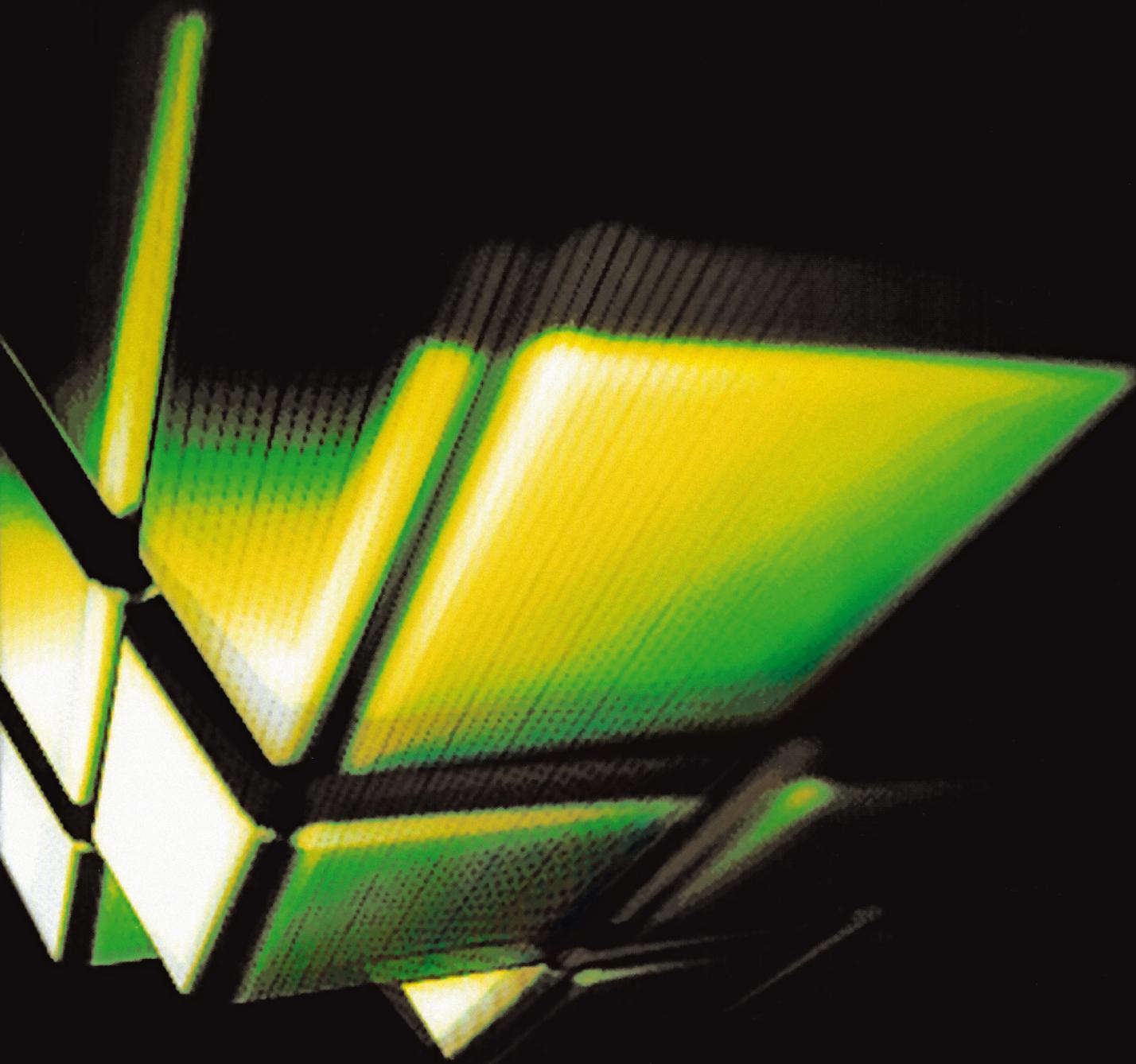


# Fundamentos de Diseño y Modelado de Datos



Francisco A. Morteo . Nicolás L. E. Bocalandro . Hernán G. Cascón  
Cristian A. Cascón . Christian D. Descalzo . Karina M. De Rosa . Diego Krauthamer



# **Fundamentos de Diseño y Modelado de Datos**

## **Primera Edición**

**Francisco A. Morteo**

Universidad de Buenos Aires  
Universidad Abierta Interamericana

**Nicolás L.E. Bocalandro**

Universidad de Buenos Aires  
Universidad Abierta Interamericana

**Cristian A. Cascón**

Universidad de Buenos Aires

**Hernán G. Cascón**

Universidad de Buenos Aires

**Christian D. Descalzo**

Universidad de Buenos Aires

**Karina M. De Rosa**

Universidad de Buenos Aires

**Diego Krauthamer**

Universidad de Buenos Aires



*Ediciones Cooperativas es un emprendimiento cooperativo de docentes de la Facultad de Ciencias Económicas de la Universidad de Buenos Aires para difundir sus trabajos e investigaciones.*

Ninguna parte de esta publicación, incluido el diseño de cubierta puede ser reproducida, almacenada o transmitida en manera alguna ni por ningún medio, ya sea electrónico, mecánico, óptico de grabación o de fotocopia sin permiso previo del Editor. Su infracción está penada por las leyes 11723 y 25446



Fundamentos de diseño y modelado de datos / Francisco A. Morteo, Nicolás L.E. Bocalandro...[et.al.].. - 2a ed. -  
Buenos Aires: Ediciones Cooperativas, 2007.  
230 p.; 28x21 cm.

ISBN 978-987-1246-51-9

1. Procesamiento de Datos. 2. Diseño de Software.  
CDD 005.3 CDD 657

2007 Francisco A. Morteo – Nicolás L.E. Bocalandro - Cristian A. Cascón - Hernán G. Cascón - Christian D. Descalzo - Karina M. De Rosa - Diego Krauthamer

Derechos exclusivos

© 2007 Ediciones Cooperativas  
Tucumán 3227, (1189)  
Ciudad Autónoma de Buenos Aires – Argentina  
(54 011) 4864 5520 / (15) 4198 5667 <http://www.edicionescoop.org.ar>  
✉ [info@edicionescoop.org.ar](mailto:info@edicionescoop.org.ar)

*Diseño de Cubierta:* Gabriela Castro ✉ [gabrielacastro@fibertel.com.ar](mailto:gabrielacastro@fibertel.com.ar)

*Diseño Interior:* Víctor Acuña, Guillermo Boces, Adrián de Rosas, Guillermo Ferioli.

*Web Hosting:* [www.com.ar](http://www.com.ar) ✉ [info@www.com.ar](mailto:info@www.com.ar)

Hecho el depósito que establece la ley 11.723

*Impreso y encuadrado por:*  
Imprenta Dorrego, Cap. Fed.  
2<sup>a</sup>. ed. Tirada: 100 ejemplares. Se terminó de imprimir en Agosto de 2007

Editorial asociada a:

IMPRESO EN ARGENTINA – PRINTED IN ARGENTINE



**Indice**

<b>I Dedicatorias .....</b>	9
<b>II Los Autores .....</b>	14
1 Francisco Agustín Morteo .....	14
2 Nicolás Leocadio Eduardo Bocalandro .....	15
3 Cristian Ariel Cascón .....	16
4 Hernán Gabriel Cascón .....	17
5 Christian Daniel Descalzo .....	18
6 Karina Mariana De Rosa .....	19
7 Diego Krauthamer .....	20
<b>III Prólogo .....</b>	21
1 Carlos Waldbott de Bassenheim .....	21
2 Juan María Ale .....	21
<b>IV Metodología de trabajo .....</b>	23
<b>V Cómo interpretar los símbolos de la bibliografía .....</b>	24
1 Consideraciones antes del abordaje .....	24
2 Grados de complejidad de la obra .....	25
<b>VI Introducción .....</b>	26
<b>A Aspectos generales .....</b>	27
1 Los Sistemas de Información y las Bases de Datos .....	28
2 Dato, información y conocimiento .....	28
3 Dato .....	28
4 Información .....	28
5 Conocimiento .....	30
6 Bases de datos .....	30
7 Modelos de datos .....	31
8 Sistema Objeto, Sistema de Información y Sistema de Datos .....	31
9 Fig. 1. Interrelación entre SO, SI y SD .....	32
10 Esquema de archivos tradicionales .....	33
11 Fig. 2. Esquema de archivos convencionales .....	33
12 Fig. 3. Relación Programas y archivos en un esquema convencional ..	34
13 Esquema de Bases de Datos .....	35
14 Fig. 4. Esquema de procesamiento con Base de datos .....	35
15 Fig. 5. Definición de datos en un esquema de BD .....	36
16 Tipos de Bases de Datos .....	37
17 Bases de datos jerárquicas .....	37
18 Fig. 6. Modelo de estructura jerárquica .....	37
19 Bases de datos de red .....	38
20 Fig. 7. Modelo de estructura jerárquica en red .....	38
21 Bases de datos relacionales .....	38
22 Bases de datos orientadas a objetos .....	38
23 Arquitectura de una Base de Datos .....	39
24 Fig. 8. Separación de niveles de abstracción de una BD .....	40
25 Independencia de Datos .....	40
26 Fig. 9. Independencia física de datos .....	40
27 Fig. 10. Independencia lógica de datos .....	41
<b>B Proceso de diseño de una base de datos .....</b>	43
1 Proceso de diseño de una base de datos .....	44
2 Ciclo de Vida de un SI .....	44
3 Fig. 1. Descomposición del proceso de diseño de los datos .....	45

4	Fig. 2. Separación de Problemas Infológico y Datalógico .....	47
5	Modelo de datos y Modelo Funcional .....	47
6	Fig. 3. Relación entre el Modelo Conceptual y el Modelo Funcional ....	48
<b>C</b>	<b>Enunciado de caso base .....</b>	<b>49</b>
1	Caso Aprendiendo .....	50
<b>D</b>	<b>Diseño Conceptual .....</b>	<b>53</b>
1	Modelo Entidad Relación básico .....	54
2	Relaciones .....	56
3	Fig. 1. Ejemplo de Relación .....	56
4	Fig. 2. Ejemplo de Relación Unaria .....	57
5	Fig. 3. Ejemplo de Relación Binaria .....	57
6	Fig. 4. Ejemplo de Relación Ternaria .....	57
7	Fig. 5. Ejemplo de relación con atributos .....	58
8	Fig. 6. Ejemplo de cardinalidad muchos a muchos .....	59
9	Fig. 7. Ejemplo de cardinalidad uno a muchos .....	59
10	Fig. 8. Ejemplo de cardinalidad muchos a uno .....	60
11	Fig. 9. Ejemplo de cardinalidad uno a uno .....	60
12	Fig. 10. Ejemplo de cardinalidad cero .....	61
13	Cardinalidad de relaciones unarias .....	61
14	Fig. 11. Ejemplo de cardinalidad unaria uno a uno .....	61
15	Fig. 12. Ejemplo de cardinalidad unaria uno a muchos .....	61
16	Entidad Débil .....	62
17	Fig. 13. Ejemplo de Entidad Débil .....	62
18	Mantenimiento de un historial .....	63
19	Fig. 14. Ejemplo de atributo multivaluado con respecto a precios .....	63
20	Fig. 15. Ejemplo de atributo multivaluado con respecto a precios povistos por mas de un proveedor .....	63
21	Notaciones Alternativas Diagrama Entidad Relación .....	66
22	Notación según Chen .....	66
23	Fig. 16. Notación según Chen .....	66
24	Notación alternativa 1 .....	66
25	Fig. 17. Notación alternativa 1 .....	66
26	Notación Alternativa 2 .....	67
27	Fig. 18. Notación alternativa 2 .....	67
<b>E</b>	<b>Modelo E-R Extendido .....</b>	<b>69</b>
1	Especialización .....	70
2	Fig. 1. Representación de una especialización .....	70
3	Fig. 2. Representación de una especialización con sus atributos .....	71
4	Generalización .....	72
5	Fig. 3. Representación de una generalización .....	72
6	Agregación .....	72
7	Fig. 4. Representación de una relación ternaria .....	72
8	Fig. 5. Representación de una relación cuaternaria .....	73
9	Fig. 6. Representación de una agregación .....	74
10	Fig. 7. Relaciones Clase-Subclase .....	75
11	Cuadro 1 .....	75
12	Fig. 8. Relaciones SuperClase-SubClase para una ocurrencia específica .....	76
13	Fig. 9. Esquema de clases con atributos específicos .....	77
14	Fig. 10. Ejemplo de Desunión .....	78
15	Fig. 11. Ejemplo Unión .....	79

<b>F Casos Prácticos Diseño Conceptual</b>	81
1  Caso 2K TRAINNING	83
2  Caso FARMACIA LA RURAL	84
3  Caso TODO LIBRO	85
4  Caso CHICHO PETS	86
5  Caso LA CLÍNICA	88
<b>G Resoluciones Casos Prácticos Diseño Conceptual</b>	89
1  Resolución Caso FARMACIA LA RURAL	91
2  Resolución Caso LA CLÍNICA	92
<b>H Diseño Lógico</b>	93
1 Cuadro 1	94
2 Proceso de Normalización	94
3 Fig. 1. Ejemplo de Factura	95
4 Cuadro 2	95
5 Cuadro 3	96
6 Primera Forma Normal (1FN)	97
7 Cuadro 4	97
8 Cuadro 5	97
9 Cuadro 6	98
10 Segunda Forma Normal (2FN)	98
11 Cuadro 7	99
12 Tercera Forma Normal (3FN)	100
13 Cuadro 8	100
14 Cuadro Resumen FN	101
15 Relaciones N a N	101
16 Fig. 2. Representación en el DER	102
17 Cuadro 9	102
18 Relaciones Recursivas	103
19 Cardinalidad 1 a 1	103
20 Fig. 3. Relación Recursiva	103
21 Cuadro 10	103
22 Cuadro 11	104
23 Cuadro 12	104
24 Cuadro 13	104
25 Cardinalidad 1 a N	105
26 Fig. 4. Relación Recursiva 1 a N	105
27 Cuadro 14	105
28 Cuadro 15	106
29 Histórico	106
30 Cuadro 16	106
31 Fig. 5. Prod y sus atributos	107
32 Fig. 6. Prod y sus atributos	107
33 Cuadro 17	108
34 Diagrama Lógico	108
35 Fig. 7. Diagrama lógico de Factura	109
<b>I Casos Prácticos Diseño Lógico</b>	113
1  Caso COMERCOM	114
2  Caso MAGIC FUN	115
3  Caso SOLAR COLLEGE	118
4  Caso TODO LIBRO	120
5  Caso SICON	121

