	Actualización	Inserción	<ul style="list-style-type: none"> Las que afectan al conjunto de las bases de datos: se refieren a las operaciones que sirven principalmente para crear y modificar la estructura de la base de datos. Las que afectan a los registros o filas concretas: son las operaciones diarias habituales e incluyen las consultas, las modificaciones, las inserciones y los borrados de información. Estas operaciones se hacen a nivel de los registros almacenados en la base de datos. En una inserción lo que se hace es añadir nuevos registros.
		Borrado	
		Modificación	
	Consulta selectiva		

Función de definición o descripción.

Los Lenguajes de Definición de Datos (LDD) describen los elementos de datos, la estructura, sus interrelaciones y las reglas de integridad para los tres niveles estructurales: lógica de usuario, lógica global y física.

La función de descripción se lleva a cabo mediante el Lenguaje de Definición de Datos. El LDD debe permitir definir las tres estructuras de datos (externa, conceptual e interna).

- A nivel físico (esquema interno) se debe poder indicar el espacio reservado para la base, la longitud de los campos, su modo de representación, caminos de acceso, etc.
- A nivel de lógica de usuario (esquema externo) y lógica global (esquema conceptual) se deben proporcionar instrumentos que permitan definir los objetos, las relaciones entre ellos y las reglas de integridad. Las descripciones del esquema externo deben referirse lógicamente, al esquema conceptual.

El SGBD también proporciona las funciones de correspondencias entre las estructuras lógicas de usuario y la estructura lógica-global y entre esta y la estructura física.

Función de manipulación

Las operaciones sobre los datos de la base se realizan mediante Lenguaje de Manipulación de Datos (LMD) que consisten en: recuperar, añadir, suprimir y modificar; para ello es necesario definir la estructura lógica a manipular, definir el criterio de selección y acceder a la estructura física.

Una vez que se han descrito los datos, los usuarios tienen la necesidad de recuperar la información o de actualizarla. Esto se lleva a cabo gracias al LMD.

La consulta de la base de datos puede ser de dos tipos:

- Totalidad de los datos: se recuperan todos los datos de un tipo determinado (por ejemplo, todos los libros del catálogo de una biblioteca).

- Consulta selectiva: se basa en un criterio de selección que se empleará para localizar los registros que cumplan una condición (por ejemplo, todos los libros sobre historia del arte).

En ambos deberá indicarse la estructura lógica externa que se desea recuperar. Por ejemplo, título, autor y editorial para la consulta total y título, tema y ubicación para la consulta selectiva. Con esta información, el SGBD accede a la estructura física de la BD, localiza los registros adecuados y los devuelve al usuario.

La función de manipulación está dirigida a todos los usuarios, sean informáticos o no. El LMD, en ocasiones, se basa en un conjunto de comandos (lenguaje huésped) que se escribe en un lenguaje de programación (lenguaje anfitrión). También puede tratarse de un lenguaje autocontenido (SQL), que no precisa apoyarse en otros lenguajes y que suele usarse de forma interactiva; la mayoría de los SGBD atienden a los dos tipos de lenguaje: huéspedes y autocontenidos.

Función de utilización o control

En esta función se agrupan todas las interfaces de los usuarios y los procedimientos para el administrador.

Las exigencias en cuanto a la utilización de las bases de datos son muy diferentes según los distintos tipos de usuarios y la función de control se encarga de responder a todas ellas.

Tienen especial importancia los instrumentos que faciliten las tareas de administración (cambiar la capacidad de los ficheros, obtener estadísticas de utilización, gestión de copias de seguridad, arranque en caso de caída del sistema, etc.).

LENGUAJES DE LOS SGBD

Un SGBD debe disponer de lenguajes y procedimientos que posibiliten la interacción con la base de datos. Nos encontramos con:

- LDD (Lenguaje de Definición de Datos): para la definición de las estructuras de datos a todos los niveles (externo, conceptual e interno).
- LMD (Lenguaje de Manipulación de Datos): permite la ejecución de operaciones de recuperación y actualización sobre un conjunto de registros (identificado a través de un criterio de selección), indicando las estructuras externas sobre las que se actúa.
- Procedimientos de administración: copias de seguridad, estadística, carga de datos...

Los lenguajes de los SGBD deben tener un enfoque distinto que responda a los diferentes tipos de usuario de una base de datos. No son iguales las exigencias de un administrador que las de un usuario final.

Generalmente, los usuarios informáticos necesitan lenguajes potentes que les permiten administrar, definir y manipular la información. Suelen utilizar uno o varios lenguajes anfitriones en los que incluyen una serie de comandos (lenguaje huésped o lenguaje embebido) que les proporciona el SGBD para interactuar con la base de datos. Entre los lenguajes anfitriones más comunes se encuentran Cobol, Ensamblador, Fortran, Basic, C, C++, Java, etc. También suelen incluirse en esta categoría los Lenguajes de Cuarta Generación (4GL) que incluyen sentencias SQL.

Por su parte, el usuario final necesita medios simples para comunicarse con la base. Para ello, suele emplearse un lenguaje de manipulación autocontenido con una sintaxis sencilla pero suficientemente potente como para poder realizar consultas complejas.

En el caso de los SGBD relacionales, el lenguaje SQL es el estándar más extendido para proporcionar facilidades de definición y manipulación.

Los lenguajes de definición deben permitir describir la estructura de los datos en los tres niveles (externo, conceptual e interno). Realmente muchos SGBD tienen solo dos niveles de descripción de datos: uno para el esquema externo y otro para los esquemas conceptual e interno. Esta perspectiva tiene el inconveniente de que cualquier cambio físico puede obligar a redefinir la estructura lógica. En los SGBD relacionales a los esquemas externos se les denomina también vistas externas y disponen de facilidades para crearlas.

Los lenguajes autocontenidos se pueden utilizar de forma interactiva desde un terminal. Se llaman también lenguajes nativos y permiten realizar funciones tanto de manipulación como de descripción.

A veces el mismo lenguaje autocontenido puede actuar como lenguaje huésped. Este es el caso de SQL que puede ser llamado desde un lenguaje anfitrión o bien puede interactuar directamente con la base de datos desde un terminal.

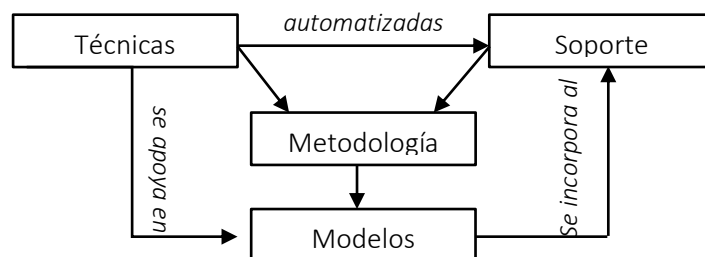
TEMA 2. METODOLOGÍA DE DESARROLLO PARA BASES DE DATOS

2.1. DEFINICIÓN DE METODOLOGÍA

Una **metodología** es un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de un producto software.

Y que describe las actividades a seguir en el desarrollo de principio a fin de la base de datos, qué es lo que hay que realizar en cada actividad, indicando qué se necesita como entrada, qué se produce como salidas y quién está involucrado (libro de recetas).

La relación entre los diferentes componentes de una metodología puede describirse de la siguiente manera:



2.2. DEFINICIÓN DE MODELADO

El **modelado** es la descripción de la estructura de la base de datos de forma que se represente fielmente la parcela del mundo real que se va a almacenar -> proceso de abstracción que se apoya en un MD.

La **abstracción** busca las propiedades comunes de un conjunto de objetos reduciendo la complejidad y ayudando a entender el mundo real. Existen diferentes tipos de abstracción según el modelo elegido.

- Clasificación | Particularización
- Agregación | Desagregación
- Generalización | Especialización
- Asociación | Disociación
- Jerarquía de abstracciones

2.3. MODELOS DE DATOS

Un **modelo de datos** proporciona un conjunto de conceptos, reglas y concesiones que nos permiten especificar y manipular los datos que queremos almacenar en la base de datos. Todo modelo de datos se compone de una parte estática y otra dinámica.

- **Estática:**
 - Conjunto de objetos
 - Conjunto de asociaciones entre ellos
 - Conjunto de restricciones
 - Inherentes (propias del modelo elegido)
 - Semánticas (propias del usuario)
- **Dinámica:**
 - Recuperación
 - Actualización

Si nos centramos en las restricciones del sistema, es decir, en aquellas limitaciones impuestas a la estructura del modelo o los datos que invalidan ciertas ocurrencias de la base de datos, nos encontramos que estas provienen fundamentalmente de dos focos muy particularizados:

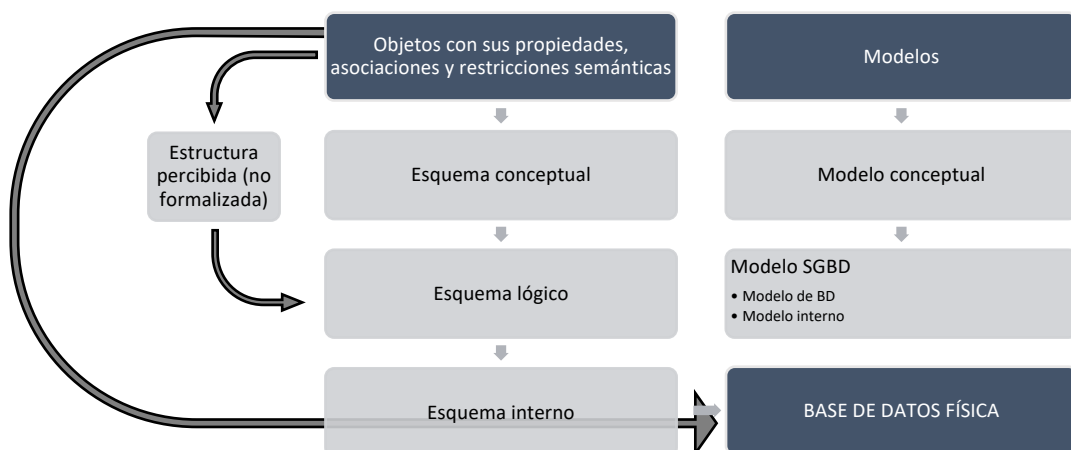
- El propio sistema, las restricciones **inherentes** que se dan por la estructura del modelo, y
- las **semánticas**, que son las limitaciones que se imponen a los valores de los atributos o a las características de las interrelaciones.

El control de la **semántica de los datos** está ligada al uso y control de la semántica en el mundo analógico (real), igualmente y de forma paralela, la semántica de datos ha ido evolucionando a lo largo del tiempo. Inicialmente, dicho control semántico no existía, salvo en la mente del usuario que era el encargado de ejecutar y valorar si los datos cumplían las restricciones semánticas o no que él mismo tenía; este control posteriormente se incluiría de forma orgánica a los programas de aplicación como parte de su código. Con la llegada de las bases de datos, dicho control está integrado junto con la descripción de los datos.

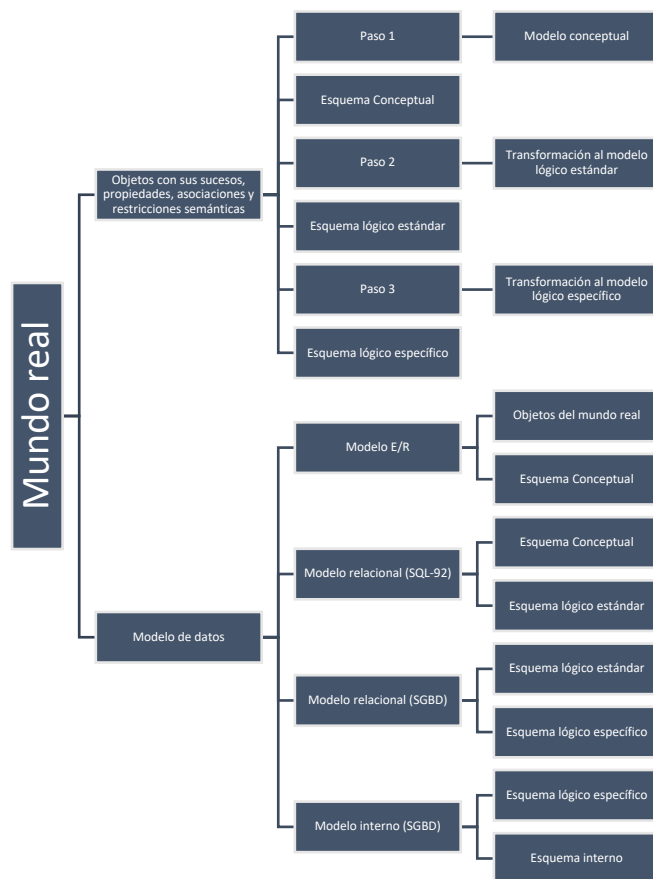
La **Integración Semántica en la base de datos** reporta una serie de ventajas que han hecho imprescindible su uso, así, podemos afirmar que las restricciones semánticas en la base de datos aportan:

- Un control único de los datos, por lo que se evitan las redundancias en la información. Además, se calcula que un gran porcentaje del código de una aplicación se podía afectar o perder por la falta de este tipo de controles.
- Se evitan los descuidos en los programas, por ejemplo, que el usuario se olvide de ejecutar de forma correcta una regla o introducir correctamente un valor; evita que el usuario deba conocer todas las reglas semánticas que el sistema utiliza.

2.4. APLICACIÓN DE UNA METODOLOGÍA



Como hemos visto, los modelos de datos son el instrumento que se aplica al mundo real para obtener un esquema.



El estado actual de la técnica recomienda comenzar el diseño aplicando un Análisis Conceptual que describa el problema (esquema conceptual) con el mayor contenido semántico posible y con independencia del SGBD en el que posteriormente se implemente la base de datos. Posteriormente se debería realizar un diseño lógico que proporcione la estructura lógica específica del SGBD que se utilizará (esquema lógico). A partir de este esquema lógico se obtiene un esquema interno que será implementado físicamente en la máquina. El esquema interno se obtiene a partir de Modelos Internos muy ligados al SGBD con el que se trabaja.

En ocasiones no se lleva a cabo el Análisis Conceptual y el diseñador realiza la abstracción del mundo real sin utilizar ningún formalismo o metodología concreta. A esto se le denomina estructura percibida y no es recomendable realizarla.

2.5. PRINCIPALES FASES DE LA METODOLOGÍA DE DESARROLLO DE BASES DE DATOS

Entrada	Fase de diseño	Herramientas	Proceso	Resultado
Reglas semánticas	Análisis o diseño conceptual	Modelo Entidad/Relación	Entidades Atributos Relaciones	Diagrama Entidad/Relación
Diagrama de Entidad/Relación	Diseño Lógico	Modelo Relacional	Reglas de Paso del modelo E/R al Relacional	Diagrama relacional
Diagrama Relacional	Diseño Físico	SGBD-R	Dar de alta las tablas en el SGBD-R	Base de datos física

Lo importante de esta metodología es que en cada fase se parte de unas determinadas entradas. A estas entradas se les aplica un tratamiento basado en la utilización de una herramienta determinada y da como resultado una salida.

En la fase inicial, el análisis conceptual. Se parte de las reglas semánticas establecidas por el problema del mundo real a resolver y aplicando como herramienta el modelo de entidad/relación, se obtiene un diagrama entidad/relación que representa ese problema a resolver.

En la segunda fase, el diseño lógico, se parte del diagrama entidad/relación creado en la fase inicial y se aplica como herramienta una serie de reglas o normas de transformación de diagramas E/R a un diagrama relacional. Como resultado se obtiene un diagrama o grafo relacional equivalente al diagrama

E/R y que también representa el problema a resolver. La diferencia con el primer paso es que este nuevo diagrama es reconocido por los SGBD.

En la última fase, el diseño físico, simplemente se traslada el diagrama relacional a las tablas en el SGBD comercial donde se vaya a implementar la base de datos de forma física.

Análisis Conceptual.

Su objetivo principal es obtener una representación formal de los datos necesarios para la empresa, sus objetivos pueden definirse como:

- La obtención del diagrama entidad/relación que representa la organización de los datos de modo gráfico.
- La verificación de la validez del diseño de los datos.
- Su utilidad como punto de partida para el *diseño lógico*.

Diseño lógico

Su principal objetivo es obtener una representación válida de los datos (recogidos del diagrama entidad/relación) más próximo a la representación física en el ordenador.

Su resultado debe ser un diagrama relacional.

Diseño físico

Es la representación de cómo va a estar configurada físicamente la base de datos. En este nivel se tienen en cuenta las estructuras de almacenamiento. Esta parte de la aplicación metodológica persigue:

- Representar a la organización física a través de la estructura de los datos de un modo gráfico.
- Servir como último modelo para la creación y modificación de ficheros que posteriormente se convertirán en la base de datos.

Es habitual encontrar que se toma como referencia de entrada al diseño físico el diagrama relacional que resulta del diseño lógico.

TEMA 3. MODELO ENTIDAD INTERRELACIÓN (ER)

El modelo entidad/interrelación (M E/R) fue propuesto por Peter P. Chen en 1976. Este modelo se presenta como una vista unificada de los datos, centrada en su estructura lógica y abstracta, como representación del mundo real y con independencia de cuestiones de tipo físico.

Como su nombre indica, el modelo E/R se basa en entidades que se interrelacionan o asocian entre sí. Una de las principales dificultades del diseño de una base de datos radica en que se concibe pensando en el modelo de datos con el que va a trabajar la máquina. El problema de esta perspectiva es que se pierde la visión conceptual del problema al estar el diseño contaminado por cuestiones de implementación.

La solución a este inconveniente está en concebir la base de datos a un nivel superior, abstrayéndose de condiciones relativas a la instrumentación (lógica o física) para trabajar con elementos relevantes al tema: entidades e interrelaciones.

Otro problema presente en muchos desarrollos informáticos era la poca atención dedicada a los datos en relación con los tratamientos. Sin embargo, se comprobó que la estructura de los datos suele ser más estable que los tratamientos por lo que se empezó a considerar un mayor rigor en el análisis y diseño de los datos.

Como respuesta a este tipo de problemas aparece el modelo E/R que ha tenido una gran difusión en la comunidad informática y ha sido objeto de importantes aportaciones que han enriquecido la versión original. Algunas de sus características son:

- Trata de aglutinar las ventajas de modelos anteriores como el modelo de red, el modelo jerárquico y el modelo relacional.
- Se adapta muy bien al nivel conceptual de la arquitectura ANSI/SPARC ya que considera los datos desde el punto de vista de sus propiedades lógicas, apartándose de consideraciones sobre el almacenamiento y la eficiencia de las estrategias de acceso.

El modelo E/R trata de representar los objetos y las relaciones del mundo real mediante un mecanismo de abstracción basado en una serie de símbolos, reglas y métodos que permiten representar el Universo del Discurso de forma gráfica. El modelo también aporta mayor contenido semántico sobre el Universo del Discurso que cualquiera de los modelos de datos anteriores (Relacional, jerárquico y red).

Inicialmente este modelo no consideraba la parte dinámica, definiendo únicamente la parte estructural y semántica. Al utilizarse como una herramienta de diseño conceptual y no estar soportado directamente por los SGBD, la parte dinámica carece de interés.

Algunos autores han completado el modelo proponiendo lenguajes de manipulación, como CLEAR y Cable.

Realmente, al ser E/R un modelo de análisis más que de implementación, no es directamente accesible por un SGBD, por lo que suele realizarse posteriormente una transformación de sus estructuras a una de las propias del SGBD en el que esté implementada la base de datos.

Inicialmente, además, solo consideraba entidades relacionales y atributos; posteriormente se añadieron conceptos como la semántica de las relaciones, la cardinalidad máxima y mínima, dependencias en

existencia e identificación, generalización, etc. Estas ampliaciones han dado lugar a una familia de modelos extendidos.

3.1. ELEMENTOS BÁSICOS DEL MODELO E/R

Como en todo modelo de datos, la componente estática permite definir la estructura de la base de datos. Dentro de los elementos permitidos de la parte estática del Modelo E/R se encuentran los siguientes:

- Entidades: representan objetos del universo del discurso.
- Interrelaciones: representan asociaciones entre objetos.
- Atributos: representan propiedades de los objetos.
- Dominios: representan el conjunto de valores que puede tomar cada atributo.

ENTIDAD

La entidad es un elemento que se define como un objeto, real o abstracto, de interés para la empresa y que necesita que este almacenado en la base de datos.

La entidad es el elemento que permite representar un objeto de la base de datos. Este objeto se caracteriza por tener una serie de propiedades que se describirán mediante atributos. Una entidad agrupa a todos los objetos del universo del discurso que son del mismo tipo, es decir, que tienen las mismas características.

Terminológicamente, podemos encontrarnos con:

- Tipo de entidad: referido a la estructura genérica (escritor, frutas...)
- Ocurrencia de la entidad: es cada una de las realizaciones concretas de este tipo de entidad (G. García Márquez, naranja...)

El tipo de entidad describe las características del objeto como resultado de una operación de abstracción. Se obtiene a partir de la semántica del problema, tras identificar una serie de características propias a determinados objetos de dicha semántica.

Según las clases, una entidad puede definirse como:

- Regular: cuando las ocurrencias de un tipo de entidad regular tienen existencia propia, es decir, existen por sí mismas.
- Débiles: cuando la existencia de cada ocurrencia de un tipo de entidad débil depende de la existencia de la ocurrencia del tipo de entidad regular de la cual depende; si se elimina una ocurrencia del tipo de entidad regular, desaparecerá con ella toda ocurrencia del tipo de entidad débil que tenía relacionada (que era dependiente).

La representación gráfica de un tipo de entidad es un rectángulo etiquetado con el nombre del tipo de entidad, preferentemente con la primera en mayúscula.

Libro

INTERRELACIONES

La interrelación es la asociación o correspondencia entre entidades. La interrelación, como indica su nombre, representa una asociación o relación entre entidades del universo del discurso (de la base de datos).

Terminológicamente, nos encontramos con la existencia de:

- Tipo de interrelación: que es la estructura genérica de la interrelación entre entidades.
- Ocurrencia de la interrelación: que es la asociación concreta de ocurrencias de entidad de los tipos de entidad que intervienen en la interrelación.

Igual que ocurría entre los tipos de entidad y las ocurrencias de entidad, aquí también es necesario distinguir entre la estructura genérica de la interrelación (tipo de interrelación) y la asociación concreta de ocurrencias de entidad (ocurrencia de interrelación).

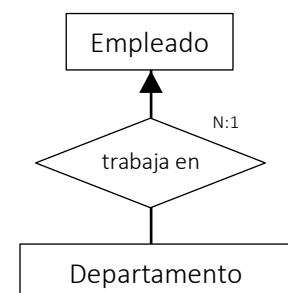
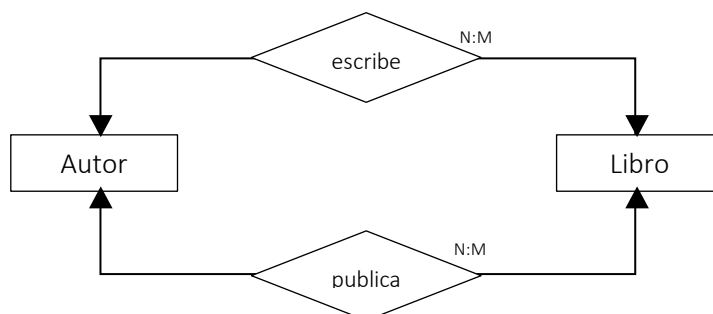
La representación gráfica de un tipo de interrelación es un rombo etiquetado con el nombre de la interrelación (preferentemente en minúscula) unido mediante arcos a los tipos de entidad que asocia.