6	<input checked="" type="checkbox"/>	Caso SOLARIUM TOSTADITOS .....	124
7	<input checked="" type="checkbox"/>	Caso SURTIDITO S.A. .....	127
8	<input checked="" type="checkbox"/>	Caso CHICHO PETS .....	130
9	<input checked="" type="checkbox"/>	Caso RENT A KAB .....	131
10	<input checked="" type="checkbox"/>	Caso ABC MOVIL .....	135
11	<input checked="" type="checkbox"/>	Caso TERRA CONTROL .....	137
12	<input checked="" type="checkbox"/>	Caso E-BOLSA .....	140
13	<input checked="" type="checkbox"/>	Caso BAUXITA & CO .....	143
14	<input checked="" type="checkbox"/>	Caso SUSODA .....	146
15	<input checked="" type="checkbox"/>	Caso CUKI S.R.L. .....	149
<b>J</b>	<b>Resolución de Casos Prácticos Diseño Lógico</b> .....		<b>153</b>
1	<input checked="" type="checkbox"/>	Resolución Caso COMERCOM .....	155
2	<input checked="" type="checkbox"/>	Resolución Caso MAGIC FUN .....	156
3	<input checked="" type="checkbox"/>	Resolución Caso SOLAR COLLEGE .....	157
4	<input checked="" type="checkbox"/>	Resolución Caso RENT A KAB .....	158
5	<input checked="" type="checkbox"/>	Resolución Caso TERRA CONTROL .....	160
6	<input checked="" type="checkbox"/>	Resolución Caso BAUXITA & CO .....	163
7	<input checked="" type="checkbox"/>	Resolución Caso CUKI S.R.L. .....	166
<b>K</b>	<b>Lenguaje de Definición de Datos</b> .....		<b>169</b>
1	Lenguaje de Definición de datos .....		170
2	CREATE .....		170
3	Cuadro 1 .....		171
4	Sintaxis .....		171
5	Restricción .....		172
6	Tipos de datos .....		172
7	Cuadro 2 .....		172
8	Restricciones (Constraints) .....		172
9	PRIMARY KEY .....		173
10	FOREIGN KEY .....		174
11	Cuadro 3 .....		175
12	Cuadro 4 .....		176
13	ALTER .....		178
14	Cuadro 5 .....		178
15	DROP .....		179
<b>L</b>	<b>Casos Prácticos de DDL</b> .....		<b>181</b>
1	<input checked="" type="checkbox"/>	Caso CUKI S.R.L. .....	182
2	<input checked="" type="checkbox"/>	Caso MAGIC FUN .....	183
3	<input checked="" type="checkbox"/>	Caso ABC MOVIL .....	184
4	<input checked="" type="checkbox"/>	Caso TERRA CONTROL .....	185
5	<input checked="" type="checkbox"/>	Caso SUSODA .....	186
6	<input checked="" type="checkbox"/>	Caso TODO LIBRO .....	187
7	<input checked="" type="checkbox"/>	Caso CHICHO PETS .....	188
8	<input checked="" type="checkbox"/>	Caso E-BOLSA .....	189
9	<input checked="" type="checkbox"/>	Caso SOLARIUM TOSTADITOS .....	190
<b>M</b>	<b>Resolución de Casos Prácticos de DDL</b> .....		<b>191</b>
1	<input checked="" type="checkbox"/>	Resolución Caso CUKI S.R.L. .....	193
2	<input checked="" type="checkbox"/>	Resolución Caso TERRA CONTROL .....	195

<b>N Uso de herramientas .....</b>	199
1 <b>DeZign .....</b>	200
1 Definir un nuevo proyecto .....	200
2 Definiendo el Diagrama .....	201
3 Creando las entidades .....	202
4 Definiendo las relaciones .....	203
5 Agregando Atributos .....	204
6 Definiendo las propiedades de los atributos .....	205
2 <b>DBDesigner .....</b>	206
1 Creando un Diagrama .....	206
2 Definiendo una nueva entidad .....	206
3 Definiendo una relación entre dos entidades .....	207
4 Agregando atributos a nuestras entidades, y definiendo sus .....	209
propiedades .....	
5 Cambiando las notación de las relaciones .....	210
6 Notación Tradicional .....	210
7 Notación EER (1,n) .....	211
<b>O Casos Adicionales .....</b>	213
1  Caso SALIENDO DEL PASO .....	214
2  Caso WEBSHOPING .....	216
3  Caso ELECTRO VENTAS .....	217
4  Caso LA TOALLITA LIMPIA .....	218
5  Caso DON PEINE .....	219
6  Caso VIAJE VELOZ .....	220
7  Caso Caso SU PEDIDO .....	221
8  Caso VAMOS A LA FERIA .....	222
9  Caso EL PUBLICISTA .....	223
<b>P Preguntas frecuentes .....</b>	225
1 ¿Cuántos tipos de claves primarias existen? .....	226
2 ¿Cuál es la cantidad máximas de atributos que pueden formar parte de una clave primaria o foránea? .....	226
3 ¿Las claves primarias puede ser cualquier atributo o conjunto de atributos? .....	226
4 ¿Es coherente que existan dos tablas que tengan como clave primaria los mismos atributos? .....	226
5 ¿Cuál es el sentido de generar una clave foránea? .....	226
6 ¿Puede existir más de una clave foránea para una misma tabla? .....	227
7 ¿Es necesario crear todas las restricciones de integridad (claves foráneas) que sean posibles sobre un atributo o conjunto de atributos? .....	227
8 ¿En que circunstancia es necesario crear claves foráneas compuestas? .....	227
9 ¿Las relaciones derivables son dibujadas en el diagrama lógico? .....	228
10 ¿Respecto a las restricciones de dominio (Check), puede una misma restricción afectar a más de un campo? .....	228
11 ¿Los campos que forman parte de la clave primaria se los debe declarar como Not Null? .....	228
12 ¿Es necesario declarar todos los campos que no forman parte de la clave primaria como Not Null? .....	228
13 ¿Se deben incluir en una tabla los campos calculables? .....	229
14 ¿Para que se utilizan las entidades recursivas? .....	229

15	¿Puede existir el caso de que una entidad se relacione con otra a través de dos atributos pero que uno sea parte de la clave primaria y el otro no? .....	229
16	¿En un diagrama de entidad relación pueden existir relaciones de muchos a muchos? .....	229
17	¿Cómo me doy cuenta en el diagrama lógico de si la relación es de uno a muchos o de muchos a muchos? .....	230
18	¿En un DER la cardinalidad de la relación utilizada para representar la recursividad es de uno a muchos o de uno a uno? .....	231
19	Tabla de históricos. ¿Cómo se debe definir las claves para una tabla en la cuál se desea llevar un histórico? .....	231
20	Una vez diseñado el modelo. ¿Cómo se hace para validarla? .....	232
21	¿Cómo se procede a borrar registros en tablas relacionadas? .....	232

## **I. Dedicatorias**

Queremos expresar un reconocimiento especial a nuestro profesor Juan Carlos Chervatín, creador y Profesor Titular de la materia Sistemas de Datos de la FCE de la UBA. Su permanente actitud docente y su insistencia sobre la necesidad de generar escritos fue el motor que permitió que este libro sea hoy una realidad.

Asimismo, queremos agradecer a las autoridades de la Facultad de Ciencias Económicas de la UBA y de Ingeniería de Sistemas de la Universidad Abierta Interamericana, como así también a los colegas, alumnos y conocidos con quien compartimos tantas horas diarias.

## **Los autores**

Además de lo ya formulado en el libro anterior quiero agregar un agradecimiento y abrazo fraternal a Antonio Rosselló, con quien en los últimos años hemos llevado adelante un proyecto profesional que me demostró que las utopías son posibles. Lo hago extensivo a los colegas con quienes compartí esos momentos: Gerardo Cabaleiro, Alexis Rastalsky, Claudio Delgado, Carlos Calvo y Alfredo Pérez.

También le hago llegar un agradecimiento especial a Juan Ale, por el prólogo que tan amablemente ha escrito y por haber compartido sus tan vastos conocimientos con paciencia y humildad.

Por último, me parece justo agradecerles a los chicos el haberme invitado a participar del proyecto. Han sido su paciencia, respeto, tolerancia y cariño las razones que me motivaron para que pudiéramos completar esta obra. Pero como lo importante son las personas con nombre propio, gracias Karina, Christian, Diego, Cristian, Hernán y Nico. Principalmente a estos dos últimos que con su perseverancia han conseguido movilizar el proyecto que se cristaliza en este libro.

Después de compartir con ellos tantos momentos vitales, tengo la convicción de que hemos integrado un grupo que se mantendrá unido más allá del tiempo y de la distancia (si no, miren el ejemplo de Christian, nuestro corresponsal italiano). Este libro no es más que una manifestación de este grupo de amigos, que en el trato cotidiano va generando, casi imperceptiblemente un crecimiento personal y de conjunto, en el que lo académico es solamente una muestra. Gracias....

**F.A.M.**

Hace tiempo que con Franco teníamos la idea de sacar una obra así, algo sencillo y a la vez útil. De esta forma hemos encarado una vez más la difícil tarea de escribir algo, lo cual no es para nada sencillo. La obra llegó a su punto final y con resultados mejores a los esperados.

Quiero dedicar la presente obra a los colegas con quienes comparto horas de mi vida, excelentes profesionales, de esta casa de estudio o de otras: Franco Morteo, Hernán y Cristian Cascón, Diego Krauthamer, Cristian Descalzo, Karina de Rosa, Andrés Kouvach, Amadeo Mazzuoli, Horacio Amorena, Marcelo Semería, Daniel Pandiani, Víctor Acuña, Rubén López, Martín Strupeni, Marcelo Mendoza, y muchos otros con los cuales intercambio opiniones y puntos de vista de este apasionante mundo de las tecnologías. Un saludo especial a Adrián Rivanera con quien compartí gratos momentos y es una lástima que ya no esté para seguir haciéndolo.

A toda mi familia, mi esposa Liliana, Susana y Buby, Ale, César y Cristina, y mis queridos sobrinos, que me acompañan en todo momento y con los cuales disfruto mucho el poco tiempo que estoy con ellos.

Por último a mis ex – compañeros laborales, de Techint, los cuales también son excelentes profesionales: Maximiliano Matyas, Julio Vener, Marcelo Peralta, Jorge Font, Gustavo Barbini, Sebastián Seitler, Andrés Zoppeletto, Sergio Olivieri, Mariano Quesada, Hugo Suárez, Ana Blanco, Victoria de Souza, Daniela Torres, Gabriel Santos, Nicolás Salinas, y Javier Quiroga. A todos ellos mis agradecimientos.

A las nuevas generaciones de profesionales por venir, hacia ellos está dirigido todo mi esfuerzo.

**N.L.E.B.**

En estas líneas quiero aprovechar para decirle simplemente GRACIAS a todas aquellas personas que de una u otra forma estuvieron siempre a mi lado y me brindaron lo mejor de si.

Antes que nada quisiera mencionar a Vicente e Inmaculada, mis abuelos, a los cuales no voy a olvidar nunca, ni a dejar de agradecerles el haberme marcado un rumbo en la vida.

A Florinda y a Paola, mi mamá y mi hermana, ojalá la vida me permita demostrarle cuanto las quiero.

A mi viejo que lo extraño mucho, a mis tíos y a mi primo que lo está cuidando.

Dora, Nelo, Lele, Laura, Sergio Vanesa, Fernanda y Martín, mis tíos y primos, los extraño muchísimo y no paso ni un solo día sin pensar en ellos y en el abrazo que les voy a dar cuando vuelva a la Argentina.

Andrés por haberme dado uno de los regalos mas importantes que recibí en mi vida, su familia, y a Teresa una de mis mamás postizas que desde el cielo nos cuida a Pablo y a mí.

A Marcela una persona muy especial que quiero mucho y por la cual se me llenan los ojos de lágrimas cuando me pongo a pensar en que no la tengo cerca.

Carlos, Zulema y Marcelo, gracias por todos esos momentos especiales y por saber que puedo contar con Uds. en los momentos difíciles, siempre.

Franco, a quien quisiera agradecerle el afecto, sus consejos, su guía y su confianza que marcaron en mi carácter y en mi vida profesional un cambio muy importante.

A Nico, Hernán, Cristian, Guille, Tonga, Paula, Guillermo, Massimiliano, Adriano, David y Valerio nada más y nada menos que mis amigos.

**C.D.D.**

Recuerdo aquellos días de estudiante en los que miraba atentamente a cada uno de mis profesores y soñaba con algún día poder llegar a estar frente de un curso. Hoy, gracias a Franco, quién confió en mí y me dio la oportunidad de formar parte de su equipo de trabajo, puedo decir que logré mi objetivo tan deseado.

Está hermosa tarea que es la docencia me ha dado la oportunidad de conocer a un grupo de personas maravillosas con quienes comparto semana a semana el dictado de las clases; ellos no sólo son mis colegas sino que por sobre todo mis amigos; me refiero a Francisco Morteo, Nicolás Bocalandro, Christian Descalzo, Karina De Rosa y Diego Krauthamer, como así también a alguien que es más que un amigo: mi hermano Cristian, con quien comparto la vocación por la docencia. A todos ellos quiero agradecerles el estar siempre conmigo en esta difícil, pero hermosa tarea que es la docencia, como así también por apoyarme y acompañarme en todos los aspectos de mi vida.

También es mi deseo dedicarle esta obra a mi familia, la cual es parte fundamental y necesaria en mi vida, quiero agradecerle a mi padre (Ángel), madre (María Rosa), hermanos (Cristian, Pablo y Vanina), abuelos (Palma y Narciso), tíos (César, Estella, Mario y Rubén), primos (Damián, Patricia, Alejandra, Carolina y Juan Ignacio) y a Lola. Gracias a todos ellos por el apoyo incondicional que me dan día a día como así también el cariño que me brindan.

Quiero hacer una mención muy especial a la mujer que me acompaña todos los días de mi vida, quien es conjuntamente con el resto de mi familia un pilar fundamental de mi vida, el amor que ella me brinda día a día es lo que me motiva a seguir adelante: gracias Analía por estar a mi lado.

Por último, dedico esta obra a Walter Triano un amigo que me apoyó y asesoró en esta tarea de escribir, a las generaciones de profesionales por venir y a mis amigos del barrio con quienes compartí parte de los momentos más hermosos de mi vida y aún sigo haciéndolo.

**H.G.C.**

De manera casi sorpresiva, hace ya más de dos años, he tenido la suerte de comenzar mi actividad como docente en la misma casa donde me formé y la cual me abrió las puertas sin pedir nada a cambio. Esta obra es solo una parte de todo lo que debo a la Universidad de Buenos Aires y a quienes me dieron esta maravillosa oportunidad.

Dedico la obra a aquellos amigos con los que día a día compartieron, comparten y compartirán esta vocación: Franco, Nicolás, Christian, Karina, Diego y en especial a Hernán (mi hermano) con el deseo de que la amistad que nos une, y sin la cual esto no hubiera sido más que un anhelo, perdure por siempre.

Hago extensible la dedicatoria a mi familia: Ángel, María Rosa, Pablo, Vanina, mi novia Marcela, mis tíos, primos, abuelos y abuelos, para la cual no me alcanzan las páginas de este libro para agradecerles el afecto brindado, la educación y principios infundidos tan necesarios para conducirme con honestidad y sanas convicciones en esta vida.

Por último dedico esta obra a mis amigos de la infancia: Sergio, Pablo, Gustavo, Martín, Leandro, Sebastián, y a mis compañeros del trabajo: Mariano, Darío, Osvaldo, Mariela, Silvia C., Silvia V., Bertrán, Diego, Andrés K., Andrés J., Laura, Claudio, Patricia, Santiago, Javier, Ale y Agustín, con quienes comparto una gran amistad y esta hermosa afición por las Bases de Datos.

**C.A.C.**

Ante todo quiero agradecer a mis colegas y compañeros de Cátedra por permitirme participar en la creación de este Libro.

Durante mi Carrera Universitaria tuve la oportunidad de conocer Docentes que realmente tienen la vocación y el placer por el dictado de clases entre ellos, Francisco Morteo, Nicolás Bocalandro, Hernán y Cristian Cascón. Este es uno de los motivos por los cuales decidí sumarme a este equipo ya que considero que esta materia constituye uno de los pilares más importantes de nuestra Carrera. Dedico este libro a todos Uds. y nuevamente les agradezco por todo!!!

También quiero dedicarle este libro a mi Familia: a mi padre quien siempre me apoyo durante la carrera y me alentó siempre. A mi madre y hermana por su paciencia y apoyo. También a Pablo y a su familia. A las personas que se llevan gran parte de mis sentimientos, a quienes fui conociendo a lo largo de la vida y están cuando las necesito, ellos son mis más queridos amigos y amigas: Claudia, Silvia, Verónica, Ariel, Luis, Javier, Juan, mis primas: Laura, Tamara, Carla, Ivanna.

A todos aquellos Colegas que me ayudaron y motivaron a terminar la carrera: Marcelo, Marcela, Diego, Jorge, Gabriel, Francisco, Laura y Analía.

Finalmente dedico este libro a todos los estudiantes, porque sin ellos nosotros no podríamos aprender a enseñar y a la facultad porque me dio la oportunidad de conocer a muchas personas y de lograr uno de mis objetivos más deseados.