Las interrelaciones, además, tienen unas características muy concretas que deben tenerse en cuenta:

- Nombre: el identificador de la interrelación.
- Grado: el número de entidades que participan en la interrelación: 2, 3... N
Dentro del grado 2 se incluyen las relaciones reflexivas (de una entidad con ella misma).
- Tipo de correspondencia: número máximo de ocurrencias de un tipo de entidad que pueden intervenir por cada ocurrencia del otro tipo de entidad asociado en la interrelación.
Para una interrelación de grado 2 entre las entidades A y B, existen tres tipos de correspondencia:
 - **Correspondencia 1:1.** Una ocurrencia de la entidad A se asocia como máximo con una única ocurrencia de la entidad B y viceversa. Ej.: un cliente con una cuenta bancaria que solo le pertenece a él.
 - **Correspondencia 1:N.** Una ocurrencia de entidad A se asocia con un número indefinido de ocurrencias de la entidad B, pero una ocurrencia de B se asocia como máximo con una ocurrencia de A. Ej.: una persona vive en una ciudad y en una ciudad viven muchas personas.
 - **Correspondencia N:M.** Una ocurrencia de la entidad A se asocia con un número indefinido de ocurrencias de la entidad B y viceversa. Ej.: un proveedor suministra varios productos y un mismo producto puede ser suministrado por diferentes proveedores.

La representación gráfica de las interrelaciones se vuelve entonces un poco más compleja, necesitando añadir una etiqueta al lado del rombo y/o una flecha hacia el lado N o M. Entre dos entidades puede existir una o más interrelaciones.



ATRIBUTOS

Un atributo es cada una de las propiedades o características que tiene un tipo de entidad o de interrelación. Estos son comunes a todas las ocurrencias del tipo de entidad o interrelación.

Por ejemplo:

- Atributos del tipo de entidad *Escritor* serían su nombre, nacionalidad, fecha de nacimiento, etc.
- Atributos del tipo de interrelación *Escribe* serían la fecha de inicio, fecha de fin del proyecto, la localización, etc.

Los atributos, como se ha dicho, tienen tipos diferenciados, así, podemos clasificar los atributos con los siguientes tipos:

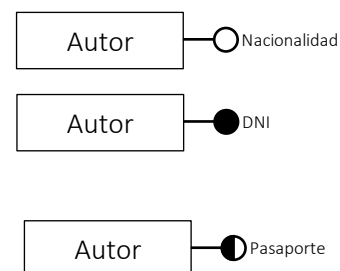
- **Atributo descriptor:** caracterizan una concurrencia pero no la distinguen del resto de ocurrencias.
- **Atributo identificador:** distingue de forma unívoca una ocurrencia de un tipo de entidad del resto de las ocurrencias. Puede estar formado por un solo atributo o por varios. Pueden existir varios atributos identificadores. A cada uno se le denomina atributo identificador candidato.
 - **Atributo Identificador Candidato (AIC):** se denominan de dos formas en función del rol que juega en la relación:
 - Atributo Identificador Principal (AIP): es el AIC elegido para representar el tipo de entidad.
 - Atributo Identificador Alternativo (AIA): son el resto de AIC del dominio de entidad.

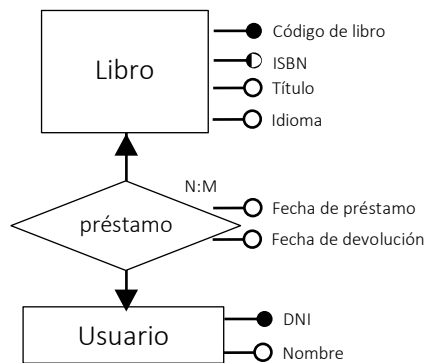
DOMINIOS

Los dominios representan el conjunto de posibles valores que puede tomar un atributo. El dominio tiene un nombre y una existencia propia con independencia de la entidad y/o atributo. Por ejemplo, un dominio de estados civiles podría ser el conjunto soltero, casado, etc. El atributo estado civil de empleado se define sobre este dominio.

REPRESENTACIÓN DEL ATRIBUTO/DOMINIO

- El **Dominio** se representa con un círculo u óvalo en cuyo interior aparece el nombre.
- El **Atributo** se escribe sobre el arco que une el dominio con el tipo de entidad o interrelación.
 - El **atributo identificador principal** se representa con un punto negro al que se le escribe sobre el arco el nombre de la clave.
 - El atributo identificador secundario se representa también con un óvalo, pero solo está marcado en negro en una mitad.
- **Conversión:** si el nombre del atributo coincide con el del dominio, puede representarse con un círculo con el nombre del atributo al lado.





Si quisiéramos representar el préstamo de un libro a un socio de la biblioteca, por ejemplo, tendríamos que implementar en el diagrama de nuestra base de datos algo similar a ←

ESTÁTICA DEL MODELO E/R

El modelo E/R básico es pobre en restricciones semánticas. Las restricciones limitan los tipos de estructuras a representar (inherentes-obligatoriedad del Atributo Identificador Personal; solo se permiten Interrelaciones

entre Entidades) o restringen los valores de los atributos o los tipos de correspondencia de las interrelaciones (Semánticas).

3.2. EXTENSIONES DEL MODELO E/R

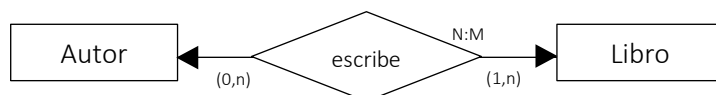
Los **Cardinales de un tipo de entidad** definen las cardinalidades máxima y mínima de los tipos de entidad participantes en un tipo de interrelación como el número máximo y mínimo de ocurrencias de un tipo de entidad que pueden estar interrelacionados con una ocurrencia del otro y otros tipos de entidad que participan en el tipo de interrelación.

Los Cardinales de un tipo de entidad se representan gráficamente como una etiqueta del tipo (0,1), (1,1), (0,n) o (1, n) según corresponda, que se coloca en el arco que une el correspondiente tipo de entidad con el rombo que representa la interrelación.

A partir del modelo E/R básico, diferentes autores han propuesto lo que se denominan extensiones del modelo, con el objetivo de lograr representar una mayor semántica del problema. Entre estas extensiones se encuentra la cardinalidad, la dependencia en existencia o identificación y las jerarquías.

! IMPORTANTE: la notación empleada para estas extensiones en los diferentes libros sobre bases de datos no suele ser uniforme, por lo que se aconseja para esta parte del temario seguir la notación empleada en el libro de texto base. El uso de otro tipo de notación es correcto siempre que se apliquen los conceptos adecuadamente.

Un escritor escribe como mínimo un libro y como máximo n libros, y un libro puede ser escrito por ningún escritor (anónimo) o por muchos (N).



Una persona puede comprar o no varios coches. Todo coche es comprado por una persona.



Dependencia en existencia:

- Entidades fuertes y débiles
- Se produce cuando las ocurrencias del tipo de entidad débil no pueden existir sin la ocurrencia del tipo de entidad regular de la que dependen.

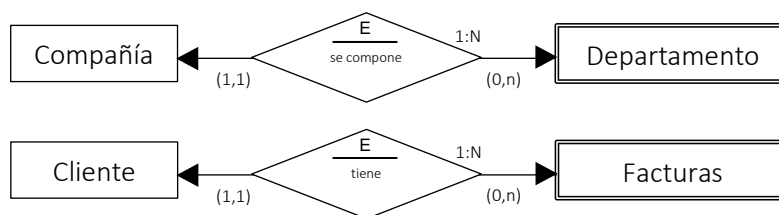
- Si desaparece la ocurrencia del tipo de entidad regular tienen que desaparecer las ocurrencias del tipo de entidad débil que dependían en existencia de ella.
- Pertenece al tipo de interrelación débil: un tipo de entidad débil está vinculado con un tipo de entidad regular.

Los tipos de interrelación se clasifican en:

- **Regulares:** si asocian tipos de entidad regulares.
- **Débiles:** si asocian un tipo de entidad regular con un tipo de entidad débil o dos tipos de entidad débiles.

Un tipo de interrelación débil exige siempre que las cardinalidades del tipo entidad regular sean (1,1) para que toda ocurrencia de la entidad débil deba tener asociada una ocurrencia de la entidad regular. Si hubiera alguna ocurrencia de la entidad débil no asociada a una ocurrencia de la entidad regular no existiría dependencia, lo cual va contra la suposición inicial de que la entidad débil depende de la entidad regular.

La representación gráfica de la **existencia** se realiza añadiéndole la etiqueta “E” al rombo que representa la interrelación débil.

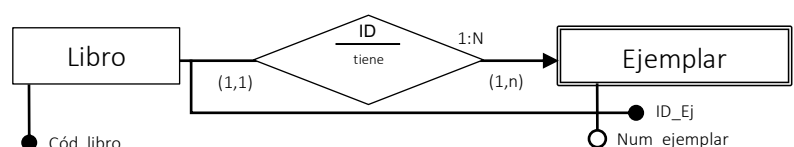


Dependencia en identificación:

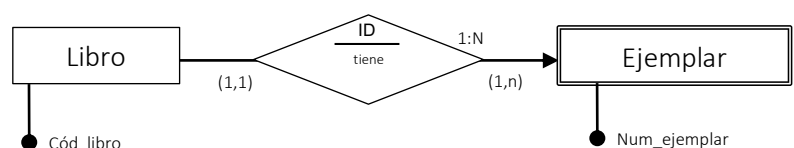
Hay dependencia en identificación cuando, además de la dependencia en existencia, las ocurrencias del tipo de entidad débil no se pueden identificar solo mediante sus atributos, sino que se tiene que añadir el atributo identificador principal del tipo de entidad regular de la que dependen.

Al igual que la Dependencia en Existencia también pertenece al tipo de interrelación débil: un tipo de entidad débil está vinculado con un tipo de entidad regular. La diferencia es que en la entidad débil no hay un Atributo Identificador Personal que permita diferenciar las ocurrencias entre sí, y es necesario utilizar alguno de los atributos de la entidad débil junto con el Atributo Identificador Principal de la entidad regular de la que depende la entidad débil.

Su representación gráfica se realiza añadiendo la etiqueta ID al rombo que representa la interrelación débil.



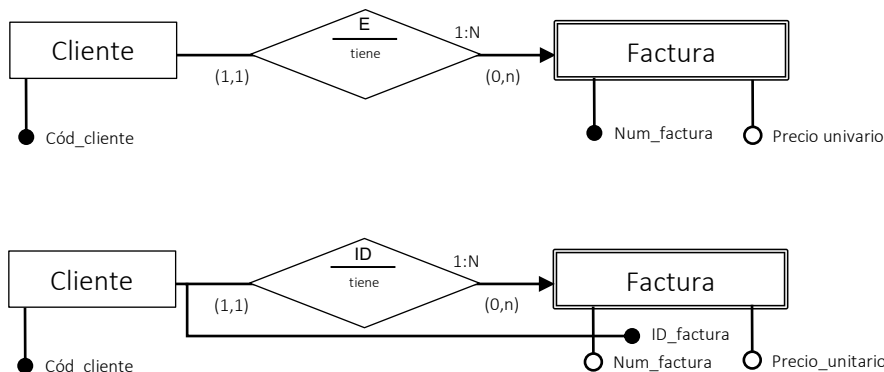
El Atributo Identificador Principal de la entidad Ejemplar será Cód_libro (AIP de Libro) y Num_Ej. Una representación alternativa que podríamos hacer de este mismo caso sería →



En este caso no hace falta poner ID_Ej, ya que la palabra ID en la relación aclara que la clave de Ejemplar estará compuesta por Cód_libro + Num_ejemplar.