**K.M.D.R.**

En primer lugar quería agradecer a mis padres, Mariana, Eduardo y a mi hermano Gastón, que en todo momento me ayudan y me motivan para que siga hacia adelante.

A todos mis docentes en especial al Dr. Francisco Morteo y Nicolás Bocalandro que son un ejemplo de dedicación y amor hacia la docencia, y que siempre me acompañan, sobre todo en los malos momentos. Igualmente no puedo dejar de nombrar a Mirta Verón, Pedro Basualdo, Fabián Martínez, Esteban Heredia, y Víctor Veriansky.

A mis compañeros de la agrupación Raúl, José, Pichula, Juan Manuel, Germán, Ulises, Santiago, Hernán, Ariel y Cristian que siempre me apoyan.

A mis compañeros de la carrera de sistemas Gustavo, Hernán, Christian, Karina y Jorge.

Al resto de las personas que me conocen.

**D.K.**

## II. Los Autores

**Francisco Agustín Morteo** nació en la Ciudad de Buenos Aires el 24 de diciembre de 1952.

Se graduó como Contador Público, como Licenciado en Administración y como Licenciado en Sistemas de información en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Fue Profesor Invitado al Seminario de Comunicación en Redes en la Facultad de Ingeniería de la Universidad Católica Boliviana

Dictó cursos de Postgrado de Auditoría y Control de los Sistemas de Información en las Facultades de Ciencias Económicas de las Universidades de Buenos Aires y Nacional de Cuyo, Mendoza.

Fue profesor en el curso de Postgrado de Auditoría de Sistemas en la sede Mar del Plata del Consejo Profesional de Ciencias Económicas de la Provincia de Buenos Aires.

Ha ganado el concurso de antecedentes y oposición de la materia Sistemas de Datos de la Facultad de Ciencias Económicas de la UBA, que lo convierte en profesor regular adjunto.

Es además profesor adjunto interino de la materia Auditoría y Control de los Sistemas de Información de la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Es profesor titular en materias vinculadas con a Base de Datos en la Universidad Abierta Interamericana, en donde tiene a su cargo el área de materias afines al tema.

Profesionalmente se ha desempeñado en el ámbito de la Administración Pública, habiendo sido designado por concurso en las jefaturas del Área de Microcomputadores y de Auditoría de Sistemas en la Dirección General Impositiva.

Ha tenido a cargo el sector de Tecnología Educativa de la Facultad de Ciencias Económicas de la Universidad de Buenos Aires habiéndose implementado durante su gestión el dictado a distancia de las materias Cálculo Financiero y Álgebra.

Fue Representante Técnico del GCBA en temas sobre los Sistemas de Información Geográfico de la ciudad y representante de la Dirección General de Rentas en la comisión de Seguimiento de la migración del Sistema Informático de Administración de Cobranzas.

Fue el responsable de llevar adelante la migración del Sistema de Cobranzas de la Dirección General de Rentas al ámbito del Banco de la Ciudad de Buenos Aires.

Ha sido consultor del BID especializado en la implementación de soluciones tecnológicas relacionadas con Sistemas de Recaudación y Fiscalización.

Tiene a su cargo el grupo de auditoría del Sistema Integrado de Administración de la Cobranza de la Dirección General de Rentas del Gobierno de la Ciudad de Buenos Aires.

**Nicolás Leocadio Eduardo Bocalandro** nació en la Ciudad de Concordia, Entre Ríos, el 4 de enero de 1973.

Se graduó como Licenciado en Sistemas de Información en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Es Profesor Adjunto Regular Concursado de la Universidad de Buenos Aires y Profesor Adjunto en la Universidad Abierta Interamericana en materias relacionadas a Tecnología Informática y Bases de Datos.

Dictó cursos sobre Software Libre para la Dirección General de Rentas y coordinó el área docente del mismo lugar.

Participó como disertante sobre orientación vocacional en diversas instituciones educativas, como invitado del departamento de Extensión Universitaria de UBA - FCE.

Realizó un Postgrado en Docencia Universitaria en la Facultad de Ciencias Económicas y una Maestría en Tecnología Informática en la Universidad Abierta Interamericana.

Participó en proyectos relacionados a tecnología en empresas de minería, telecomunicaciones, consumo masivo, gobierno, e integración y captura automática de datos, como soporte de sistemas, desarrollador de aplicaciones, administrador de redes, administrador de bases de datos, analista funcional y líder de proyecto.

Se desempeñó como Auditor Externo del Órgano de Control del Sistema Integrado de Administración y Cobranzas de la Dirección General de Rentas Ciudad de Buenos Aires.

En la actualidad se dedica a Business Intelligence, campo a través del cual incursionó desde un modelo impositivo fiscal para DGR, como líder de proyecto, en una siderurgia multinacional diseñando y poniendo en marcha soluciones de Business Intelligence.

En la actualidad continúa desarrollándose en este campo en una consultora dedicada a Business Intelligence, para una empresa minera multinacional.

**Cristian Ariel Cascón** nació en Lomas de Zamora, Buenos Aires, el 14 de febrero de 1979.

Se graduó como Licenciado en Sistemas de Información en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Es ayudante de cátedra de la materia Sistemas de Datos en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Dicta cursos de Administración de Base de datos Oracle a través de la secretaría de Extensión Universitaria de la Universidad Tecnológica Nacional Regional Buenos Aires.

Dictó cursos de informática y Nuevas Tecnologías para la Ciudad Autónoma de Buenos Aires en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

A lo largo de su carrera profesional participó en diversos proyectos relacionados con el desarrollo de aplicaciones, optimización de Base de datos y aplicaciones en empresas de servicios y telecomunicaciones.

En la actualidad se dedica a tareas de Administración de bases de datos relacionales y multidimensionales Oracle y SQL-Server en una empresa de telecomunicaciones.

**Hernán Gabriel Cascón** nació en Lomas de Zamora, Buenos Aires, el 19 de abril de 1977.

Se graduó como Licenciado en Sistemas de Información en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Es ayudante de cátedra de la materia Sistemas de Datos en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Dicta cursos de especialización en modelado y desarrolló de aplicaciones de sistemas de información a través de la secretaría de Extensión Universitaria de la Universidad Tecnológica Nacional.

A lo largo de su carrera profesional participó en diversos proyectos relacionados con el desarrollo de software para empresas de servicios, transporte, telecomunicaciones, consumo masivo, gobierno, etc., desarrollando tareas de desarrollo de aplicaciones, consultoría funcional y líder de proyecto.

**Christian Daniel Descalzo** nació en la Ciudad de Lanús, Provincia de Buenos Aires, el 5 de diciembre de 1973.

Se graduó como Licenciado en Sistemas de Información en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires y actualmente cursa el último año de Ingeniería Informática en la Universidad de la Sapienza a Roma.

Realizó un Postgrado en Docencia Universitaria en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Se desempeñó como ayudante de cátedra en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires en el curso de Sistema de Datos hasta diciembre de 2004.

Dictó cursos de informática para la Ciudad Autónoma de Buenos Aires en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

En marzo del 2005 se radicó en Italia donde se desempeñó como programador de base de datos en Telecom Italia y la Banca Intesa.

En la actualidad se dedica a la administración de base de datos, aplicativos y sistemas como DBA en la Poste Italiana en la sede central de Roma.

**Karina Mariana De Rosa** nació en la Ciudad de San Martín, Provincia de Buenos Aires, el 13 de junio de 1977.

Se graduó como Licenciada en Sistemas de Información en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires y actualmente se encuentra cursando la Maestría en Administración de Proyectos dictada por la Universidad Tecnológica Nacional.

Se desempeña como ayudante de cátedra en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires en el curso de Sistema de Datos y de Tecnología de la Información.

Realizo una pasantía rentada por la Facultad de Ciencias Económicas en la Dirección Nacional de Vialidad dentro del Área Control Presupuestario brindando soporte en el Armado y Generación de la Base de Datos del sector.

Se desempeño laboralmente en Empresas de Telecomunicaciones. Participo activamente en el Outsourcing realizado entre IBM-Telefónica brindando soporte a los aplicativos de Gestión y dictando cursos de capacitación referidos a Base de Datos e Informática.

En la actualidad se dedica a realizar tareas de Consultoría, planificación y seguimiento de requerimientos a desarrollar para una empresa de telecomunicaciones y se encuentra capacitándose en SAP.

**Diego Krauthamer** nació en la Ciudad de Buenos Aires, el 28 de diciembre de 1977.

Se graduó como Licenciado en Sistemas de Información de las Organizaciones en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Es Auxiliar docente de la Universidad de Buenos Aires en las Materia Tecnología de la Información y Sistemas de Datos.

Se encuentra cursando un Postgrado en Docencia Universitaria en la Facultad de Ciencias Económicas. Asimismo también fue auxiliar de las materias Tecnología de los Computadores y Teoría de los Lenguajes y Sistemas Operativos.

Participa activamente en el consejo académico asesor de la Licenciatura en Sistemas de Información de las Organizaciones y también es invitado como disertar sobre temas de las tecnologías de la información y las comunicaciones (Tic's) en diversas organizaciones educativas.

Participó como consultor en proyectos relacionados a Desarrollo de Sistemas de información Geográfica, Análisis y Control de Sistemas, Administrador de redes, Soporte Técnico y Mesa de ayuda en Organizaciones privadas y Organismos públicos.

En la actualidad se desempeña como asesor en temas informáticos de la Dirección General de Industria, Comercio y Servicios del Gobierno de la Ciudad de Buenos Aires.

### **III. Prólogo**

"Fundamentos de Diseño y Modelado de Datos" es una obra clara y amena para introducir al lector en los conceptos de bases de datos y en la modelización y diseño. Su abundante ejercitación sobre los temas tratados le da un mayor valor aún, para comprender real y acabadamente cada uno de los temas.

La importancia de las Bases de Datos para el cumplimiento de los objetivos organizacionales, hacen que un adecuado diseño de las mismas sea un componente indispensable para lograrlos.

En mi carácter de Profesor Titular de Sistemas de Datos y Director del Departamento de Sistemas resulta un verdadero orgullo que este destacado grupo de docentes y egresados de la Licenciatura en Sistemas de Información de las Organizaciones de la Facultad de Ciencias Económicas de la UBA, haya plasmado esta excelente publicación que, sin duda, resultará un valioso material para estudiantes y profesionales. En consecuencia, deseo felicitar a los autores por su generosa contribución bibliográfica.

**Carlos Waldbott de Bassenheim**

Profesor Titular Interino de Sistemas de Datos de la FCE-UBA  
Director del Departamento de Sistemas de la FCE-UBA

Las bases de datos, a través de las computadoras, están instaladas en nuestras vidas y constituyen el núcleo de cualquier sistema de información. Continuamente surgen grandes cantidades de datos que necesitamos almacenar y consultar. Sin embargo, la tarea de organizar el almacenamiento de esos datos no es trivial. Esto es lo que se denomina diseño de bases de datos y como toda tarea de diseño, es en parte un arte y en parte una disciplina científica. Por este motivo, para diseñar bases de datos eficaces y eficientes, se requiere no solo conocimiento formal sino también una extensa práctica.

Esta obra utiliza un enfoque práctico, basado en el método de "Análisis de Casos" y representa una interesante propuesta pedagógica para un curso de bases de datos, con foco en el diseño.

Si bien existen muchos excelentes libros sobre bases de datos en general, pocos tratan específicamente el tema crucial del diseño. Este libro está dedicado, especialmente, al diseño de bases de datos, en particular bases de datos relacionales.

Comienza con un capítulo sobre aspectos generales de las bases de datos y otro sobre el proceso de diseño, donde brinda el marco de la metodología a utilizar: el diseño conceptual, el lógico y, finalmente, el físico.

A partir de allí, un capítulo se dedica al diseño conceptual, donde se introduce como formalismo de representación el Modelo Entidad-Relación, y otro a ejemplos que completamente desarrollados ilustran esta etapa del diseño.

A continuación, un capítulo describe el diseño lógico, donde el formalismo es el modelo relacional, incluyendo el subproceso de normalización. Este capítulo es acompañado por otro con numerosos ejemplos.

A partir de allí, el lector podrá encontrar uno de los aspectos de mayor valor agregado de esta obra: un gran número de casos para desarrollar, con sus respectivas soluciones.

De la lectura de este libro se beneficiarán no sólo los estudiantes de Sistemas e Informática, sino también profesionales que deseen refrescar o perfeccionar sus conocimientos de diseño de bases de datos. Los autores, el equipo de la cátedra liderada por Franco Morteo en la Facultad de Ciencias Económicas, han realizado un excelente trabajo, basado en la experiencia de Franco de muchos años, tanto en la actividad profesional como en la docencia universitaria, específicamente en el área de Sistemas.

**Juan M. Ale**  
Profesor Asociado Regular  
Cátedra Bases de Datos  
Facultad de Ingeniería – UBA

#### **IV. Metodología de trabajo**

El objetivo del libro es abarcar aspectos conceptuales vinculados con el diseño de Bases de Datos, pero abordados de manera que puedan ser trabajados por el lector de una manera práctica.

Para cumplir con dicho objetivo hemos estructurado la obra de la siguiente manera:

- ✓ Se adopta como metodología de estudio la modalidad de "Análisis de Casos".
- ✓ Se basa en el desarrollo de los temas tomando como guía un caso base llamado "Aprendiendo", que se presenta en el capítulo C.
- ✓ Se comienza cada capítulo brindando los elementos conceptuales necesarios para comprender el tema.
- ✓ Se introducen casos reales adaptados, simplificados y reestructurados, como para dar más realismo a las situaciones planteadas. Pretendemos con esto que el lector se familiarice con la terminología y tenga un primer contacto con el tema.
- ✓ Sobre el final de cada capítulo se procede a resolver el punto correspondiente del caso "Aprendiendo".
- ✓ Se proponen enunciados de casos para que sean trabajados por los lectores. Algunos de ellos ya se encuentran resueltos para que sirvan de referencia a la compresión y resolución de los casos. Todos ellos están ordenados según su complejidad, para lo cual recomendamos ir abarcándolos desde los menos complejos a los más complejos.
- ✓ La resolución de los mismos, o cualquier consulta relacionada con el tema pueden ser giradas a la dirección de email [disenioymodelado@abcbasesdedatos.com.ar](mailto:disenioymodelado@abcbasesdedatos.com.ar) que ha sido creada exclusivamente con ese fin.
- ✓ Los ejercicios del capítulo vinculado con el tema de definición de datos ha sido resuelto con SQL ANSI, de manera que el lector pueda plantear la solución en cualquier SGBD que posea.
- ✓ Asimismo, se presentan ejemplos de herramientas de diseño tales como RFlow, DeZign y DBDesigner, versiones que han sido descargadas para su prueba y evaluación, y que pueden ser encontrados en . El objetivo del uso de estas herramientas no es a los fines de su difusión sino el de lograr una permanencia en los aprendizajes alcanzados, cuestión que se termina de lograr cuando el lector interactúa con la realidad, y se enfrenta a un problema concreto con una herramienta específica. Este sería nuestro mayor deseo.
- ✓ Se propone una lista de preguntas que consideramos de gran utilidad sobre las cuales se pueden despejar las dudas más recurrentes sobre los temas tratados.
- ✓ Se utilizará el sitio [www.abcbasesdedatos.com.ar](http://www.abcbasesdedatos.com.ar) el cual posee material para descargar, tal como solución a ejercicios, enunciados, presentaciones, y links a bibliografía recomendada de lectura, entre otros.

## V. Cómo interpretar los símbolos de la bibliografía

### Consideraciones antes del abordaje de la lectura

Dado que la obra pretende enseñar y desarrollar por parte del lector determinadas habilidades hemos tenido ciertos recaudos en su confección.

Para ubicar al lector en cada una de las actividades planteadas se utilizarán símbolos representativos de cada una de ellas, a saber:

- ✓ **Bibliografía vinculada al tema tratado.** Esta bibliografía permite profundizar determinados temas vistos en la presente obra. Le permiten al lector evacuar dudas y ampliar aspectos sobre temas específicos.
- ✓ **Aclaraciones que consideramos importantes.** Permiten advertir sobre posibles confusiones, que según nuestra experiencia pueden despertar dudas o son complejas de entender.
- ✓ **Ejercitación a realizar por cuenta propia.** El propósito es fijar los conocimientos adquiridos y encarar el proceso de aprendizaje por cuenta propia.

Estos iconos o imágenes serán encontrados a lo largo de la bibliografía:

	<b>Bibliografía relacionada al tema.</b> Cuando este ícono es mostrado nos referiremos a bibliografía adicional que amplía y profundiza el tema planteado.
	<b>Aclaraciones consideradas importantes.</b> Este ícono se refiere a aclaraciones y tiene por objetivo advertir sobre posibles confusiones, despejando dudas o impartiendo un punto de vista propio sobre cómo despejar dichas dudas.
	<b>Ejercitación a realizar por cuenta del lector.</b> Son actividades pensadas para que Ud. adquiera cierta destreza, fijando conocimientos a través de la práctica, que le permitirán ejercitarse y aprender en base a los errores cometidos. Cada vez que vea éste ícono, deténgase y efectúe dicha práctica. Es pedagógicamente muy importante detenerse en estos puntos y realizar la práctica solicitada. Tenga en cuenta que a medida que avanzamos la complejidad va en aumento.

Por otra parte hemos utilizado casos ficticios, pero que pretenden mostrar la realidad en toda su riqueza, enfrentando al lector con situaciones reales. De esta forma no pretendemos que se aprenda lo que se enseña directamente sino que se haga a través de un proceso complejo de reestructuración de los saberes previos individuales, implicando la reconstrucción, la contrastación

y la resignificación. De esta forma el docente es un mediador entre los contenidos y los alumnos, actuando como facilitador y orientador, y permitiendo además los intercambios educativos generados en el aula.

Creemos que una forma muy valida de aprendizaje es el trabajo en grupo, ya que, por un lado los siguientes motivos: permiten una integración e interacción adecuada que posibilitan vencer ciertas dificultades (aburrimiento, desgano, indiferencia, incomprendión, ansiedad) y por el otro logran movilizar afectos acrecentando la creatividad, confrontando ideas, y opiniones, construyen vínculos afectivos, permitiendo la cooperación común.

El uso de casos permite describir una determinada situación de la realidad, es decir que a partir de la redacción se seleccionan los datos considerados relevantes para su desarrollo posterior.

Esta técnica favorece el desarrollo de la capacidad para diagnosticar y resolver problemas y formular cursos de acción para tomar decisiones a posteriori. Dado que la resolución del caso exige por parte de quien lo resuelve determinado nivel de interpretación, comprensión y resolución. El intercambio de opiniones también estimula la comunicación grupal y la motivación.

De esta forma tratamos de habituar a quien aprende a que se puede encontrar diversas opciones frente a un determinado problema, y cambiar la idea de que sólo existe una única manera posible de resolver las cosas y que generalmente es rutinaria y tradicional. De esta forma se permite contrastar y comparar diversas soluciones y encontrar distintos puntos de vista del mismo problema, enriqueciendo el aprendizaje. Esto se condice mas con la realidad de las organizaciones en las cuales la resolución de problemas y procesos decisarios no se basan en una selección individual de una única opción.

De esta forma se logra que quien aprende desarrolle capacidades, se internalicen actitudes y se adquieran destrezas para analizar y resolver problemas complejos de la realidad.

#### **Grados de complejidad de la obra**

Finalmente el abordaje de casos esta ordenado según el grado de dificultad.

	Fácil
	Regular
	Intermedio
	Complejo
	Muy Complejo

## **VI. Introducción**

Como cuerpo docente del curso de Sistemas de Datos de la Facultad de Ciencias Económicas de la UBA nos hacemos presente nuevamente con una publicación que tiene identidad propia pero que, al abordar la temática específica del curso, puede considerarse como complemento de la anterior relacionada con SQL.

Haciendo un juego de palabras, podemos considerarla una “continuidad previa”, en el sentido de que es un segundo libro con temas del programa (continuidad), pero que por las características del contenido es aconsejable que sea tratado con anterioridad al previo escrito sobre SQL.

Pese a ser un libro que en principio nace con fines académicos, no está dirigido exclusivamente a los alumnos. Comprendimos que con un mínimo esfuerzo adicional podíamos realizar una obra para aquellos que estén interesados y necesiten tener una guía a la vez sencilla y práctica sobre la metodología para diseñar una Base de Datos.

Por esta razón es que podemos decir que, además de tener una visión académica, es lo suficientemente práctica como para que sea comprendida por un público más amplio (profesionales que necesitan sistematizar sus actividades, adolescentes que necesitan mantener un archivo ordenado de sus CD's, etc.).

Somos conscientes de que el alcance del presente libro de ninguna manera abarca la totalidad de temas, ni la profundidad con que cada uno puede ser tratado. Sin embargo, hemos preferido publicarlo, en la creencia de que lo óptimo es enemigo de lo bueno.

Decía Borges que una de las razones que justifican publicar, es no tener que corregir más. Sin pretender contradecir lo dicho, el dar a luz este libro nos permitirá transferir la tarea de corrección a nuestros lectores y alumnos, como ya lo han hecho con anterioridad. Vaya para ellos mi agradecimiento. Con su colaboración podremos realizar nuevas ediciones mejoradas y profundizadas, Mientras tanto va ésta con la certeza de que sabrán perdonar los errores.

# A

## Aspectos Generales

---

### Objetivos

- ✓ Adquirir los conceptos de “dato”, “información”, “conocimiento” y “sistemas”.
- ✓ Comprender las diferencias entre “sistema objeto”, “sistema de información” y “sistema de datos”.
- ✓ Identificar las diferencias, ventajas y limitaciones entre un sistema de archivos tradicionales y un Sistema de Base de datos.
- ✓ Entender y definir el concepto de Base de Datos y su diferencia con un “Sistema de Gestión de Base de Datos”.

## A. Aspectos generales

### Los Sistemas de Información y las Bases de Datos

#### Dato, información y conocimiento<sup>1</sup>

Como existen ciertos conceptos que tienen interpretaciones diferentes según distintos autores, comenzaremos definiendo algunos de ellos, para aclarar el real alcance que pretendemos darle.

#### Dato

Un dato es un conjunto discreto, de factores objetivos sobre un hecho real. Dentro de un contexto empresarial, el concepto de dato es definido como un registro de transacciones. Un dato no dice nada sobre el porqué de las cosas, y por sí mismo tiene poca o ninguna relevancia o propósito.

Las organizaciones actuales normalmente almacenan datos mediante el uso de tecnologías. Desde un punto de vista cuantitativo, las empresas evalúan la gestión de los datos en términos de costo, velocidad y capacidad.

Todas las organizaciones necesitan datos y algunos sectores son totalmente dependientes de ellos. Bancos, compañías de seguros, agencias gubernamentales y la Seguridad Social son ejemplos obvios. En este tipo de organizaciones la buena gestión de los datos es esencial para su funcionamiento, ya que operan con millones de transacciones diarias. Pero en general, para la mayoría de las empresas tener muchos datos no siempre es bueno. Las organizaciones almacenan datos sin sentido. Realmente esta actitud no tiene sentido por dos razones. La primera es que demasiados datos hacen más complicado identificar aquellos que son relevantes. Segundo, y todavía más importante, es que los datos no tienen significado en sí mismos.

Los datos describen únicamente una parte de lo que pasa en la realidad y no proporcionan juicios de valor o interpretaciones; por lo tanto no son orientativos para la acción. La toma de decisiones se basará en datos, pero estos nunca nos dirán qué hacer. Los datos no dicen nada acerca de lo que es importante o no. A pesar de todo, los datos son importantes para las organizaciones, ya que son la base para la creación de información.

#### Información

Como han hecho muchos investigadores que han estudiaron el concepto de información, lo describiremos como un mensaje, normalmente bajo la forma de un documento o algún tipo de comunicación audible o visible. Como cualquier mensaje, tiene un emisor y un receptor. La información es capaz de cambiar la forma en que el receptor percibe algo, es capaz de impactar sobre su juicio de valor y comportamiento. Tiene que informar; son datos que marcan la diferencia. La palabra "informar" significa originalmente "dar forma a" y la información es capaz de formar a la persona que la consigue, proporcionando ciertas diferencias en su interior o exterior. Por lo tanto,

---

<sup>1</sup> Davenport y T. Prusak, L. "Working Knowledge: How Organizations Manage What They Know". Harvard Business School Press. 1998.

estrictamente hablando, es el receptor, y no el emisor, el que decide si el mensaje que ha recibido es realmente información, es decir, si realmente lo informa. Un informe lleno de tablas inconexas, puede ser considerado información por el que lo escribe, pero a su vez puede ser juzgado como "ruido" por el que lo recibe.

En este sentido para que la información sea eficiente es necesario que reúna una serie de requisitos, planteados por Saroka<sup>2</sup> en los siguientes términos:

- ✓ Economía. El costo de producir información debe ser menor a los beneficios que se obtengan de ella.
- ✓ Oportunidad. La información debe estar disponible en el momento y con la frecuencia que se requiera.
- ✓ Utilidad. Toda salida de un sistema de información debe satisfacer una necesidad concreta.
- ✓ Comparabilidad. La información debe ser comparable en el espacio, el tiempo y el alcance.
- ✓ Flexibilidad. Todo sistema de información debe ser adaptable a los cambios del sistema objeto.
- ✓ Claridad. La información debe atender un nivel intelectual y técnico del destinatario.
- ✓ Confidabilidad. La información debe ser lo suficientemente confiable como para tomar decisiones basadas en ella.

La información se mueve en las organizaciones a través de redes formales e informales. Las formales tienen una infraestructura visible y definida: cables, buzones de correo electrónico, direcciones, etc. Los mensajes que estas redes proporcionan incluyen email, servicio de entrega de paquetes, y transmisiones a través de Internet. Las redes informales son invisibles. Se hacen a medida. Un ejemplo de este tipo de red es cuando alguien envía una nota o una copia de un artículo con las siglas "FYI" (For Your Information).

A diferencia de los datos, la información tiene significado (relevancia y propósito). No sólo puede formar potencialmente al que la recibe, sino que está organizada para algún propósito. Los datos se convierten en información cuando su creador les añade significado. Transformamos datos en información añadiéndoles valor en varios sentidos. Hay varios métodos:

- ✓ Contextualizando: sabemos para qué propósito se generaron los datos.
- ✓ Categorizando: conocemos las unidades de análisis de los componentes principales de los datos.
- ✓ Calculando: los datos pueden haber sido analizados matemática o estadísticamente.
- ✓ Corrigiendo: los errores se han eliminado de los datos.
- ✓ Condensando: los datos se han podido resumir de forma más concisa.

Los ordenadores nos pueden ayudar a añadir valor y transformar datos en información, pero es muy difícil que nos puedan ayudar a analizar el contexto de dicha información. Un problema muy común es confundir la información (o el conocimiento) con la tecnología que la soporta. Desde la televisión a Internet, es importante tener en cuenta que el medio no es el mensaje. Lo que se intercambia es más importante que el medio que se usa para hacerlo. Muchas veces se comenta que tener un teléfono no garantiza mantener conversaciones brillantes. En definitiva, que

---

<sup>2</sup> Saroka, Raúl. "Sistemas de Información en la era digital". Fundación OSDE. 2002.

actualmente tengamos acceso a más tecnologías de la información no implica que hayamos mejorado nuestro nivel de información.

### **Conocimiento**

La mayoría de la gente tiene la sensación intuitiva de que el conocimiento es algo más amplio, más profundo y más rico que los datos y la información.

Vamos a intentar realizar una primera definición de conocimiento que nos permita comunicar que queremos decir cuando hablamos de conocimiento dentro de las organizaciones.

El conocimiento es una mezcla de experiencia, valores, información y **saber hacer** que sirve como marco para la incorporación de nuevas experiencias e información, y es útil para la acción. Se origina y aplica en la mente de los conocedores. En las organizaciones con frecuencia no sólo se encuentra dentro de documentos o almacenes de datos, sino que también está en rutinas organizativas, procesos, prácticas, y normas.

Lo que inmediatamente deja claro la definición es que ese conocimiento no es simple. Es una mezcla de varios elementos; es un flujo al mismo tiempo que tiene una estructura formalizada; es intuitivo y difícil de captar en palabras o de entender plenamente de forma lógica. El conocimiento existe dentro de las personas, como parte de la complejidad humana y de nuestra impredecibilidad. Aunque solemos pensar en activos definibles y concretos, los activos de conocimiento son mucho más difíciles de manejar. El conocimiento puede ser visto como un proceso (flujo) o como un stock.

El conocimiento se deriva de la información, así como la información se deriva de los datos. Para que la información se convierte en conocimiento, las personas deben hacer prácticamente todo el trabajo. Esta transformación se produce gracias a:

- ✓ Comparación.
- ✓ Consecuencias.
- ✓ Conexiones.
- ✓ Conversación.

Estas actividades de creación de conocimiento tienen lugar dentro y entre personas. Al igual que encontramos datos en registros, e información en mensajes, podemos obtener conocimiento de individuos, grupos de conocimiento, o incluso en rutinas organizativas.

### **Bases de datos**

Una base de datos es un conjunto de datos que pertenecen al mismo contexto, almacenados sistemáticamente para su uso posterior. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

Las bases de datos almacenan, como su nombre dice, datos. Estos datos son representaciones de sucesos y objetos, a diferente nivel, existentes en el mundo real: en su conjunto, representan algún tipo de entidad existente. En el mundo real se tiene percepción sobre las entidades u objetos y sobre los atributos de esos objetos; en el mundo de los datos, hay registros de eventos y datos de eventos. Además, en ambos escenarios se puede incluso distinguir una tercera faceta: aquella que comprende las definiciones de las entidades externas, o bien las definiciones de los registros y de los datos.

En informática existen los sistemas gestores de bases de datos (**SGBD**), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

### **Modelos de datos**

Un modelo es la representación de la realidad, descripta mediante un determinado formalismo. En términos de Codd,

``Un modelo de datos es una combinación de tres componentes:

- ✓ una colección de estructuras de datos (los bloques constructores de cualquier base de datos que conforman el modelo);
- ✓ una colección de operadores o reglas de inferencia, los cuales pueden ser aplicados a cualquier instancia de los tipos de datos listados en (1), para consultar o derivar datos de cualquier parte de estas estructuras en cualquier combinación deseada;
- ✓ una colección de reglas generales de integridad, las cuales explícita o implícitamente definen un conjunto de estados consistentes --estas reglas algunas veces son expresadas como reglas de insertar-actualizar-borrar."

Uno de los fines de un SGBD es el mantenimiento de un modelo abstracto de datos en grado de representar la realidad a la cual hacen referencia estos datos y de mantener la independencia entre el aspecto formal de la base de datos y su efectiva implementación física.

Una vez que se tiene la percepción de la realidad, se procede a la descripción de la misma, dejando de lado los aspectos particulares que no son relevantes al fin del modelo y si utilizan aquellas estructuras que el modelo pone a nuestra disposición.

Los varios tipos de modelos sirven para simplificar la proyección de la base de datos.