La interrelación que existe entre Libro y Ejemplar es dependiente en identificación ya que un Ejemplar determinado, además de depender en existencia de un cierto Libro, está identificado con la clave del libro (Código_libro) del cual depende el Ejemplar más un código propio, el Num_ejemplar.

La principal **diferencia** entre la **Dependencia en existencia** y la **Dependencia en interrelación** se da de forma visual en la etiqueta y forma que tienen de representar la relación.



En este ejemplo, el primer caso (existencia, factura única), cada factura es identificada de forma unívoca por el atributo Num_factura (que es único para cada factura y constituye

su AIP); mientras que para el segundo caso (identificación, factura no única) cada factura debe identificarse, además de por su Num_factura (que no será único para cada factura y, por lo tanto, podrá tener duplicados, ya que depende del cliente del que se trate) por el Cód_cliente. La combinación de ambos da como resultado la identificación de una determinada factura en ID_factura como: **ID_factura = Cód_cliente + Num_factura**.

Identificará de forma única cada ocurrencia de factura, siendo el AIP de factura; dicho de otra forma:

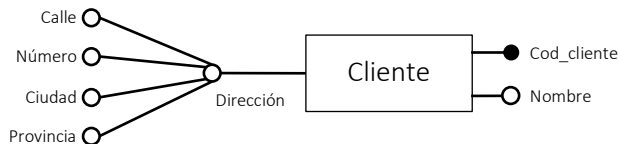
En la **Dependencia en Existencia** el atributo Num_factura del tipo de entidad débil factura permite distinguir cada una de las ocurrencias de la entidad factura, es decir, Num_factura es el AIP de dicha entidad. (Valores distintos para clientes distintos). En la **Dependencia en Identificación** el atributo Num_factura por si solo no permite distinguir cada una de las ocurrencias de la entidad Factura. Num_factura no es el AIP de dicha entidad. El AIP de factura será Cod_cliente + Num_factura. (Valores repetidos de Num_factura para clientes distintos).

Para concluir las dependencias, se puede resumir que:

- La dependencia de existencia no implica una dependencia de identificación, hecho que sí sucede en el caso inverso pues una entidad que depende de otra por su clave no tiene sentido sin la existencia de esta última.
- En una relación con tipo de correspondencia N:M nunca habrá entidades débiles porque la supuesta ocurrencia de la entidad débil que tendría que borrarse podría estar asociada a más de una ocurrencia de la entidad fuerte, lo que implica la imposibilidad de borrarla y por tanto la contraposición con la definición de entidad débil.

Atributos multivaluados y compuestos

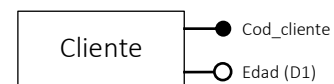
Los **atributos multivaluados** son aquellos atributos que para una misma ocurrencia de la entidad pueden tomar más de un valor. Como ejemplo, si cada cliente puede tener más de un teléfono y es de interés guardar todos sus posibles valores, el atributo teléfono sería multivaluado o multiocurrente. →



← Los **atributos compuestos** son aquellos que agrupan en sí mismos por afinidad o por forma de uso más de un atributo. Como ejemplos podemos considerar, por su forma habitual de utilización, el atributo de

dirección que engloba los atributos simples de calle, número, ciudad y provincia; o, por su significado, el atributo nombre engloba el nombre y los dos apellidos.

Existen también **atributos derivados**, que son aquellos que se obtienen a partir de atributos almacenados, por ejemplo, el atributo edad de una persona puede derivarse a partir de la diferencia entre el atributo fecha de nacimiento y de la fecha corriente. También el atributo número de empleados de un departamento puede obtenerse a partir del número de empleados que trabajan en ese departamento. Los atributos derivados se representan de la siguiente forma.

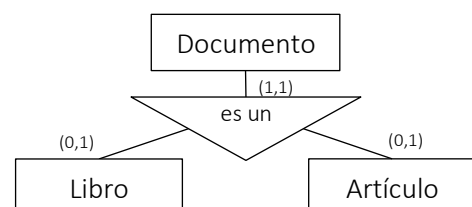


Como puede apreciarse en la figura, el atributo derivado simplemente consiste en añadir un identificador variable (en este caso, D1) entre paréntesis.

Generalización y herencia

La **generalización** es el tipo de interrelación que existe entre un tipo de entidad y los tipos de entidad más específicos que dependen de él. La descomposición de un tipo de entidad crea una jerarquía de tipos de entidad donde se puede distinguir un supertipo del cual dependen varios tipos. La generalización es una operación de abstracción que consiste en identificar una serie de características comunes a varios objetos que, además, se diferencian entre sí por otras características específicas. De esta forma se dividen los objetos en categorías y subcategorías siguiendo la norma de que en la categoría principal se detallan las características comunes y en las subcategorías las características específicas. Habitualmente conocemos a las generalizaciones con el nombre de *jerarquías*.

La representación gráfica de este tipo de interrelación consiste en un triángulo invertido con la base paralela al rectángulo que representa el supertipo y conectado a este por los angulares a los subtipos. Las cardinalidades siempre son (1,1) en el supertipo y (0,1) en los subtipos.



En el ejemplo de la figura, el tipo de la entidad Documento es el supertipo de la jerarquía y constituye la generalización de los subtipos de entidades Libro y Artículo. A su vez, estos últimos son una especialización del supertipo Documento.

Una de las características más importantes de las jerarquías es la **herencia**, por la que los atributos de un supertipo son heredados por sus subtipos. En el ejemplo anterior, Libro y Artículo poseerán todos

los atributos del supertipo documento más los atributos específicos de cada subtipo. Gracias a la herencia en una generalización no es necesario escribir los atributos comunes (incluido el Atributo Identificador Principal) más que en el supertipo.

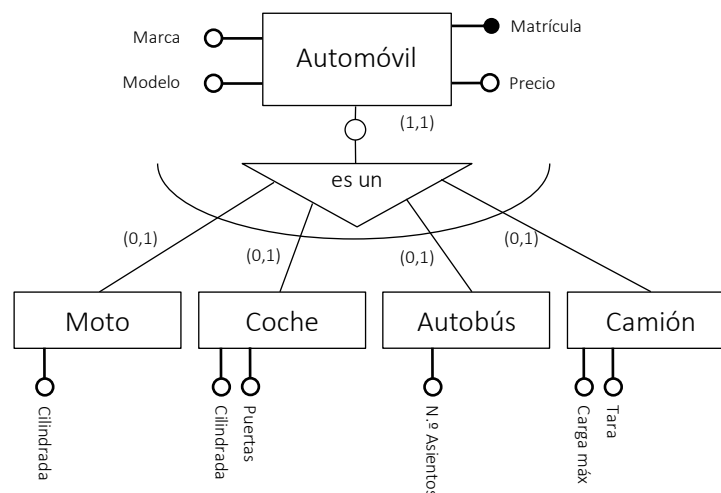
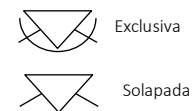
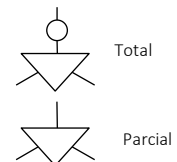
¡Lo que tienes que tener claro!

- En una generalización, los atributos comunes a los subtipos (incluidos los identificadores) se asignan al supertipo, mientras que los atributos específicos se asocian al subtipo correspondiente.
- Las interrelaciones que afectan a todos los subtipos se asocian al supertipo, dejándose para los subtipos las interrelaciones específicas en las que participa solo el correspondiente subtipo.
- El Atributo Identificador Principal (AIP) se sitúa en el supertipo y no puede haber otro AIP en los subtipos porque estos ya heredan el AIP del supertipo. Esta regla no impide que los subtipos puedan tener Atributos Identificadores Alternativos (AIA).

Ocurrencia

Recuerda que una ocurrencia es una entrada de datos específica dentro de una Entidad; es decir, el conjunto de valores particulares que toman todos los campos que lo componen en un momento determinado. Podemos clasificar las ocurrencias en función de la localización de las mismas:

- **Total / parcial:** Si no hay ocurrencias en el supertipo que no pertenezcan a ningún subtipo, la generalización es total. En caso contrario es parcial. Para representar esta en una jerarquía se emplea un círculo por encima del triángulo. Si la jerarquía es total se dibuja, si es parcial se omite.
- **Exclusiva / solapada:** Si puede haber ocurrencias que pertenezcan a más de uno de los subtipos, entonces se tiene una generalización solapada, si la ocurrencia no tiene *duplicados* es exclusiva. Para representar esta en una jerarquía se emplea un arco por debajo del triángulo. Si la jerarquía es exclusiva se dibuja, si es solapada se omite.



En el ejemplo de la figura se recoge la creación de una jerarquía o generalización del concepto de automóvil, todos los automóviles tienen una matrícula (AIP), una marca, un modelo y un precio. A partir de aquí es preciso distinguir entre: coches (tienen puertas y cilindrada), motos (solo cilindrada), autobuses (número de asientos) y camiones (carga máxima y tara).

La jerarquía representada en el ejemplo es del tipo total y exclusiva. lo que significa que (total) todo automóvil es un coche, una moto, un autobús o un camión; no hay automóvil que no pertenezca a ninguna de estas categorías. Además, (exclusiva), si un automóvil es un coche no puede ser ni una moto ni un autobús ni un camión. Es decir, si el automóvil pertenece a un subtipo no puede pertenecer a otro.

3.3.CONSTRUIR UN DIAGRAMA E/R

IDENTIFICADOR DE LOS CONSTRUCTORES

Las entidades aparecen reflejadas en el enunciado normalmente como nombres, al igual que los atributos. En cambio, las interrelaciones suelen aparecer reflejadas como verbos.

Hay que advertir que la actividad de diseñar bases de datos empleando el Modelo E/R exige cierta práctica; salvo contados casos, a diseñar bases de datos se aprende diseñando, lo que quiere decir que cuantos más problemas de modelado se realicen mejores serán las habilidades y agilidad adquiridas.

A partir de la lectura cuidadosa del enunciado de los requisitos o necesidades de los usuarios se van desgranando los componentes de la base de datos. En el caso del Modelado E/R interesa detectar las entidades, los atributos y las interrelaciones o relaciones. Para ello se pueden seguir las reglas citadas teniendo presente que no son reglas fijas ni invariables y hay que tomarlas como representativas de los casos más habituales.

ELECCIÓN ENTRE VARIOS CONSTRUCTORES

Entidades/atributos

- Las entidades poseen información descriptiva y los atributos no.
- Una entidad debe estar caracterizada por algo más que su Atributo Identificador Principal (AIP). Si existe información descriptiva sobre el objeto, podría clasificarse como entidad.
- Si solo se necesita un identificador (AIP) para el objeto, debería clasificarse como un atributo a no ser que el objeto se relacione con más de una entidad, en cuyo caso puede clasificarse como entidad.

Ejemplo 1: En el supuesto de “los almacenes se localizan en ciudades”, si existe información descriptiva sobre la ciudad debería clasificarse como una entidad. Si solo se necesita el AIP “nombre ciudad”, entonces debería clasificarse como un atributo de almacén.

Ejemplo 2: En otro supuesto “los proveedores tiene asignadas varias ciudades”, podría dejarse ciudad como entidad relacionada con almacén y proveedor o como atributo de ambas.

Entidades/atributos multivaluados

Algunos autores prefieren incorporar en los esquemas E/R un atributo multivaluado como una entidad y otros como un atributo simple.