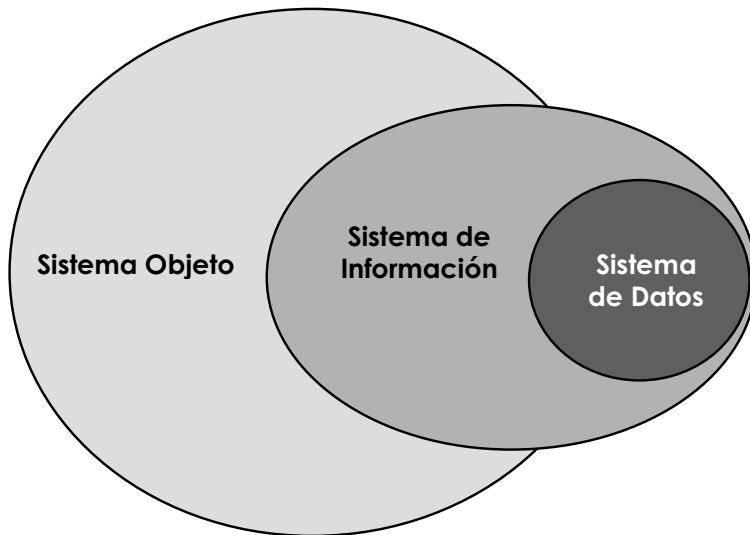
En el proceso de diseño de una BD existen distintos modelos que tienen fines diferentes. Los mismos serán analizados en el capítulo siguiente.

### **Sistema Objeto, Sistema de Información y Sistema de Datos**

La implementación de un sistema de información requiere de una serie de elementos que permitan realizar un modelo del sistema real que se quiere implementar. Algunos de ellos son dinámicos, por ejemplo los programas que representan el accionar del sistema; y otros en cambio son estáticos como por ejemplo los símbolos necesarios para representar aspectos de la realidad que se intenta modelizar.

Estos conceptos han sido tratados por Langefors<sup>3</sup>, quien distingue 3 tipos de sistemas:

- ✓ El Sistema Objeto (SO). Es aquel al que un sistema de información le provee servicios. Los elementos que lo componen corresponden al mundo real. Por ejemplo, son elementos del SO los empleados, los escritorios, las computadoras, etc.
- ✓ Asimismo, la parte operacional del sistema de información creado para servir al SO termina siendo parte de él.
- ✓ El Sistema de Información (SI). Langefors realiza una definición amplia del concepto, considerando como tal a: "... cualquier sistema usado para proveer información (incluyendo su procesamiento), para cualquier uso que pueda hacerse de ella...".
- ✓ El Sistema de Datos (SD). Es el conjunto de símbolos que se refieren a fenómenos asociados con su SO.



**Fig. 1. Interrelación entre SO, SI y SD**

En la figura 1 mostramos la relación que existe entre los distintos conceptos. Nótese que el sistema de información cubre parcialmente al sistema objeto. Esto sucede precisamente por lo que decíamos anteriormente: cuando un SI se pone operativo pasa a formar parte del SO. La porción que queda fuera de él representa todas aquellas versiones de programas y líneas de código que no forman parte de la operatoria diaria del sistema, pero que se crearon para darle forma definitiva.

Asimismo, vemos que el sistema de datos forma parte en su totalidad del SI. Ya analizamos que se trata de la parte estática que alimenta al sistema de información. A su vez, parte del sistema de datos se corresponde con aquellos que se encuentran en la base de datos de producción (y por lo tanto forman parte del sistema Objeto), y otra está fuera de del sistema objeto, que serían aquellos esquemas de BD que se crearon en desarrollo y que permanecen luego de la implementación con fines de mantenimiento.

<sup>3</sup> Langefors, Börje. "Sistemas de Información. En Tecnología Informática". Vol. 1 Nro. 5 Mayo 1982.

En los comienzos los sistemas de información no trabajaban con Bases de Datos (BD), tal como las conocemos hoy. El almacenamiento de datos se realizaba sobre los llamados archivos tradicionales, que no tenían incorporadas funciones de seguridad, integridad, confiabilidad, etc. Éstas debían ser programadas dentro del código de la aplicación.

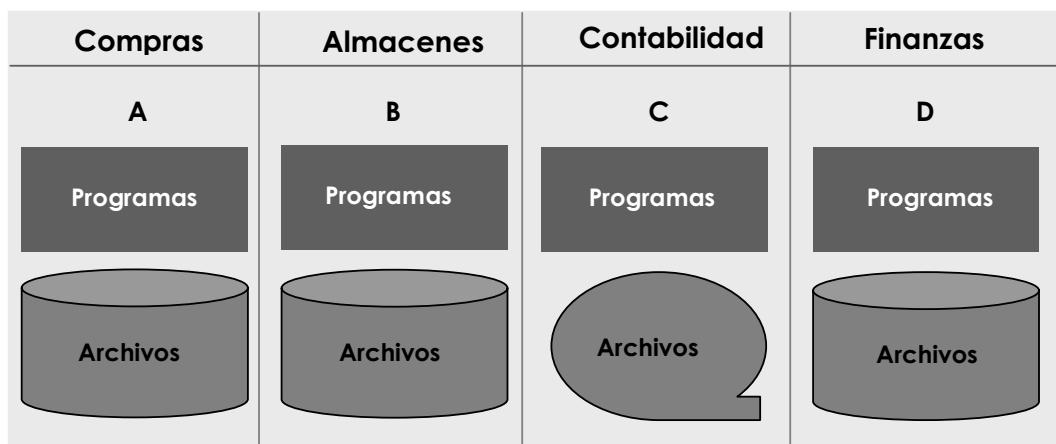
La realidad ha cambiado en las dos últimas décadas de manera significativa. Hoy en día es difícil encontrar sistemas que no trabajen con Bases de Datos, la industria ha acompañado la evolución de las mismas, incorporando nuevas funcionalidades y reduciendo sus precios y la complejidad en su operación.

Ahora bien, la correcta implementación de una BD (que brinde utilidad y sea confiable para quienes la usan), debe estar precedida de un correcto diseño. Los Sistemas de Gestión de Bases de Datos (SGBD) no son más que herramientas que en absoluto pueden suplir errores de diseño. Este libro se refiere precisamente a repasar técnicas de buen diseño de una Base de Datos.

#### **Esquema de archivos tradicionales**

El concepto de Base de Datos supera el tradicional de archivos tradicionales o convencionales. En este último cada uno de los analistas realizaba la definición de los archivos que se utilizarían en la aplicación, y era en cierta medida el “dueño” de sus características y quizás el encargado de mantenerlo actualizado.

Estructurado de esta forma, existían archivos que se diseñaban para cada una de las aplicaciones, que a su vez eran mantenidas por los analistas que los creaban. Como a su vez era habitual encontrar analistas asignados funcionalmente a las aplicaciones de cada sector, existían entonces aplicaciones para cada área, con los programas propios de esa aplicación, con archivos con las características propias de cada una de ellas. En la Figura 2 se muestran estos conceptos.



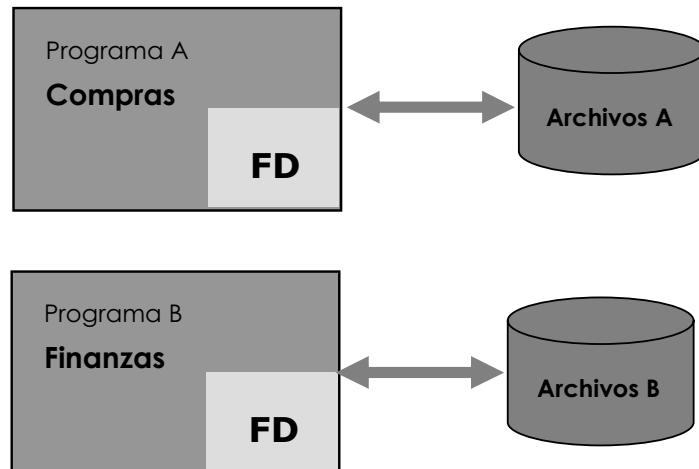
**Fig. 2. Esquema de archivos convencionales**

Las letras A, B, C y D que figuran corresponden a los analistas encargados de cada una de las aplicaciones.

Este tipo de esquema traía aparejados ciertos inconvenientes. Era probable que si dos aplicaciones utilizaban los mismos datos, estos residieran duplicados en cada una de ellas. Por ejemplo, en el ejemplo de la figura 2, tanto las aplicaciones de Compras como las de Almacenes deben utilizar datos sobre los artículos. Era habitual encontrar archivos de artículos diferentes en cada una de las aplicaciones con los problemas que ello implicaba, como por ejemplo diseños de campo diferentes (distinta extensión, tipo de dato, codificación, etc.), descripciones diferentes para el mismo artículo, etc.

Otro problema que se presentaba era la rigidez en la relación archivo/programa. Estos últimos tenían un área en donde se definían las características lógicas y físicas de los archivos. Si algunas de ellas se modificaban, debía modificarse la definición de todos los programas que utilizaban ese archivo, y luego al menos recompilarlo, no obstante existir ciertas técnicas automáticas que simplificaban el proceso pero que aún así se debían llevar a cabo.

En la figura 3 se describe la relación directa que existe entre la definición de cada archivo con los programas que los utilizan. La FD (File Description) es el área del programa en la que se definen las características de los archivos, que debe adaptarse ante cambios en esas definiciones.



**Fig. 3. Relación Programas y archivos en un esquema convencional**

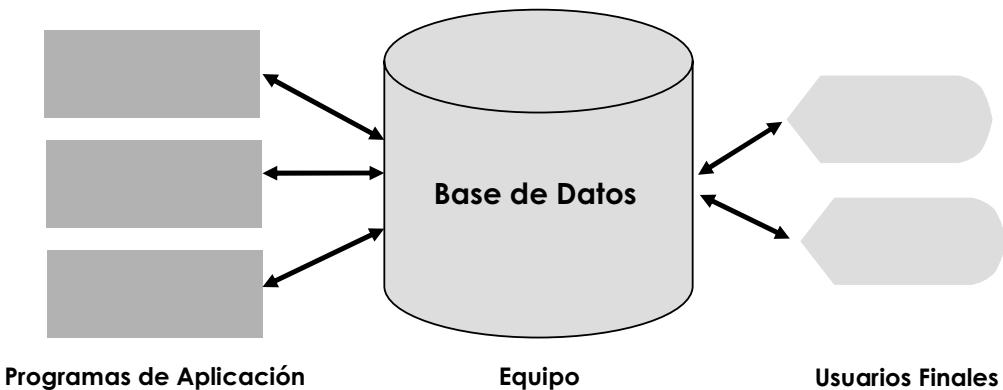
Este tipo de diseño está cayendo en desuso, aunque aún subsisten algunas aplicaciones que lo preservan. Las limitaciones más salientes que tiene son:

- ✓ La creación de archivos adicionales implica redundancia de datos. Esto trae aparejados los siguientes problemas: desperdicio de espacio en el almacenamiento, posible incoherencia entre los nuevos datos y los existentes y duplicidad de datos.
- ✓ Escasa o nula interrelación entre los datos de los distintos archivos. Esto puede acarrear el problema de estar trabajando de dos maneras diferentes el mismo dato.

- ✓ Dependencia de los programas aplicativos. La definición de los datos está asociada con los programas que los utilizan.
  - ✓ Problemas de integridad por actualización del mismo dato en diferentes momentos. Como los distintos sistemas tienen tiempos de actualización diferentes, puede ser que el mismo dato esté almacenado con contenido diferente en el mismo momento, si se actualizó en el archivo A y aun no se actualizó en el B.

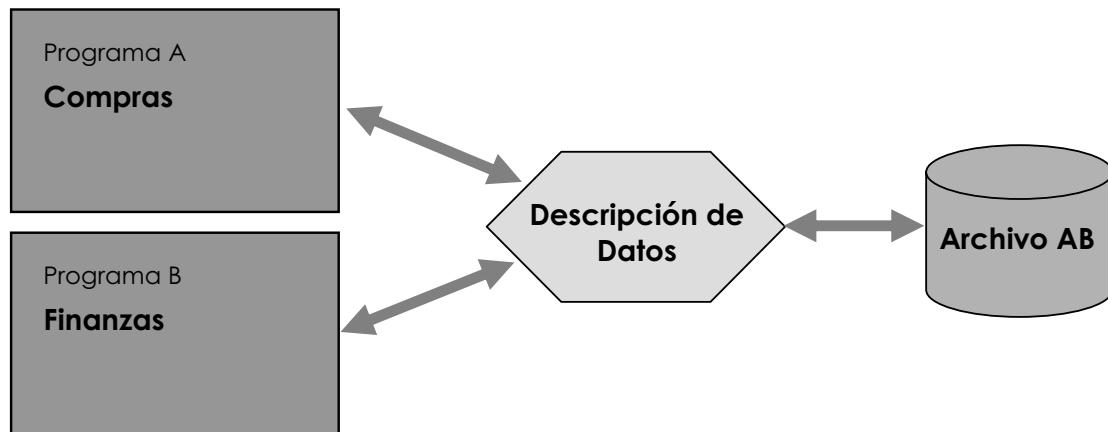
## **Esquema de Bases de Datos**

Como respuesta para solucionar las limitaciones que presentaba el esquema de archivos tradicionales surge el concepto de Base de Datos, que centra su visión en el sencillo concepto de que si un dato es la representación de un hecho, acontecimiento u objeto de la realidad, debe almacenarse solamente una vez y cualquiera que necesite utilizarlo debe accederlo desde allí. Por supuesto, los avances tecnológicos han permitido cristalizar esa idea, y día a día las empresas incorporan nuevas funciones a sus productos.



**Fig. 4. Esquema de procesamiento con Base de datos**

En la figura 4 se observa que todos los programas acceden a un mismo reservorio de información, independientemente de la aplicación a la que pertenezcan. De esta manera, las definiciones internas que debían realizarse en los programas desaparecen, y se incorporan dentro de la Base de Datos.



**Fig. 5. Definición de datos en un esquema de BD**

En la figura superior se representan estos conceptos. Desaparecen de los programas aplicativos las FD y se incluye un único espacio de definiciones de datos, que llamamos Data Description y que genéricamente se denomina metadatos de la BD.

Vemos así que ciertas responsabilidades se le quitan a los tradicionales definidores, y pasan a formar parte de la BD. Pero como esta es un elemento concreto, alguien debe cumplir esas funciones. Aparece así la figura del Administrador de Base de Datos (ABD o DBA en inglés –Data Base Administrator), que es alguien que conoce sobre la herramienta en la que está implementada la Base de Datos, y realiza el mantenimiento correctivo y adaptativo de la misma.

Al producto que implementa físicamente una base de datos y la mantiene en el tiempo lo llamaremos Sistema de Gestión de Base de Datos (SGBD), tales como SQL-Server, Oracle, o DB2.

¿Cuáles son los beneficios que traen aparejados la implementación de una BD? Algunas de ellas son:

- ✓ Independencia de datos. Como vimos diciendo, en un esquema de Base de Datos la definición de estos es una función separada de los programas de aplicación, lo que requiere que quienes realicen esas tareas deban poseer aptitudes y capacidades diferentes. Esto se realiza a través de la definición de los metadatos.
- ✓ Manejo de múltiples vistas. Como una base de datos es accedida por distintos usuarios, de los que no todos necesitan la totalidad de la información, se requieren funciones que, básicamente por temas de seguridad, garanticen que no van a acceder a los datos usuarios que no estén autorizados.
- ✓ Control de concurrencia. Como varios usuarios acceden simultáneamente a la BD, puede ocurrir que más de uno de ellos intente acceder a los mismos recursos, simultáneamente. Esta situación puede producir problemas de inconsistencia en los datos, razón por la cual, las actuales bases de datos proveen el servicio de control de esta situación.

- ✓ Control de Transacciones. Los datos de las BD son utilizados por transacciones, que son secuencias de sentencias que se ejecutan como una unidad lógica de proceso. Dentro de una transacción pueden mantenerse estados inconsistentes temporales, hasta que la transacción se convierte en persistente. Si por alguna razón la transacción cancela, pueden producirse situaciones de inconsistencia de datos que en la actualidad los SGBD los tratan automáticamente.
- ✓ Resguardo de la BD. La operación de Back Up y recupero están actualmente incorporadas como funciones de los SGBD.



Puede profundizar más estos conceptos leyendo el capítulo 1 de **"Fundamentos de Sistemas de Bases de Datos"** 3era Edición de Elmasri – Navathe. Edit. Pearson Addison Wesley.

### Tipos de Bases de Datos

Los datos pueden ser almacenados en "tipos" de Bases de Datos, que poseen estructuras diferentes, y que tienen un nivel de abstracción que las independiza de los productos que se comercializan. Analizaremos algunos de ellos.

### Bases de datos jerárquicas

En este modelo la base de datos se representa en una estructura jerárquica. Los datos se organizan en segmentos con una forma similar a la de un árbol invertido, en donde el nodo padre puede tener varios nodos hijos. El nodo que no tiene padres es llamado raíz, y los nodos que no tienen hijos son llamados hojas.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

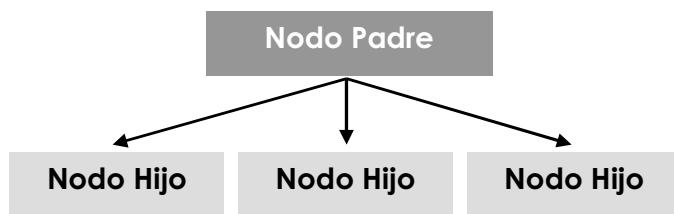


Fig. 6. Modelo de estructura jerárquica

### Bases de datos de red

Este modelo es ligeramente distinto al modelo jerárquico; su diferencia fundamental radica en la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Esta modificación fue introducida para resolver el problema que se presentaba en el modelo jerárquico con respecto a la redundancia de datos; de todas formas lo complejo que resultaba manejar la información ha llevado a que este modelo sea utilizado en su gran mayoría por programadores y no tanto por usuarios finales.

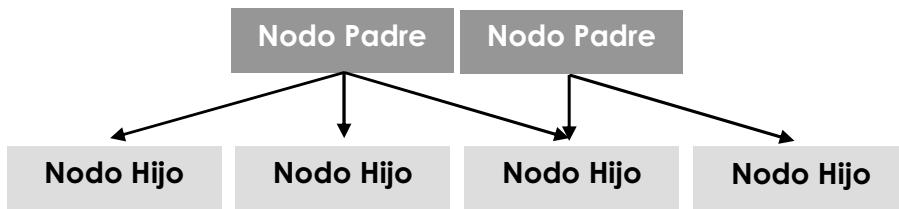


Fig. 7. Modelo de estructura jerárquica en red

### Bases de datos relacionales

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

### Bases de datos orientadas a objetos

Este modelo se basa en la teoría orientada a objetos y maneja la persistencia y almacenamiento de los objetos dentro de ella como objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- ✓ Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- ✓ Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- ✓ Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

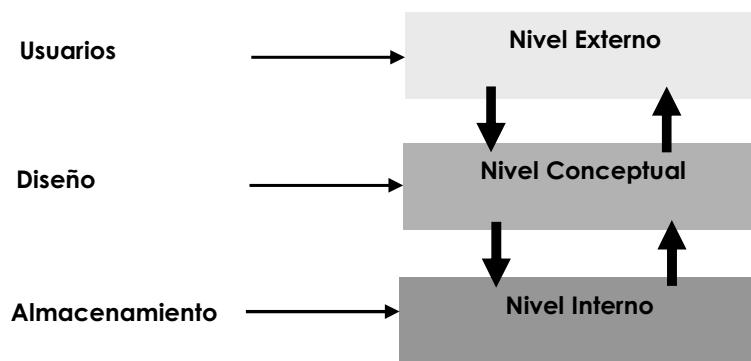
En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

### **Arquitectura de una Base de Datos**

Los SGBD permitieron a todo tipo de usuarios crear y mantener sus bases de datos, dotándolos de una herramienta que era capaz de transformar el nivel lógico que éstos diseñaban en un conjunto de datos, representaciones y relaciones, traduciéndolo al nivel físico correspondiente. Para que fuese posible dar seguridad en el intercambio de datos fue necesario normalizar los esquemas que guiaban la creación de las bases de datos.

Las bases de datos respetan la arquitectura de tres niveles definida, por el grupo ANSI/SPARC. En esta arquitectura la base de datos se divide en los niveles interno, conceptual y externo.

1. **Nivel interno:** es el nivel más bajo de abstracción, y define cómo se almacenan los datos en el soporte físico, así como los métodos de acceso.
2. **Nivel conceptual:** es el nivel medio de abstracción. Se trata de la representación de los datos realizada por la organización, que recoge las vistas parciales de los requerimientos de los diferentes usuarios y las aplicaciones posibles. Se configura como visión organizativa total, e incluye la definición de datos y las relaciones entre ellos.
3. **Nivel externo:** es el nivel de mayor abstracción. A este nivel corresponden las diferentes vistas parciales que tienen de la base de datos los diferentes usuarios. En cierto modo, es la parte del modelo conceptual a la que tienen acceso.



**Fig. 8. Separación de niveles de abstracción de una BD**

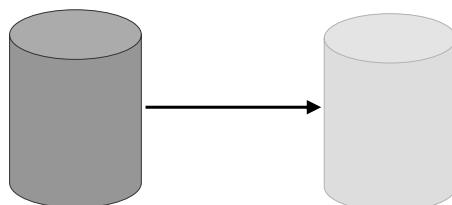
En ocasiones puede encontrarse el nivel conceptual dividido en dos niveles, conceptual y lógico. El primero de ellos corresponde a la visión del sistema global desde un punto de vista organizativo independiente, no informático. El segundo correspondería a la visión de la base de datos expresada en términos del sistema que se va a implantar con medios informáticos.

El modelo de arquitectura propuesto permite establecer el principio de independencia de los datos. Esta independencia puede ser lógica y física

### Independencia de Datos

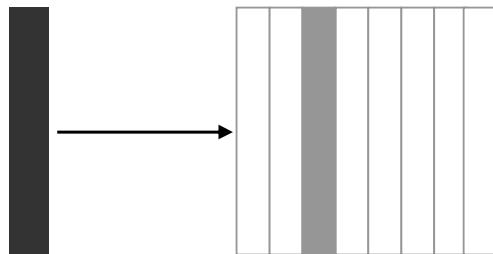
Definimos independencia de datos como la capacidad de modificar la definición de un nivel sin afectar los niveles superiores.

A su vez la independencia de datos se puede subdividir en independencia física de datos e independencia lógica de datos.



**Fig. 9. Independencia física de datos**

Por independencia física entendemos a la capacidad de modificar la definición física y/o métodos de almacenamientos, sin afectar el esquema, las aplicaciones de usuario ni las vistas; mientras que independencia lógica de datos quiere describir la capacidad de modificar el esquema sin afectar las aplicaciones de usuario ni las vistas.



**Fig. 10. Independencia lógica de datos**



Puede profundizar más estos conceptos leyendo los capítulos 1 y 2 de **“Fundamentos de Sistemas de Bases de Datos”** 3era Edición de Elmasri – Navathe. Edit. Pearson Addison Wesley.



# B

## Proceso de diseño de una base de datos

---

### Objetivos

- ✓ Adquirir conocimientos sobre el concepto de modelado de bases de Datos.
- ✓ Identificar las etapas en que se divide un proceso de modelado de Bases de Datos.
- ✓ Comprender la diferencia entre el problema infológico y datalógico.
- ✓ Establecer diferencias entre los distintos modelos.

## B. Proceso de diseño de una base de datos

Como ya dijimos, las BD son un componente fundamental de los sistemas de información. Intentaremos en este capítulo determinar el momento dentro del proceso de generación de un SI en que se procede el diseño de la BD.

En un sentido amplio, un SI trasciende lo tecnológico y lo podemos hacer extensivo a SI manuales, SI mecanizados, etc. No obstante, aquí acotaremos nuestra visión al ámbito informatizado.

Ya hemos mencionado que un SI es un modelo del mundo real, y como tal opera de una manera semejante a este, aunque en términos estrictos no es la realidad en sí misma. Sus elementos no son los mismos, su comportamiento se limita solo a lo relevado, etc. En el esfuerzo por realizar modelos de la realidad, se han creado una serie de procedimientos y elementos que ayudan en este camino.

### Ciclo de Vida de un SI

El desarrollo de un SI comprende una serie de etapas cuya totalidad conforma el denominado ciclo de vida. Saroka<sup>4</sup> propone, en base a publicaciones de la empresa Rational Software Corporation un ciclo de vida compuesto de las siguientes etapas:

1. **Inicio.** Se define aquí la visión general del sistema, su alcance y se toma la decisión de invertir o no en el análisis detallado.
2. **Elaboración.** En esta etapa se realiza un análisis del sistema objeto para el que se busca una solución. Aquí se define la estructura preliminar del sistema, se analizan los riesgos del proyecto y se elabora un plan detallado de trabajo.
3. **Construcción.** En este momento se “fabrica” el sistema y los productos de apoyo necesarios.
4. **Transición.** Es la etapa que se inicia cuando el sistema se transfiere a los usuarios y concluye cuando este da su visto bueno.

De estas cuatro etapas, es en la de **Construcción** en la que se formaliza el diseño de la Base de Datos del sistema y por lo tanto es en ella en la que se llevan a cabo las tareas que mayor tiempo demandan.

El buen diseño de una BD requiere ir tomando decisiones a distintos niveles, y en este sentido es que conviene descomponer el proceso de diseño en varias etapas, cada una de las cuales tendrá sus objetivos, utilizará sus técnicas y obtendrá sus productos específicos.

Adoptaremos para este análisis un modelo desarrollado en la década de los '70 que tiene plena vigencia en la actualidad. Centra la atención en el proceso de diseño de los datos y sus

---

<sup>4</sup> Saroka, R. H. "Sistemas de Información en la Era Digital". Fundación OSDE. 2002.

propiedades. Este método diseña por un lado las Bases de Datos y por el otro las aplicaciones que usan de ella. Descompone el proceso en 4 etapas: la de relevamiento, la de diseño conceptual, la de diseño lógica y la de diseño físico.

En la etapa de **relevamiento** se determinan los límites del problema que debemos solucionar con el desarrollo del sistema, y se identifican las ideas y definiciones que pueden ayudar a su mejor definición. Al resultado de este proceso lo llamaremos Esquema Descriptivo.

El **diseño conceptual** parte de las definiciones del relevamiento y realiza una descripción de alto nivel de la estructura de manera independiente del software de BD que utilice. Batini<sup>5</sup> identifica que el propósito de esta etapa es describir el contenido de información de la BD, mas que las estructuras de almacenamiento que se necesitarán para manejar esta información y que el diseño conceptual debe hacerse aun cuando la implantación final no use un SGBD, sino archivos convencionales.

En el **diseño lógico** se definen los significados de las palabras del lenguaje natural y otras construcciones empleadas en el diseño conceptual de manera que puedan ser entendidas por el modelo de BD que vaya a utilizarse. El modelo lógico mas utilizado que adoptaremos en este libro es el relacional. Aclaremos aquí que el diseño lógico depende del modelo de datos utilizado por el SGBD y no del producto específico que se utilice. Por ejemplo, tanto SQL Server, Oracle o DB2 utilizan el modelo relacional, por lo tanto, cualquier diseño que haya adoptado este modelo puede ser utilizado en cualquiera de ellos. Es en esta etapa en la que se le quitan al modelo cierta redundancia y complejidad.

Por último, en la etapa de **diseño físico** se definen las estructuras de almacenamiento y los métodos de acceso efectivo a los datos. Por lo tanto ya estamos trabajando con un SGBD específico, del cual depende esta implementación.

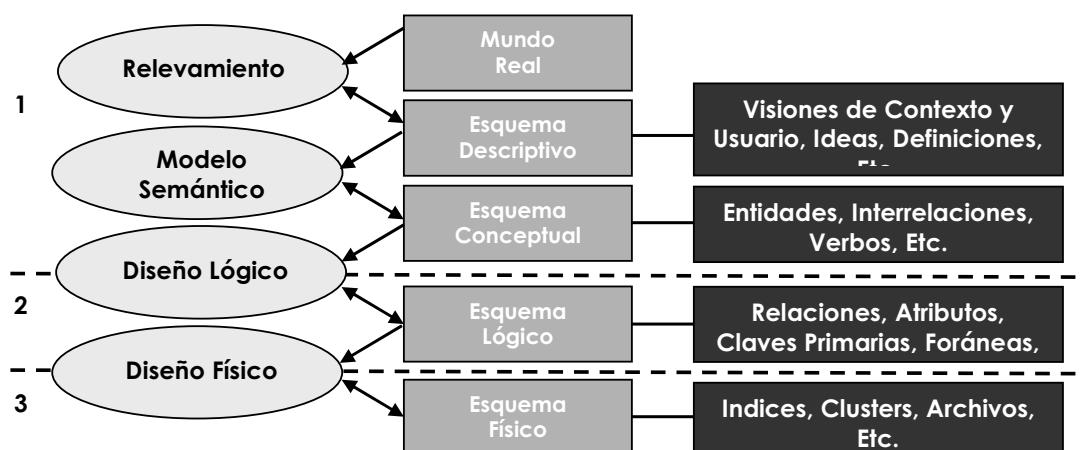


Fig. 1. Descomposición del proceso de diseño de los datos

<sup>5</sup> Batini C., Ceri S y Navathe S. "Diseño Conceptual de bases de datos". Edit. Addison Wesley. 1992.

La figura superior ha sido extraído del libro de Dejean<sup>6</sup> y clarifica aspectos de lo expresado. En la primera columna se especifican los pasos del proceso de diseño, en la segunda el resultado de dicho proceso y en la tercera se ejemplifican algunos de los resultados y/o productos que se pueden obtener en cada etapa.

Las líneas punteadas horizontales que hemos agregado delimitan actividades y su relación con los tipos de bases de datos y los productos en particular. Por ejemplo, todas aquellas actividades y tareas realizadas en el espacio 1 no dependen del tipo ni del producto de BD que se utilice. Lo especificado en el espacio 2 depende del tipo pero no del producto, y la etapa 3 depende del producto y por lo tanto del tipo de base de datos soportado por él.

Nótese que si bien se han descripto las etapas de manera lineal, la secuencia de ejecución en realidad no lo es. Es posible que a raíz de observaciones realizadas en la etapa de diseño lógico deban reverse temas de la etapa de diseño conceptual. Lo mismo puede suceder en las otras etapas.

Lo que es importante observar es que debe avanzarse significativamente en cada etapa para pasar a la siguiente. Si las especificaciones fueran superficiales, poco sólidas y pasáramos a la etapa posterior, los costos de corrección serían muy elevados. Muchas veces los errores son observados en las etapas finales del diseño. Imaginemos tener que corregir el diseño conceptual por un error observado en la etapa de diseño físico.

Otro punto a destacar es que la presión de la tecnología fuerza ciertas modificaciones en la modalidad de trabajo. Actualmente existen productos que trabajan el Diagrama de Entidad/Relación, pero que solicitan información adicional que permiten automatizar el proceso de conversión directa al modelo lógico. De esta manera, estamos pensando casi simultáneamente aspectos que tienen lógicas diferentes.

En este punto nos parece importante y esclarecedor traer unos conceptos especificados por Langefors<sup>7</sup>, porque a partir de ellos se visualiza claramente las habilidades y aptitudes que deben tenerse en cada etapa. Nos habla de dos categorías de problemas:

- ✓ ¿Qué información proveerá el sistema a fin de satisfacer las necesidades de los usuarios? (problema infológico).
- ✓ ¿Cómo será estructurado y operado el sistema aprovechando eficientemente la tecnología de procesamiento de datos disponible? (problema datológico).

El mismo Langefors nos llama la atención de la importancia sobre esta clasificación, ya que los problemas en cada una de las categorías son muy diferentes y demandan distintos métodos y habilidades.

<sup>6</sup> Dejean, Gustavo. "Modelado de Base de Datos". Edit. C&C. 1998.

<sup>7</sup> Langefors, Börje. "Sistemas de Información. En Tecnología Informática". Vol. 1 Nro. 5 Mayo 1982.