Si suponemos que el atributo multivaluado tiene un número limitado y no muy elevado de ocurrencias, entonces formará parte de la entidad como atributo siempre que el concepto representado no esté relacionado con otras entidades.

En un supuesto de “información sobre empleados”, si se considera que un empleado puede tener varios números de teléfono, el teléfono será un atributo del empleado. En el caso de “información sobre

profesores”, la información del campus en el que imparte docencia puede ser un atributo multivaluado o una entidad si va a relacionarse con otras entidades: departamento, etc.

Atributos/Interrelaciones

Lo normal es no tener problemas a la hora de distinguirlos.

Si el atributo refleja una referencia a otra entidad puede ser conveniente que se convierta en una interrelación.

Consideramos las entidades empleado y departamento. Si en empleado hay un atributo que representa el departamento que dirige, *Direccion_dep*, se puede sustituir ese atributo por una relación “dirige” entre empleado y departamento.

PASOS EN LA REALIZACIÓN DEL DIAGRAMA E/R:

1. Estudiar el enunciado con el objetivo de identificar y enumerar las posibles entidades, teniendo en cuenta que, como se comentó anteriormente, un tipo de entidad es un sustantivo dentro de una oración con una serie de propiedades.
2. Identificar y enumerar las posibles interrelaciones, teniendo en cuenta que, en general, una interrelación viene reflejada por un verbo dentro de una oración que relaciona dos objetos. Para que sea más sencillo saber qué tipo de entidades están relacionadas podemos construir una Matriz de Entidad donde la primera fila y columna enuncien los tipos de entidad y señalen en el cruce de filas y columnas aquellas interrelaciones que tengan.
3. Dibujar las interrelaciones (estudiando el tipo de correspondencia y las cardinalidades) y los tipos de entidad con los atributos correspondientes.
4. Comprobar si el diseño realizado contiene la mayor cantidad posible de supuestos semánticos del problema. En caso contrario, refinar los pasos anteriores. El diseño de un diagrama E/R es un proceso iterativo y normalmente las soluciones no son únicas porque cada diseñador puede tener una visión particular del mundo real. Sin embargo, sí puede estudiar si un determinado esquema E/R refleja mejor que otro los supuestos semánticos del problema.

Ejemplo. Se quiere diseñar una base de datos que permita a una universidad gestionar la matriculación de sus alumnos. La información de la que se parte es la siguiente:

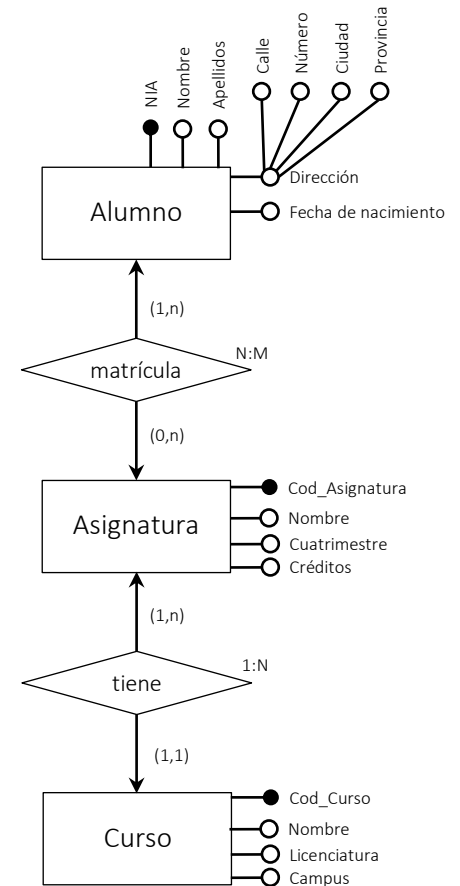
- Los alumnos se caracterizan por su NIA, Nombre, Apellidos, Dirección y Fecha de nacimiento.
- Los alumnos pueden matricularse de las asignaturas que deseen. De las asignaturas interesa conocer su Código de asignatura, Nombre, Cuatrimestre, Créditos y Precio.
- Cada asignatura solo puede pertenecer a un curso. De los cursos interesa guardar su Código de curso, Nombre, Licenciatura y Campus.

Interpretación de los tipos de correspondencia:

- Un alumno puede matricularse en varias asignaturas (Matrículas = N:M)
- De una asignatura se pueden matricular varios alumnos (Matrículas = N:M)
- Un curso tiene varias asignaturas (tiene = 1:N)
- Una asignatura pertenece a un solo curso (tiene = 1:N)

Interpretación de las cardinalidades:

- Un alumno puede matricularse de varias asignaturas o ninguna (0,n)
- De una asignatura se matricula al menos un alumno (1,n)
- Un curso tiene al menos una asignatura (1,n)
- Cada asignatura pertenece a un solo curso (1,1)



TEMA 4. MODELO RELACIONAL

El modelo relacional fue presentado por Codd a finales de los años sesenta. Codd propuso un modelo de datos basado en la teoría matemática de las relaciones, en el que los datos se estructuran lógicamente en forma de relaciones. El modelo relacional fue publicado en un artículo de 1970, titulado *A Relational Model of Data for Large Shared Data Banks*, y en él se encuentran las bases teóricas del modelo.

Los objetivos que persigue el Modelo Relacional son:

- **Independencia física:** el modelo en que se guardan los datos no influye en su manipulación lógica.
- **Independencia lógica:** las modificaciones en los elementos de la base de datos no deben repercutir en los programas y/o usuarios que acceden a las vistas.
- **Flexibilidad:** ofrece a cada usuario lo que necesita.
- **Uniformidad:** en las estructuras lógicas de los datos (todo son relaciones) lo que facilita su utilización.
- **Sencillez:** facilidad de comprensión y utilización.

Para conseguir los objetivos, Codd introduce el concepto de relación como estructura básica del modelo en su parte estática. Con respecto a la parte dinámica, propone un conjunto de operadores que se aplican a las relaciones. Codd propuso el álgebra y el cálculo relacional.

El modelo relacional propuesto por Codd trataba de resolver algunos de los problemas asociados a los modelos de datos dominantes en la época: el modelo jerárquico y el modelo en red. Para ello, propuso un modelo basado en la teoría de las relaciones matemáticas, asociando una relación con la estructura básica de la información.

4.1. ESTRUCTURA DEL MODELO RELACIONAL

El elemento básico es la relación que se puede representar como la siguiente tabla:

Nombre			
Atributo 1	Atributo 2	...	Atributo N
Valor 11	Valor 12	...	Valor 1N
...
Valor N1	Valor N2	...	Valor NN

Tupla 1

...

Tupla N

- El **nombre** identifica la relación, aunque no es obligatorio. Existen relaciones sin nombre (por ejemplo, los resultados intermedios).
- Los **atributos** representan propiedades de la relación.
- El conjunto de filas se denomina **tuplas** y contienen los valores que toma cada uno de los atributos de la relación. Representan los objetos particulares

Las características de la relación del modelo para tener en cuenta son:

- **Grado:** número de atributos. Poco variable en el tiempo. El grado correspondería a la parte estática (parte que no varía mientras no cambien las necesidades del universo del discurso).
- **Cardinalidad:** número de tuplas. Normalmente, variable en el tiempo. Correspondería a la parte dinámica, parte que va añadiendo datos.

<u>DNI</u> Números <60 millones	<u>Apellidos</u> 75 caracteres	<u>Nombres</u> 25 caracteres	<u>Puesto</u> Técnico Secretaria Director	<u>Dominios</u> Empleado
DNI	APELLIDO	NOMBRE	PUESTO	ATRIBUTOS
1111111	García	Daniel	Técnico	Cardinalidad (3)
2222222	Pérez	Manuel	Secretaria	
3333333	Sánchez	Belén	Director	
Grado (4)				

DOMINIOS Y ATRIBUTOS

- Un dominio es un conjunto finito de valores homogéneos y atómicos: V_1, V_2, \dots, V_n
- Todo dominio tiene un nombre y un tipo de datos.
- Un atributo toma sus valores de un dominio y está asociado a la relación. El dominio existe por sí mismo.
- Es habitual que el atributo y el dominio tengan el mismo nombre.
- Distintos atributos pueden definirse sobre el mismo dominio. En este caso, los atributos tendrán distinto nombre porque una relación no puede tener dos atributos iguales.
- El atributo “puesto” definido en el dominio “puesto” indica que ese atributo juega el rol del puesto de trabajo del empleado.
- Se dice que D es el dominio de A y se suele expresar como $\text{Dom}(A)$.

INTENCIÓN DE UNA RELACIÓN

Representa la parte definitoria y estática de una relación. También se denomina esquema de relación.

- EMPLEADO (dni: DNI, Apellido: APELLIDO, Nombre: NOMBRES, Puesto: PUESTOS)

Su notación en el esquema de relación es la siguiente: $R(\{A_i:D_i\}_{i=1..n})$. Al conjunto de n pares se le denomina Cabecera de la relación y se corresponde con la primera fila de la tabla. El esquema de relación, por tanto, está constituido por el nombre de la relación y la cabecera de la relación.

EXTENSIÓN DE UNA RELACIÓN

Definición de la relación a través del conjunto de tuplas que, en un momento determinado, cumplen el esquema de relación.

La extensión se denomina relación o estado de relación. Por tanto, una relación o extensión está constituida por el par <esquema, cuerpo>, denominándose cuerpo de la relación al conjunto de m tuplas (Cardinalidad) en la que cada una está formada por un conjunto de n pares (Grado) atributo-valor.

Una base de datos relacional está formada por varias relaciones que varían con el tiempo.

CLASES DE RELACIÓN

- **Relaciones básicas:** se definen con independencia de las demás relaciones. Se corresponden con el nivel conceptual. Siempre tienen un nombre:
- **Relaciones derivadas:** se definen como resultado de ejecutar operaciones sobre otras relaciones básicas o derivadas. Pueden no tener existencia física, bastando con almacenarse su definición. Pueden no tener un nombre específico:
 - **Vistas:** se corresponden con el nivel externo de la arquitectura ANSI. Se definen dando un nombre a una expresión de consulta y no tienen datos propios almacenados.

- **Instantáneas:** se corresponden con el nivel interno de la arquitectura ANSI. Tienen datos propios almacenados que son el resultado de ejecutar una consulta específica.
- **Resultados de una consulta:** no tienen nombre y pueden ser tanto resultados finales como intermedios.

Otra clasificación de las relaciones tiene en cuenta si las relaciones son permanentes o temporales; en el primer caso, la relación permanece almacenada mientras que en el segundo, desaparece cuando termina la sesión o la transacción.

CLAVES

- **Clave candidata:** es un conjunto no vacío de atributos que identifican unívoca y mínimamente cada tupla de la relación.
 - **Clave primaria:** es aquella clave candidata que se escoge para identificar las tuplas.
 - **Clave alternativa:** son las claves candidatas que no actúan como clave primaria.
- **Clave ajena:** de una relación R2 es un conjunto no vacío de atributos cuyos valores deben coincidir con los valores de la clave candidata de una relación R1 (R1 y R2 pueden ser iguales). Los dominios de las clave candidata y de la clave ajena deben ser iguales.

El conjunto de claves de una relación se denomina Claves Candidatas. De este conjunto se elige una cualquiera para actuar como clave candidata. El resto se denominan Claves Alternativas. La elección de clave primaria se define automática o manualmente, según estipule el diseñador o tenga elección el usuario. También podría obedecer a criterios de eficiencia en la implementación posterior.

La Clave Ajena permite implementar el mecanismo de Integridad Referencial por el que las relaciones están asociadas.