Realizando un paralelo entre el proceso de diseño de una BD y estas categorías expuestas, podemos realizar el siguiente cuadro:

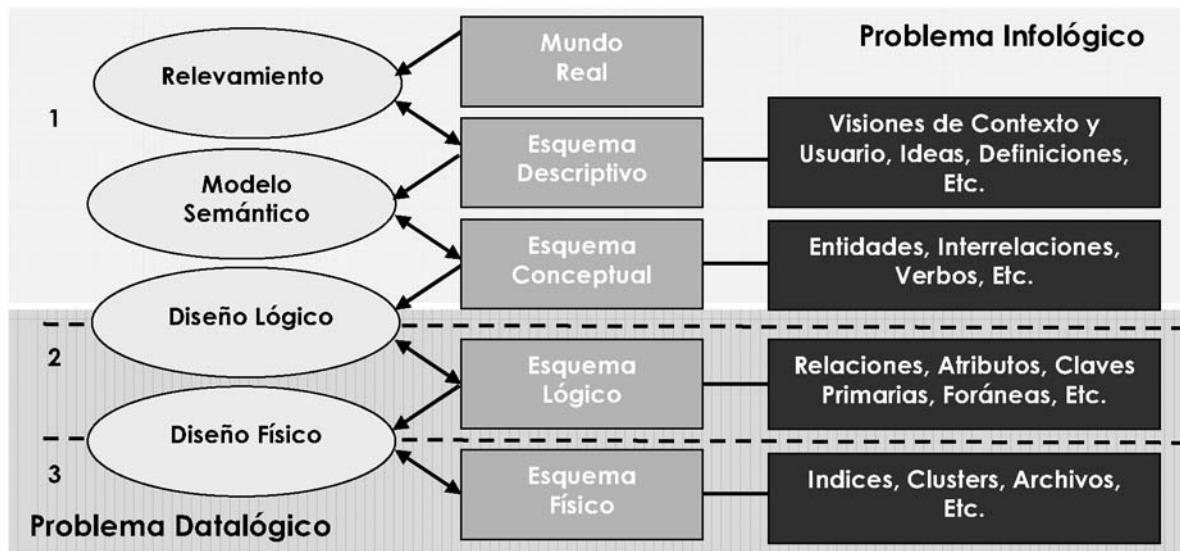


Fig. 2. Separación de Problemas Infológico y Datalógico

Todas aquellas actividades y tareas vinculadas con los esquemas descriptivo y conceptual, se asocian con el problema Infológico al que hace referencia Langefors, y las especificaciones tanto del esquema lógico como físico se asocian con el problema datalógico.

En este libro nos ocuparemos de aquellos temas vinculados con el diseño conceptual y lógico; y en cuanto a la etapa de diseño físico solo abordaremos los aspectos vinculados con la descripción de las estructuras de almacenamiento.

Es cierto que hoy en día la tecnología CASE permiten generar código en forma automática, bajo ciertas estructuras, por ejemplo diagramas de clases que generan código respecto de atributos y propiedades, o los mismos SBGD que generan sintaxis DDL de forma automática, hace cada vez más ambiguo el límite entre estas dos etapas (conceptual y lógica). No obstante es necesario separarlas y definirlas, ya que cada una de ellas requiere de conocimientos diferentes, habilidades profesionales distintas, etc.

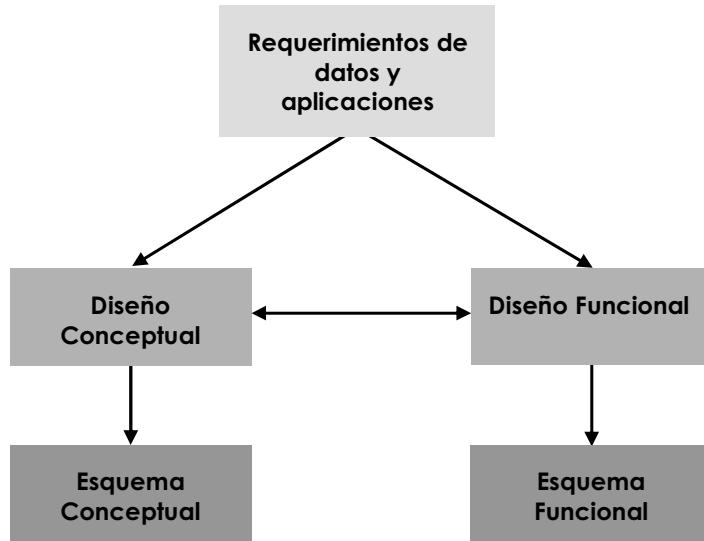
### Modelo de datos y Modelo Funcional<sup>8</sup>

Existe otro modelo muy difundido a partir de la década de los '60, que es el modelo funcional. Comienza realizando un análisis de alto nivel de los requerimientos de las aplicaciones. El resultado de esta etapa es un esquema de funciones en el que se describen las acciones y flujos de información. En esta etapa, las bases de datos son los depósitos específicos de datos que utilizan los procesos, especificados de manera global.

<sup>8</sup> Batini C., Ceri S y Navathe S. "Diseño Conceptual de bases de datos". Edit. Addison Wesley. 1992.

El paso siguiente en el modelo funcional es el de diseño de las aplicaciones de alto nivel. Es aquí en donde, con un alto nivel de abstracción, se describe la conducta de los programas de aplicación, incluyendo la manera en que estos obtienen acceso a las bases de datos. Las especificaciones obtenidas en esta etapa sirven de base para el diseño de los programas de aplicación, que es la última etapa del proceso.

El modelo funcional y el modelo conceptual no son excluyentes, sino complementarios. La idea es ir generando paralelamente un diseño funcional y uno de datos, de manera que ambos sean mutuamente coherentes y completos. Esta sinergia es la que se muestra en la figura siguiente:



**Fig. 3. Relación entre el Modelo Conceptual y el Modelo Funcional**

Los modelos cuentan con herramientas de graficación y diagramación que en algunos casos los caracterizan fuertemente. Así como en el modelo conceptual se ha aceptado el uso del Diagrama de Entidad Relación como un estándar, en el modelo funcional ha adoptado el Diagrama de Flujo de Datos como patrón generalizado de uso.

En este libro abarcaremos exclusivamente lo relacionado con el modelo conceptual.

# C

## Enunciado del Caso Base

---

### Objetivos

- ✓ Plantear el caso base de la obra a ser resuelto en los capítulos siguientes.
- ✓ Introducir a un relevamiento formalmente especificado.
- ✓ Dar a conocer el concepto de requerimiento y funcionalidad de un sistema.

### C. Enunciado del Caso Base

La empresa Aprendiendo se dedica al dictado de cursos de capacitación relacionados con herramientas de diseño y desarrollo de software.

Según lo manifestado por Adolfo Reinera, gerente del área de Capacitación de la empresa, los cursos son administrados en forma manual, lo que ocasiona una serie de inconvenientes de índole organizativos y financieros. Por esta razón es que se han decidido a iniciar un proceso de informatización y nos contratan a nosotros para llevarlo adelante.

La primera tarea desarrollada fue realizar un relevamiento en las distintas áreas de la empresa, del que se ha obtenido la siguiente información:

- ✓ Los cursos poseen un nombre, una descripción, un tiempo de duración medido en horas, una categoría y un precio. Dado que el precio de un curso puede cambiar en el tiempo es que resulta necesario guardar información histórica del mismo para poder conocer como varió el mismo en el transcurso del tiempo.
- ✓ La categoría de un curso es introductorio, avanzado, etc.
- ✓ A fin de evitar errores en la captura de la categoría asociada a un determinado curso, se ha decidido proceder a codificar los distintos valores.
- ✓ Un curso puede tener como prerequisito la realización previa de otro.
- ✓ Un mismo curso puede tener diferentes ediciones, es decir, se puede impartir en distintos lugares (sedes), fechas y turnos (mañana, tarde o noche).
- ✓ Un curso puede ser dictado por distintos docentes, pero una edición sólo tiene un profesor a cargo. Es importante conocer que docente puede dictar cada uno de los cursos.
- ✓ Un docente puede impartir cursos en cualquier sede.
- ✓ En todas las sedes en las que se dicta un curso, no puede existir dos ediciones del mismo con igual fecha de inicio y turno.
- ✓ Los cursos ofrecidos por Aprendiendo no necesariamente se dictan en todas las sedes.
- ✓ De los asistentes a los cursos interesa conocer su nombre, apellido, dirección, teléfono/s, Número de Documento, fecha de nacimiento, edad, nacionalidad, sexo, dirección (país, provincia, localidad, calle, numero, piso y departamento), empresa a la que pertenece, los cursos que realizó, en qué sede, la edición en que lo hizo y la nota final obtenida.
- ✓ De las empresas que envían alumnos a los cursos interesa conocer su nombre, dirección, país, provincia, localidad y actividad a la que se dedica.

En otra entrevista, mantenida con Adolfo Reinera, nos comentó que existen cursos en donde algunos de los temas que forman parte del programa, también son dictados como parte del programa de otro/s curso/s. Por este motivo necesita conocer que cursos comparten los mismos temas.

Como los contenidos de los cursos pueden variar en el tiempo, es importante guardar la información histórica de los temas comunes entre los diferentes cursos.

Además, Adolfo nos solicitó que el nuevo sistema permita responder a las siguientes preguntas:

- ✓ ¿Quién impartió la edición de fecha 26 de Abril de 2005 del curso 201 denominado "Diseño de Bases de Datos" en una determinada sede y turno?
- ✓ ¿Quienes asistieron a la edición de fecha 4 de Febrero de 2005 del curso 47 denominado "Programación en Java"?

- ✓ ¿Cuáles son los temas que tienen en común el curso de "Introducción a SQL Oracle" y "Introducción a SQL Server?"

**Importante**

A lo largo de los capítulos siguientes de la obra se procederá a resolver cada uno de los puntos correspondientes.

**Se pide**

1. Realizar un modelo conceptual que cubra todos puntos relevados.
2. Realizar el pasaje del modelo conceptual a un modelo de datos expresado en tercera forma normal. Adicionalmente realizar el diagrama lógico correspondiente al modelo de datos expresado en tercera forma normal.
3. Realizar la creación de la tabla en la que se guardan los asistentes a una edición de un curso, definiendo la clave primaria y las claves foráneas que considere necesarias.
4. Realizar la creación de la tabla en la que se guardan los cursos que dicta Aprendiendo, definiendo la clave primaria, las claves foráneas, las restricciones de obligatoriedad (Not Null) que considere necesarias y al menos una restricción de dominio(Check).



# D

## **Diseño Conceptual**

---

### Objetivos

- ✓ Conocer los elementos que componen el modelo de Entidad-Relación.
- ✓ Adquirir destreza en el manejo y construcción de Diagramas de Entidad-Relación.

## D. Diseño Conceptual

En este capítulo analizaremos la etapa de diseño conceptual de la metodología propuesta en el **Capítulo B**. En ella se parte de los datos obtenidos en el relevamiento y se formalizan en un modelo con un alto nivel de abstracción. Hablamos de abstracción, ya que se extraen del mundo real aquellos elementos y objetos con las propiedades que se consideran pertinentes para el análisis que se efectúa.

El modelo más difundido, que utilizaremos en el capítulo es el conocido como Modelo de Entidad Relación (**MER**). Con respecto a este nombre cabe señalar que gramaticalmente debería utilizarse Modelo Entidad Interrelación, ya que deriva de la expresión inglesa **relationship**, que significa precisamente interrelación. Por seguir la expresión más difundida en la bibliografía castellana en este libro utilizaremos la primera expresión.

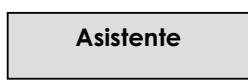
El modelo fue propuesto por Chen en la década de los '70. Se focaliza en los datos, y no toma aspectos vinculados con los procesos. La versión básica del modelo reconoce 3 elementos: las entidades, las relaciones y los atributos. Algunos autores agregaron con posterioridad otros conceptos para que el modelo fuera más expresivo, arribando al conocido como MER extendido.

### Modelo Entidad Relación básico

Parte de 3 conceptos que son las entidades, las relaciones y los atributos.

Las **entidades** son elementos que existen en el sistema objeto de análisis, y como tales necesito identificarlos dentro de mi modelo. Korth<sup>9</sup>, define a una entidad como: "Una cosa u objeto en el mundo real que es distingible de todos los demás objetos". Por Ejemplo **Empleado** es una entidad del sistema de sueldos, **Autos** es una entidad de un sistema de remises, etc.

El símbolo utilizado para representar una entidad es un rectángulo en cuyo interior se le agrega su nombre. El nombre de una entidad es un sustantivo escrito en singular. Por ejemplo, en el caso Aprendiendo existe una entidad que representa los asistentes a los cursos, que debería representarse de la siguiente manera:



Asistente

Los **atributos** son características de las entidades que considero necesario individualizar para mi modelo. Se representan a través de un óvalo en cuyo centro se agrega el nombre; por ejemplo, el atributo nacionalidad de los asistentes se representaría:

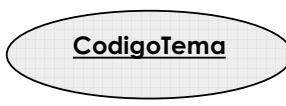
---

<sup>9</sup> A. Silberschatz, H. Korth y S. Sudarshan. "Fundamentos de Bases de Datos" 5ta. Edición. Edit. McGraw-Hill. 2006.



### Nacionalidad

El modelo requiere que exista un atributo o conjunto de atributos que identifiquen inequívocamente a cada entidad. A este atributo o atributos se lo/s denomina/n **identificador único** y se representa/n de la misma manera que cualquier atributo pero con su nombre subrayado. Por ejemplo, el campo Código de Tema es el identificador único de la entidad Tema, y se representa:



### CódigoTema

El identificador único debe ser uno o la conjunción de varios atributos que en la entidad no pueda/n repetirse. Por ejemplo, en la entidad Asistente, la Fecha de Nacimiento no puede ser identificador único porque pueden existir 2 asistentes que hayan nacido el mismo día. En cambio, el Código de Asistente sí, porque es un número que identifica a cada asistente individualmente, y 2 asistentes no van a tener nunca el mismo Código.

Para que un atributo pueda ser considerado identificador único tiene que existir una característica adicional: el atributo debe ser conocido para todas las ocurrencias de la entidad. Por ejemplo, el campo Número de Documento puede ser considerado para que sea el identificador único de la entidad Asistente ya que no van a existir 2 asistentes con el mismo número de documento, pero puede ocurrir que no sepamos el número de documento de todos los asistentes. En ese caso no podemos tomarlo como identificador único.

Cuando existen 2 o más atributos que puedan ser considerados identificadores únicos, debemos elegir uno de ellos como tal, y los que no hemos seleccionado pasan a ser **claves candidatas**.

Existen atributos que pueden tomar más de un valor para cada ocurrencia de la entidad. Por ejemplo, si nos interesaría conocer más de una dirección (comercial, particular, etc.) de cada asistente, el atributo Dirección de la entidad asistente sería un **atributo multivalorado**. Se lo suele denominar también grupo repetitivo, grupo multivalor o atributo no atómico.

Nótese que en el ejemplo que dimos, es el diseñador quien considera que Domicilio es un atributo multivalorado. Si por el contrario considerara que es necesario tener un solo domicilio, este sería entonces un atributo atómico. Con esto estamos queriendo decir que la representación del modelo dependerá de las observaciones que rescatemos de la realidad que estamos representando.

Un atributo multivaluado se representa con un óvalo de doble línea. Por ejemplo en el caso Aprendiendo, el atributo CódigoDocente es multivaluado en la entidad Curso, ya que muchos docentes pueden dictar el mismo curso. La representación del atributo sería:



Existe otro tipo de atributo, que es el **derivable**, cuya característica es que puede ser obtenido a través de un cálculo realizado a partir de otro atributo que conocemos. Por ejemplo, el atributo Edad de un Asistente del caso Aprendiendo, puede ser obtenido a partir de su Fecha de Nacimiento, que es un atributo que conocemos. Se representa a través de un óvalo punteado; en ese ejemplo sería:



### Relaciones

Por la importancia y complejidad del tema le vamos a dedicar un apartado especial. Las relaciones (o interrelaciones como vimos al principio), son objetos que vinculan 2 o más entidades, con un caso especial en que una entidad se vincula consigo misma.