4.2. RESTRICCIONES

RESTRICCIONES INHERENTES

En el modelo relacional se derivan de la definición matemática de la relación y son:

- No hay tuplas duplicadas (implica la existencia de una clave primaria).
- El orden de las tuplas no es relevante.
- El orden de los atributos no es significativo.
- Cada atributo solo puede tomar un valor de su dominio. No se permiten grupos repetitivos.

La inexistencia de duplicados es una restricción del modelo, pero no de los productos comerciales que sí permiten que una tabla tenga filas duplicadas, porque no obligan a que la tabla tenga clave primaria.

Cuando una relación no permite grupos repetitivos en sus atributos se dice que está en *primera forma normal*.

DNI	Apellido	Nombre	Puesto	Idioma
1111111	García	Daniel	Técnico	EN, FR
2222222	Pérez	Manuel	Secretaría	DE
3333333	Sánchez	Belén	Director	EN, IT

Esta tabla sería errónea conceptualmente, ya que en *Idioma* hay tuplas que guardan más de un valor. Para representar esta situación habría que repetir (en cada fila con valores múltiples) la fila con cada idioma dejando iguales los demás atributos. En este caso la clave primaria sería el DNI y el idioma.

A estas restricciones se añade la Regla de Integridad de Entidad, que impone lo siguiente: **ningún atributo de la clave primaria de una relación puede ser desconocido o inexistente**, es decir, tomar un valor nulo.

La Regla de Integridad de Entidad asegura que todos los atributos que forman una clave tienen contenido, es decir, que no tienen valores nulos y será obligatorio introducir algún valor para ellos.

RESTRICCIONES SEMÁNTICAS

Clave primaria (primary key): permite declarar un atributo(s) como clave primaria, por lo que sus valores no pueden repetirse ni admitir nulos (valores desconocidos).

Unidad (unique): indica que los valores del atributo(s) no pueden repetirse; se permite definir claves alternativas.

Obligatoriedad (not null): indica que el atributo(s) no admite valores nulos.

Las Restricciones Semánticas permiten captar reglas que existen en el Universo del Discurso. Algunas de las restricciones semánticas permitidas por el Modelo Relacional son la identificación de la clave primaria, la posibilidad de definir un atributo o varios como únicos (no pueden contener valores duplicados), la obligatoriedad de introducir un valor para un atributo o la integridad referencial.

Las palabras *primary key*, *unique* y *not null* corresponden a las denominaciones que se emplean en lenguaje SQL (permite definir y manipular estructuras de datos) para hacer referencia a estas restricciones semánticas.

Integridad Referencial (foreign key): si en una relación R2 hay un conjunto de atributos que son una clave candidata de una relación R1, los valores de ese conjunto de atributos deben coincidir con un valor de la clave candidata (R1) o ser nulos. El conjunto de atributos es, por tanto, una clave ajena de la relación R2. R1 y R2 pueden ser iguales y la clave ajena en R2 puede ser parte (o la totalidad) de la clave primaria de R2.

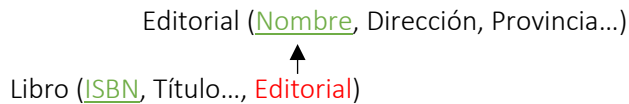
La Integridad Referencial permite que los valores que pueden aparecer en la clave ajena coincidan (o sean nulos) con los valores que aparecen como clave principal en otra relación (que podría ser la misma en caso de relaciones reflexivas). La palabra *foreign key* se utiliza en SQL para definir las claves ajenas; cuando se establece la integridad referencial del SGBD se encarga de comprobar que los valores de la clave ajena son correctos; es decir, que corresponden a valores en la clave primaria de otra relación.

Editoriales (R1)				Catálogo (2)			
Nombre	Dirección	Provincia	...	ISBN	Título	...	Editorial
Alfaguara	C/ aaa,1	Madrid	...	978-84-...	El Quijote	...	Gredos
Gredos	C/ bbb,2	Madrid	...	978-84-...	100 años de soledad	...	Alfaguara
Planeta	C/ ccc,3	Barcelona	...	978-84-...	La Colmena	...	NULO

El atributo Editorial en la relación Catálogo es la clave ajena primaria. Nombre de la relación Editorial.

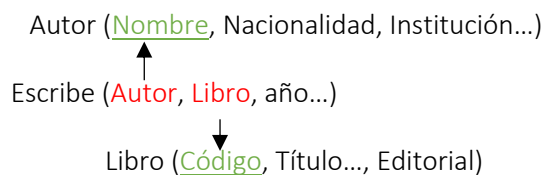
REPRESENTACIÓN DE LA INTEGRIDAD REFERENCIAL

Una posible forma de representar las claves ajenas y la integridad referencial es la siguiente:



Este dibujo se denomina *Grafo* o Diagrama Relacional y representa la estructura de la información. La flecha indica una asociación entre las relaciones Editorial y Catálogo mantenida a través del campo Editorial (clave ajena) en la relación Catálogo.

Los valores que puede tomar el campo Editorial de la relación Catálogo son los que aparezcan previamente en el campo Nombre de la relación Editorial. Como puede verse en la figura no es necesario que la Clave Ajena tenga el mismo nombre que la Clave Primaria a la que referencia. En el siguiente ejemplo, las claves ajenas de *Escribe* son también su Clave Primaria:



En este ejemplo, ninguna de las claves ajenas de Escribe (Autor y Libro) puede contener valores nulos porque son la clave primaria de Escribe.

TERMINOLOGÍA DE LA INTEGRIDAD REFERENCIAL

- **Relación que referencia:** es la relación donde se incluye la clave ajena. En la definición sería R2.
- **Relación referenciada:** es la relación que contiene de clave candidata, cuyos valores coincidirán con los de la clave ajena. En la definición sería R1.

INTEGRIDAD REFERENCIAL: CONSECUENCIAS DE LAS OPERACIONES DE BORRADO/MODIFICACIÓN

- **Operación en cascada (Cascade):** el borrado de tuplas en la relación (R1) con la clave referenciada (o la modificación de esa clave) desencadena el borrado o la modificación en cascada de las tuplas, con ese valor en la clave ajena, de la relación que referencia (R2).
Al borrar una editorial (en Editorial) se borrarán automáticamente todos los libros (en Catálogo) publicados por ella. La modificación es análoga.
- **Operación con puesta a nulos (Set null):** el borrado de tuplas en la relación (R1) con la clave referenciada (o la modificación de esa clave) implica poner nulos en las tuplas, con ese valor en la clave ajena, de la relación que referencia (R2). La clave ajena debería permitir valores nulos para que esta opción tenga sentido.
Al borrar una editorial (en Editorial) se pone automáticamente un valor nulo en el campo editorial de los libros (en Catálogo) publicados por ella. La modificación sería análoga.
- **Operación con puesta a valor por defecto (set default):** el borrado de tuplas en la relación (R1) con la clave referenciada (o la modificación de esa clave) implica poner un valor por defecto en las tuplas, con ese valor en la clave ajena, de la relación que referencia (R2). El valor por defecto se establece al crear la tabla.
Al borrar una editorial (en Editorial) se ponen automáticamente a un valor predeterminado los libros (en Catálogo) publicados por ella. La modificación sería análoga.

RESTRICCIONES SEMÁNTICAS

Rechazo

Es un tipo de restricción semántica que consiste en formular una condición sobre un conjunto de atributos, tuplas o dominios. La condición debe verificarse para que la operación de actualización sea admitida. En caso de no cumplirse, se rechaza la operación.

Con el objetivo de mantener en la base de datos la mayor cantidad semántica posible del Universo del Discurso, el Modelo Relacional permite definir operaciones que eviten estados incorrectos de la base de datos. Un ejemplo sería cuando tenemos una relación de electores. Uno de los atributos indica la edad del votante. Como es conocido, existe un límite para poder votar. Una operación de rechazo simplemente consistiría en evitar votantes que no cumplen ese límite. De esta forma, el rechazo comprueba la validez de los datos y decide, en función de las reglas semánticas, si son válidos para incorporarse a la base de datos o no lo son. En caso de no serlo, se rechazaría la operación.

Tipos

En el modelo relacional descrito por SQL92 hay dos tipos de rechazo:

- **Verificación (check):** comprueba en toda la actualización si se cumple la condición (definida sobre un único elemento). Si no se cumple se rechaza la actualización.
- **Aserción (assertion):** igual que la anterior con la diferencia de que la condición puede definirse sobre varios elementos, por ejemplo, dos relaciones. Debe tener un nombre porque tienen existencia propia.

La propia existencia del Dominio impone una importante componente semántica. Estas dos reglas son declarativas. La verificación sería una condición intraelemento (la comprobación se hace en la misma relación), mientras que la aserción sería una condición interelementos (la comprobación se hace analizando varias relaciones).

El esquema de relación está compuesto por los atributos de la relación, los dominios sobre los que toman valor y las restricciones de integridad que afectan a cada uno de los elementos del esquema (restricciones intraelementos.)

$$R \langle A:D,s \rangle$$

A (atributos), D (dominios), s (restricciones intraelementos)

El esquema relacional es una colección de esquemas de relación junto con las restricciones de integridad definidas sobre más de un esquema de relación (restricciones interelementos).

$$E \langle \{R_i\}, \{I_i\} \rangle$$

E (nombre del esquema relacional), $\{R_i\}$ (conjunto de esquemas de relación), $\{I_i\}$ (restricciones interelementos)

Disparadores (Triggers)

Permiten, además de indicar una condición, especificar la acción que se llevará a cabo si la condición se hace verdadera. Pueden interpretarse como reglas del tipo evento-condición-acción (ECA) que especifican que cuando se produce un evento, si se cumple una condición, entonces se realiza una determinada acción.

Los *Triggers* amplían la funcionalidad de las bases de datos ya que permiten establecer reglas que deben cumplirse cuando se realicen operaciones Insert, Update o Delete contra una relación específica. No están soportados en SQL92, pero sí en muchos productos comerciales.

Además de forzar el cumplimiento de las reglas, permiten recoger semántica que no puede registrarse mediante las reglas de transformación del modelo E/R al modelo relacional. Los disparadores son procedimentales; el usuario escribe el procedimiento (acción) que debe ejecutarse cuando se produce el evento. La condición no es obligatoria.

4.3 GRADO RELACIONAL

Un esquema relacional se representa mediante un grafo conocido como Grafo Relacional; este grafo representa gráficamente las relaciones de la base de datos y las restricciones de integridad referencial (clave ajena). En el Grafo Relacional se incluyen, además de las relaciones y restricciones de integridad referencial, otras restricciones como clave primaria, clave alternativa y obligatoriedad.

El Grafo es el elemento de representación habitualmente empleado en el Modelo Relacional; corresponde al resultado de aplicar el Modelo Relacional al problema de diseño. En él, se muestran las relaciones y asociaciones entre ellas mediante claves ajenas; también se deben incluir las opciones de Borrado y Modificación.

La **notación** que se utiliza es la siguiente:

- Las relaciones se escriben con mayúscula.
- Los atributos pueden expresarse al lado del nombre de la relación, bien entre paréntesis, bien en cajas rectangulares consecutivas.
- Las claves primarias se subrayan con trazo continuo.
- Las claves alternativas se subrayan en trazo discontinuo.
- Las claves ajenas referencian a la relación en la que está la clave primaria mediante una flecha dirigida hacia el nombre de la relación o hacia la propia clave primaria de la relación.
- Los atributos que pueden tomar valores nulos (no son obligatorios) aparecen con un asterisco.
- Para las opciones de borrado y actualización relativas a la integridad referencial, pueden emplearse abreviaturas, por ejemplo:

Para el borrado:

○B.R. Borrado restringido

○B.C. Borrado en cascada

○B.N. Borrado con puesta a nulos

○B.D. Borrado con puesta a valor por defecto

Para la modificación:

○MR. Modificación restringida

○M.C. Modificación en cascada

○M.N. Modificación con puesta a nulos

○M.D. Modificación con puesta a valor por defecto

Alumno (Código alumno, Apellidos, Nombre, Dir., Pob., CP, Tel., Fecha nac., Beca, Licenciatura)

Matrícula (Alumno, Asignatura)

Asignatura (Código asignatura, Nombre, Créditos, Cuatrimestre, Obligatoriedad, Precio, Curso)

Curso (Código curso, Nombre, Campus, Licenciatura)

El Grafo Relacional anterior puede verse que es necesario describir las relaciones y sus atributos con el mayor detalle posible (claves primarias, claves alternativas, atributos opcionales, claves ajenas, etc. No debemos olvidar nunca las opciones de borrado y modificación.

TEMA 5. TRANSFORMACIÓN DEL MODELO E/R AL MODELO RELACIONAL

Las reglas de transformación permiten obtener un diseño lógico basado en el Modelo Relacional a partir de un esquema conceptual expresado por medio del Modelo Conceptual (Diagrama E/R). Las tres reglas básicas para transformar un esquema conceptual E/R en un esquema relacional son:

- Toda entidad del esquema E/R se transforma en una relación en el esquema relacional.
- Toda interrelación N:M del esquema E/R se transforma en una relación en el esquema relacional.
- Las interrelaciones 1:N dan lugar, o bien a un fenómeno de propagación de clave, o bien a una relación.

5.1. TRANSFORMACIÓN DE ENTIDADES, DOMINIOS Y ATRIBUTOS

En general, los productos comerciales no implementan dominios, aunque sí se definen en el Modelo Relacional estándar.

- **Transformación de entidades:** cada entidad del esquema E/R se transforma en una relación cuya Clave Primaria es el Atributo Identificador Principal de la entidad.
- **Transformación de dominios:** Los dominios del esquema E/R se transforman en el mismo dominio en el esquema relacional aunque normalmente no se expresan en el grafo relacional.
- **Transformación de atributos:** Cada atributo de la entidad en el esquema E/R se corresponde con un atributo de la relación teniendo en cuenta las siguientes restricciones:
 - Atributos univaluados: dan lugar a un atributo de la relación.
 - Atributos multievaluados: dan lugar a una nueva relación cuya clave primaria es la concatenación de claves primarias de la entidad en la que está el atributo multivaluado. Los atributos multivaluados del diagrama E/R no pueden representarse como tales en el diagrama relacional (se produce una pérdida semántica en este sentido); para representarlos se debe crear una nueva relación con un nombre apropiado y los atributos siguientes: el AIP de la entidad que tiene el atributo multivaluado y el propio atributo multivaluado. La clave primaria de esta nueva relación será la combinación de estos dos atributos.
 - Atributos obligatorios: atributos con la restricción *Not null*. No permiten valores nulos; los atributos obligatorios no tienen una representación especial en el Diagrama Relacional; se representan igual que en el Diagrama E/R, es decir, solo con su nombre. En este sentido se comportan igual que los univaluados.
 - Atributos opcionales: permiten valores nulos. Están formados por el nombre atributo y el símbolo *.
 - Atributo identificador principal: se convierte en la clave primaria de la entidad. Aparece con el nombre subrayado continuo.
 - Atributo identificador alternativo: se transforman en atributos con la restricción *Unique*. Son las claves alternativas y aparecen con subrayado discontinuo.
 - Atributo compuesto: no hay forma de representar un atributo compuesto en el modelo relacional. Hay una pérdida semántica de este concepto. Una alternativa es representarlos como atributos simples univaluados; de ahí la pérdida semántica, ya que no se diferencian de este tipo de atributos.

5.2. TRANSFORMACIÓN DE INTERRELACIONES N:M

TRANSFORMACIÓN DE RELACIONES N:M

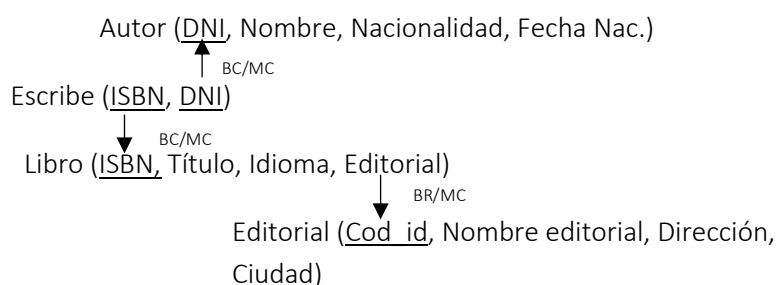
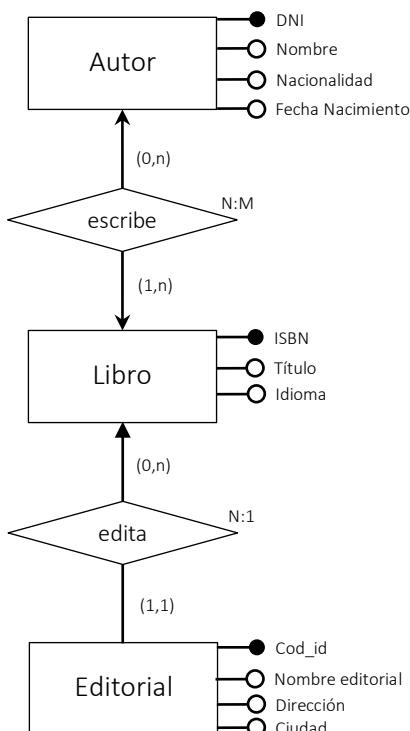
Las interrelaciones N:M dan lugar a una relación cuya clave primaria es la combinación de los atributos identificadores principales de las entidades que enlaza la interrelación. De esta forma, los atributos que forman la clave primaria de la nueva relación son claves ajenas respecto a las relaciones en las que estos atributos son clave primaria.

Si en la interrelación N:M existen atributos, estos pasarán a formar parte de la relación creada para la interrelación. Si existe un atributo multivaluado, dicho atributo debe formar parte de la Clave Primaria de la nueva relación, a no ser que denote una dimensión temporal (fechas, horas, etc.) en cuyo caso es necesario estudiar la semántica para decidir qué atributos componen la Clave Primaria.

TRANSFORMACIÓN DE RELACIONES 1:N

En interrelaciones 1:N hay dos posibilidades de transformación:

1. Crear una nueva relación para la interrelación con un tratamiento similar a las interrelaciones N:M con la diferencia de que, en este caso, la clave primaria está formada solo por el atributo identificador principal de la entidad que está en el lado N de la interrelación. Suele utilizarse cuando hay atributos en la interrelación 1:N. También se emplea esta opción si hay muchos valores nulos en la entidad que propagaría la clave.
2. La posibilidad segunda de transformación de una interrelación 1:N consiste en propagar el atributo identificador principal desde la entidad que se encuentra en el lado 1 a la entidad que se encuentra en el lado N como clave ajena (propagación de clave). Suele utilizarse cuando la interrelación 1:N no tiene atributos aunque en caso de que los tenga también pueden propagarse esos atributos a la relación del lado N.

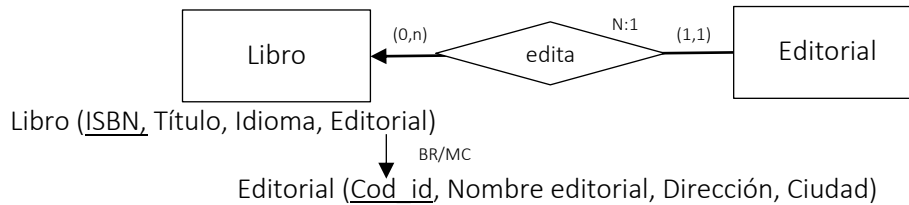


La interrelación *Edita* (1:N) podría dar lugar a una relación *Edita* (Libro, Editorial) si no se quisiera usar la opción de propagación de clave, aunque en este caso lo más lógico es usar propagación porque no hay atributos propios de la interrelación *Edita*, tampoco puede haber valores nulos en la entidad que propaga su clave (Editorial) porque todo Libro es Editado cuando es publicado.

5.3 ESTUDIO DE LA CARDINALIDAD MÍNIMA

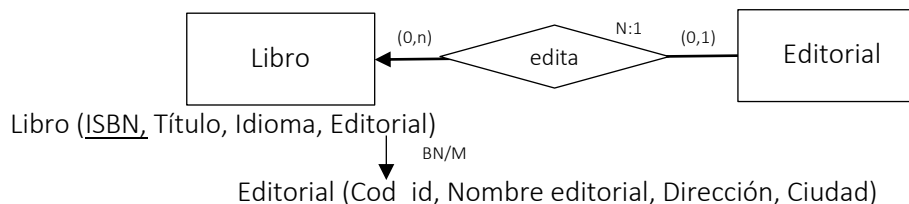
INTERRELACIONES 1:N – LADO 1

Si la cardinalidad mínima en el lado 1 es 1, la clave ajena no admitirá valores nulos (not null).



En interrelaciones 1:N y en las interrelaciones N:M es preciso estudiar las cardinalidades mínimas para no perder semántica en la transformación. El estudio de la cardinalidad permite determinar si la clave ajena puede o no admitir valores nulos.

Si la cardinalidad mínima en el lado 1 es 0, la clave ajena podrá admitir nulos (null).



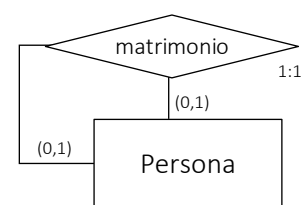
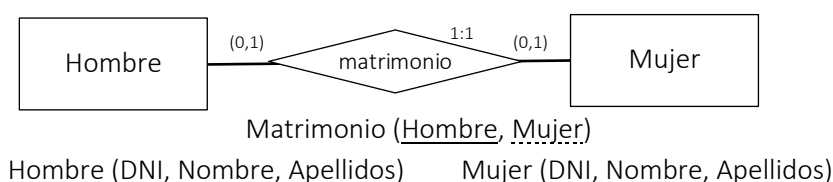
INTERRELACIONES 1:N – LADO N

Si la cardinalidad mínima en el lado N es 0, no se pierde semántica en la transformación. Si la cardinalidad mínima en el lado N es 1, se pierde semántica en la transformación; puede controlarse con aserciones o disparadores (triggers). Estos casos no pueden controlarse con la restricción de obligatoriedad (Not Null) de la clave ajena.

5.4 TRANSFORMACIÓN DE INTERRELACIONES 1:1

- Las interrelaciones 1:1 son un caso particular de las 1:N o N:M. En el caso de tratar con una, se puede proceder de dos maneras diferentes:
 - Crear una nueva relación (como en el caso N:M)
 - Propagar el Atributo Identificador Principal desde una de las entidades a la otra entidad en la que actuará como clave ajena (como se hace en el caso 1:N). La elección de la entidad que propaga su Atributo Identificador depende de la cardinalidad mínima.

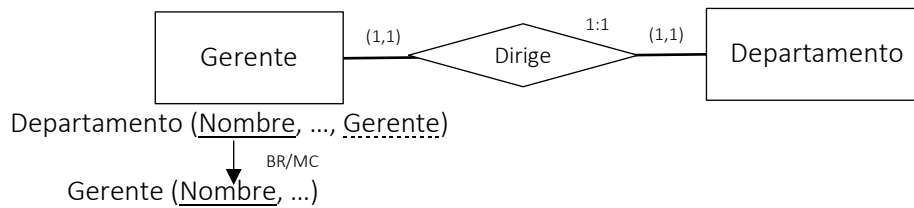
Un ejemplo de creación de nueva relación sería:



Representación actualizada y correcta del caso

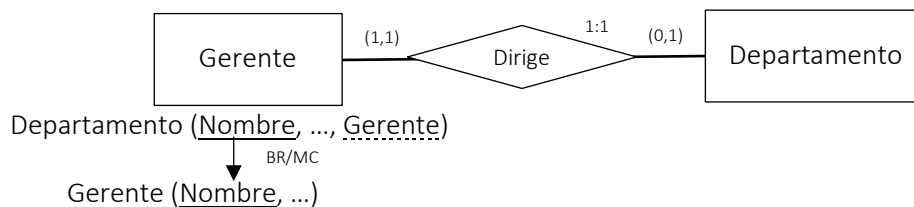
Aunque pueda considerarse un ejemplo políticamente incorrecto, en este ejemplo, es lógico crear una nueva relación porque al ser las cardinalidades mínimas 0 en los dos casos, es posible que haya muchos valores nulos ya que puede haber muchas personas que se queden solteras.

- Si las dos entidades tienen cardinalidad mínima 1, la elección es indiferente y la clave ajena no admite nulos (Not Null)



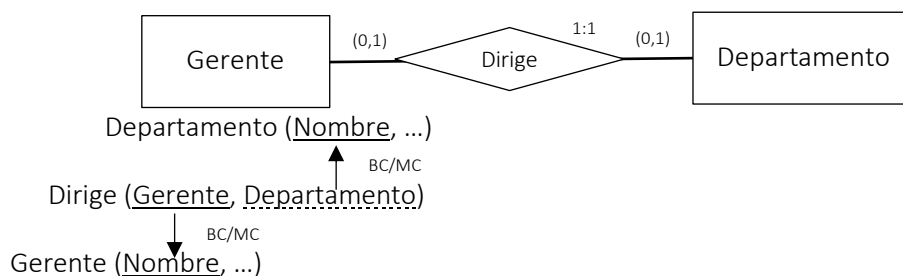
En este ejemplo, hay pérdida de semántica en el lado izquierdo, ya que todo gerente tiene que dirigir un departamento, y esto no se refleja en el Grafo Referencial; sería necesario utilizar Disparadores (Triggers).

- Si una entidad tiene cardinalidad mínima 0, se pasa su AIP como clave ajena sin admitir nulos (Not Null).



En este caso, no hay pérdida de semántica si se elige la entidad con la cardinalidad mínima 0. Si se cogiera la que tiene cardinalidad mínima 1, sí habría pérdida semántica.

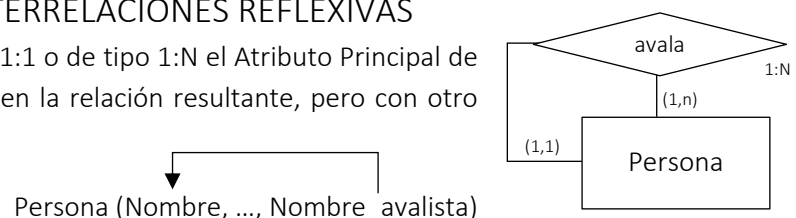
- Si las dos entidades tienen cardinalidad mínima 0 es conveniente crear una relación mínima de valores entre ellas.



Así evitaremos perder semántica.

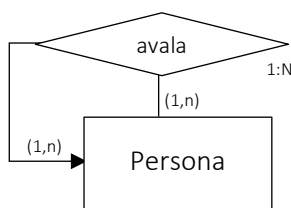
5.5 TRANSFORMACIÓN DE INTERRELACIONES REFLEXIVAS

En una interrelación reflexiva del tipo 1:1 o de tipo 1:N el Atributo Principal de la entidad aparece como clave ajena en la relación resultante, pero con otro nombre para poder distinguirla.

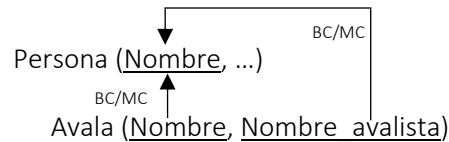


La transformación de una Relación Reflexiva no ofrece demasiados problemas salvo que, como es lógico, la clave ajena aparecerá en la misma relación. Si estamos transformando una relación 1:N. Si la relación es N:M tendremos que crear una nueva relación como ocurre siempre con las relaciones N:M.

- En una Interrelación Reflexiva del tipo N:M, al igual que ocurre con las binarias, siempre se genera una nueva relación.



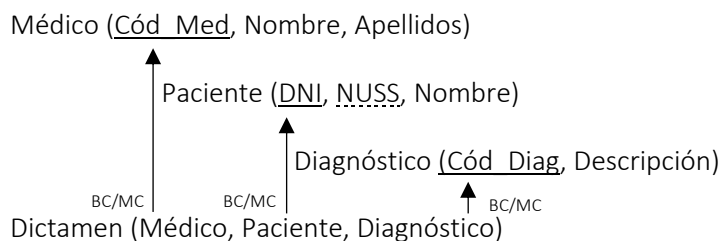
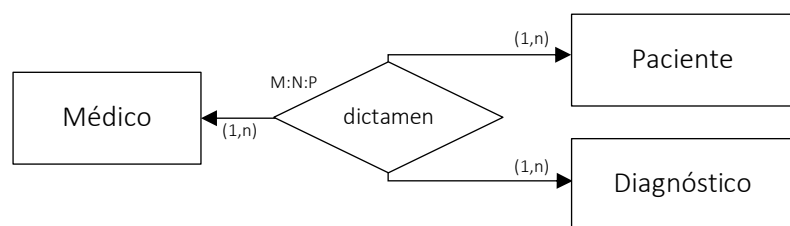
Como la relación reflexiva es del tipo N:M tenemos que construir una relación para representar en el Diagrama Relacional esta asociación N:M. La novedad es que las dos claves ajenas apuntarán a la misma relación ya que la asociación es reflexiva.



5.6 TRANSFORMACIÓN DE INTERRELACIONES DE GRADO MAYOR QUE 2

Una interrelación de grado mayor que 2 se transforma en una relación para cada entidad y una relación para la interrelación que asocia a las entidades. La Clave Primaria de la relación está formada por la concatenación de los atributos identificadores principales de las entidades que relaciona

En general, se deberá estudiar las cardinalidades mínimas de las relaciones para comprobar que no se pierde semántica en la relación

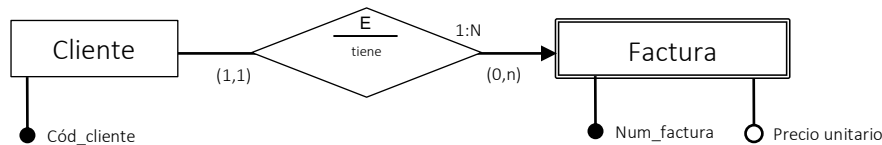


Si la interrelación de grado mayor que dos tuviera atributos, estos pasarían a formar parte de la nueva relación.

5.7 TRANSFORMACIÓN DE DEPENDENCIAS

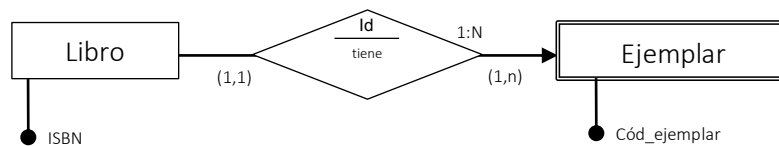
Una interrelación con dependencia de existencia origina que la clave ajena, propagada desde la entidad regular a la débil, deba tener la opción de borrado en cascada (BC). La clave ajena no forma parte de la clave primaria de la entidad débil.

En la transformación de una dependencia, tanto en existencia como en identificación, debemos tener en cuenta que este tipo de elemento plantea en primer lugar una asociación del tipo 1:N entre las entidades; por tanto, su transformación empezará aplicando la regla de transformación de las relaciones 1:N, es decir, se podrá crear una relación o se podrá propagar la clave. Lo normal es utilizar propagación de clave.



Se propaga la clave del Cliente a la Factura; en una dependencia en Existencia esta clave será simplemente clave ajena.

Transformación de dependencias en identificación: una origina que la clave ajena, propagada desde la entidad regular a la débil, deba tener la opción de borrado en cascada (BC). La clave ajena forma parte de la clave primaria de la entidad débil.

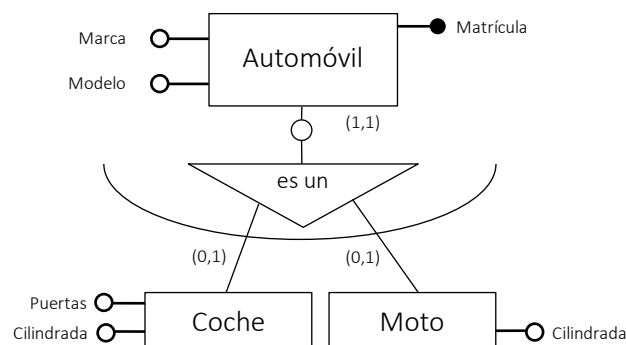


En una dependencia en Identificación, la clave propaga, además de ser clave ajena del Ejemplar, formará parte de la clave primaria del mismo. Esta es la diferencia entre una Dependencia en Identificación y una Dependencia en existencia

5.8 TRANSFORMACIÓN DE JERARQUÍAS

En la transformación de una jerarquía del modelo E/R al modelo Relacional se puede optar por tres estrategias:

- Englobar todos los atributos del supertipo y de los subtipos en una sola relación y añadir como atributo el atributo discriminante
- Crear una relación para el supertipo y una para cada uno de los subtipos con sus atributos correspondientes.
- Considerar relaciones distintas para cada subtipo que contengan además de los atributos propios, los atributos comunes.

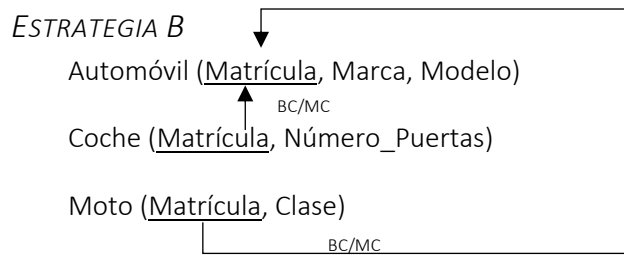


Estrategia A

Automóvil (Matrícula, Marca, Modelo, Puertas, Clase, Tipo)

Englobar todos los atributos del supertipo y de los subtipos en una sola relación a la que se añade como atributo el atributo discriminante: se suele usar cuando hay pocos atributos distintos en los subtipos y las interrelaciones en las que participan son las mismas. Es la mejor opción desde el punto de vista de la eficiencia. Si la jerarquía es total el atributo discriminante no permitiría valores nulos. Si la jerarquía

es parcial el atributo discriminante sí permitiría valores nulos. Si la jerarquía es exclusiva el atributo discriminante será univaluado y si es solapada será multivaluado.



Crear una relación para el supertipo y una para cada uno de los subtipos con sus atributos correspondientes: en este caso la clave primaria del supertipo aparece como clave primaria de los subtipos y también como clave ajena de los mismos; esta es la mejor opción desde el punto de vista semántico. Para representar la totalidad/parcialidad y el solapamiento/Exclusividad es necesario utilizar triggers, aserciones o llevar el control vía aplicación.

ESTRATEGIA C

Coche (Matrícula, Marca, Modelo, Núm_Puertas)

Moto (Matrícula, Marca, Modelo, Clase)

Considerar relaciones distintas para cada subtipo que tenga además de los atributos propios, los atributos comunes. Esta solución interesa cuando hay muchos atributos distintos de los subtipos y los accesos realizados afectan casi siempre a atributos comunes. Esta opción no permite representar jerarquías parciales.