Por ejemplo, en el caso Aprendiendo, existe un relación entre las entidades Provincia y País, ya que una Provincia “pertenece a un” País. El símbolo para representar una Relación es un rombo con el nombre de la relación en su interior (verbo seguido de una preposición), y líneas que la conectan con las entidades que relaciona. Por ejemplo, en nuestro caso sería:

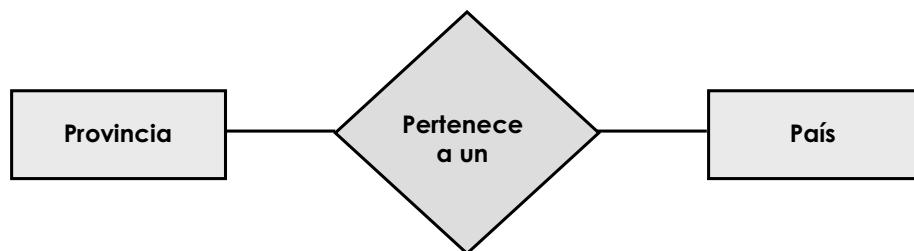
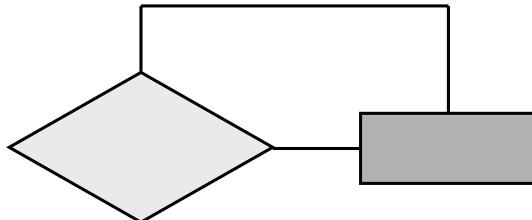


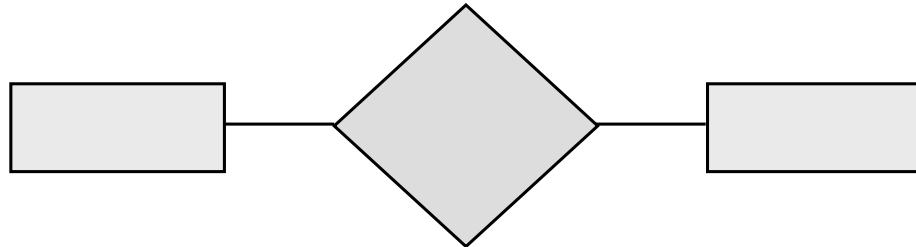
Fig. 1. Ejemplo de Relación

Una relación puede vincular una, dos o más entidades. A la cantidad de entidades que relaciona se lo llama grado de la relación. Así, si vincula una sola entidad se la denomina relación unaria o de grado uno y se representa:



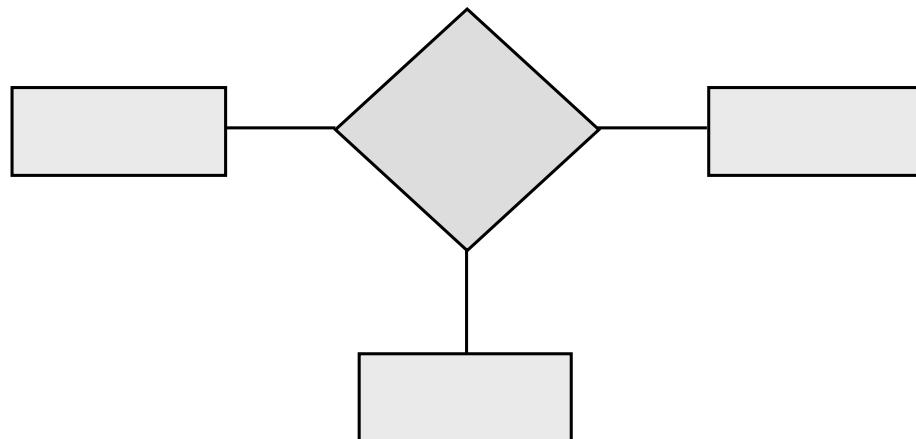
**Fig. 2. Ejemplo de Relación Unaria**

Si relaciona a dos binaria o de grado dos y se representa:



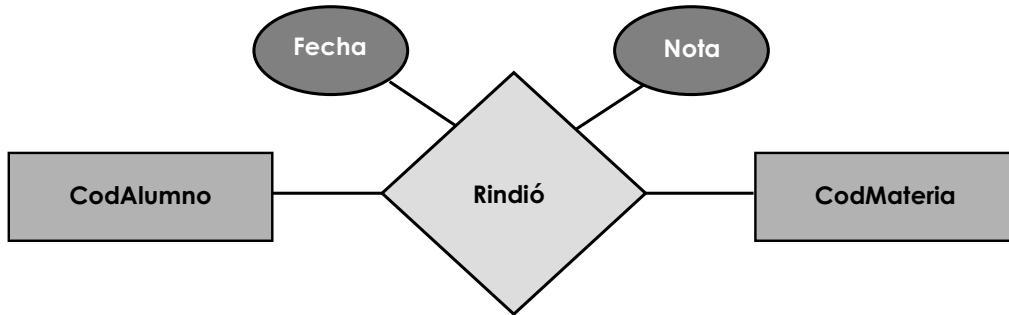
**Fig. 3. Ejemplo de Relación Binaria**

Si vincula a tres se la llama ternaria o de grado tres y su representación:



**Fig. 4. Ejemplo de Relación Ternaria**

Así como las entidades tienen atributos, las relaciones también pueden tenerlos. Por ejemplo, en la relación Rindió que vincula a las entidades Alumno y Examen, existen al menos 2 atributos que son Fecha de Examen y Nota, que no pertenecen a ninguna de las entidades individualmente, sino a la relación, ya que representa la nota que un determinado alumno obtuvo en un determinado examen. Se representa de la siguiente manera:



**Fig. 5. Ejemplo de relación con atributos**

Salvo casos especiales que veremos mas adelante, el identificador único de una relación está formada por los identificadores únicos de las relaciones que une.

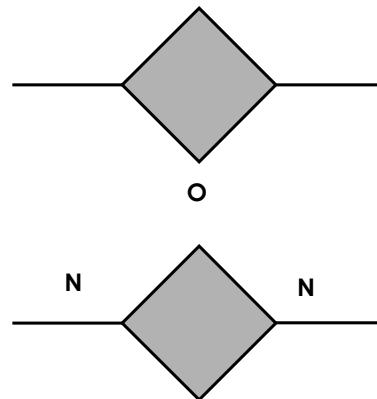
De acuerdo con la visión que se tenga del problema que se está analizando, las relaciones pueden tener algunas limitaciones, una de las cuales se denomina cardinalidad y es la cantidad de elementos de una entidad que pueden vincularse con los elementos de la otra.

Decimos que surge de la visión que se tenga del contexto, porque una misma relación analizada de dos formas distintas puede producir cardinalidades también diferentes. Por ejemplo, analicemos la relación "Tiene por esposa", que vincula las entidades "Empleado" y "Esposa". En un país como la Argentina, la cardinalidad es de 1 a 1, ya que un hombre no puede tener más de una esposa. En cambio, si el contexto es un país que admite la bigamia, la cardinalidad de la relación es de 1 a muchos, ya que un hombre puede tener más de una esposa.

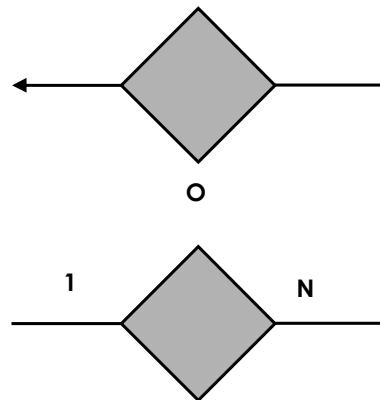
La notación que utilizaremos para marcar cardinalidad es la de dos elementos separados por el signo dos puntos (:). Por ejemplo, la relación uno a uno la marcaremos 1:1, y la relación uno a muchos como 1:N. Cuando la cantidad de elementos es 1 o más, se utilizan las letras M y N.

Las notaciones para relaciones binarias son las siguientes:

- ✓ Cardinalidad muchos a muchos (N:M). Ejemplo: La relación **Tiene a cargo**, que vincula las entidades Profesor y Curso, en el caso en que un profesor puede tener varios cursos a cargo y un curso esta a cargo de más de un profesor.

**Fig. 6. Ejemplo de cardinalidad muchos a muchos**

- ✓ Cardinalidad uno a muchos (1:N). Ejemplo: la relación **Tiene** que vincula a las entidades CabeceraFactura y RenglonesFactura.

**Fig. 7. Ejemplo de cardinalidad uno a muchos**

- ✓ Cardinalidad muchos a uno (N:1). Ejemplo: la relación **Tiene por encargado** que vincula a las entidades Farmacia y Farmacéutico, en el caso en que muchas farmacias son responsabilidad del mismo farmacéutico.

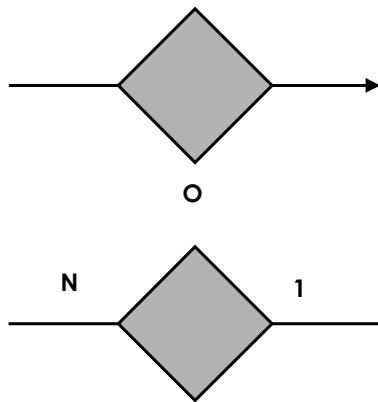


Fig. 8. Ejemplo de cardinalidad muchos a uno

- ✓ Cardinalidad uno a uno (1:1). Ejemplo: la relación **Existe un** que vincula las entidades Ciudad y Farmacéutico, en el caso en que exista un único farmacéutico por cada ciudad.

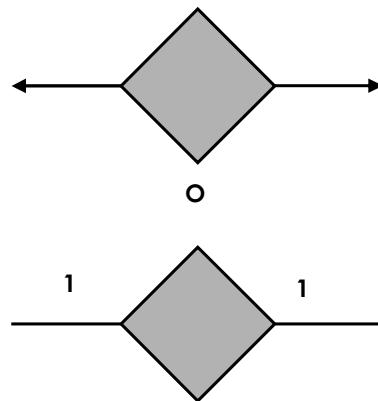


Fig. 9. Ejemplo de cardinalidad uno a uno

- ✓ Existe un caso particular de cardinalidad que es aquella en donde puede suceder que la cantidad mínima de ocurrencias sea cero; por ejemplo en la relación **Tiene por esposa**, los casos de empleados solteros es cero esposa. La forma de anotar esta forma de cardinalidad opcional es:

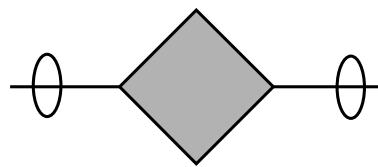


Fig. 10. Ejemplo de cardinalidad cero

### Cardinalidad de relaciones unarias

Las relaciones unarias también poseen su cardinalidad. Tomemos como ejemplo la relación **Tiene por jefe**, correspondiente a la entidad **Empleado**. Posee una cardinalidad 1 a 1, ya que las reglas de negocio relevadas establecen que un empleado tiene sólo un jefe. El diagrama de la relación es la siguiente:

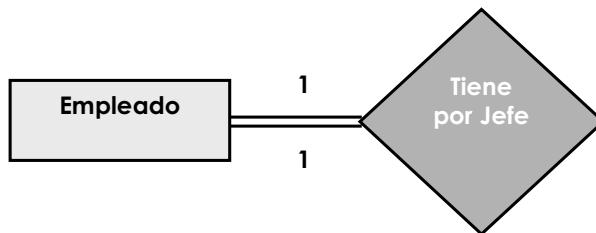


Fig. 11. Ejemplo de cardinalidad unaria uno a uno

En cambio, la relación **Tiene por límite**, correspondiente a la entidad País es de 1 a n, dado que un país puede tener 0 (Cuba), 1 (Portugal) o muchos (Argentina) países límites. La representación del modelo en un DER es equivalente para ambas situaciones con la diferencia de la cardinalidad de la relación.

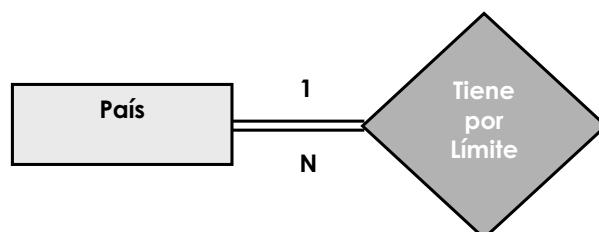


Fig. 12. Ejemplo de cardinalidad unaria uno a muchos

**Entidad Débil**

Es una entidad cuya existencia depende de la existencia previa de otra entidad, la que recibe el nombre de entidad fuerte. Individualmente la entidad débil no posee los suficientes atributos para formar su clave primaria, por lo que debe complementarla heredando la clave primaria de la entidad de la que depende.

Representación de una entidad débil:



Una relación que une una entidad fuerte con una débil se denomina relación débil y se representa:



Un ejemplo de entidad débil es el caso de los hijos que puede tener un empleado. La entidad hijo no posee los atributos suficientes para formar su propia clave primaria, ya que aunque cada hijo puede ser identificado en forma única por su número de documento, esto por sí solo no indica de qué empleado es hijo.

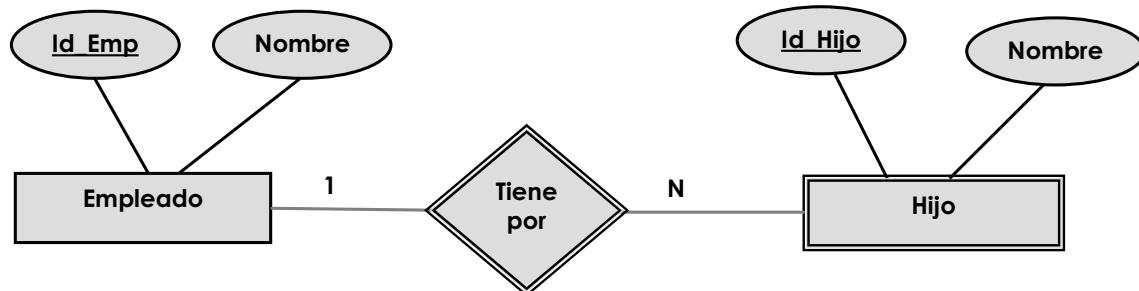


Fig. 13. Ejemplo de Entidad Débil

### Mantenimiento de un historial

Un caso particular de atributo multivaluado que nos parece importante mencionar por su frecuente uso se da cuando deseamos conservar los precios de nuestros productos a lo largo del tiempo; es lo que genéricamente se conoce como histórico de precios.

El problema se plantea cuando queremos incluir el precio como atributo específico de la entidad Producto. En realidad, el precio en este caso no depende únicamente del producto, sino además del lapso en el que cada producto mantuvo el mismo precio. Tendremos así tantos precios del producto como veces se haya modificado a lo largo del tiempo. Podemos decir entonces que el precio es un atributo multivaluado de producto, que está también en función de la fecha a partir de la cual varió. La manera de representar esta situación en un DER es la siguiente:

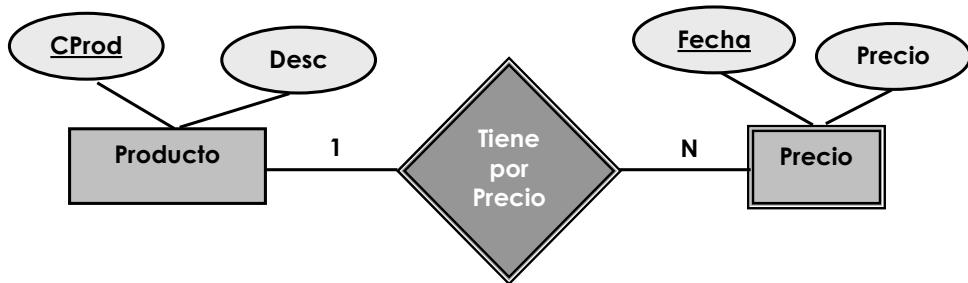


Fig. 14. Ejemplo de atributo multivaluado con respecto a precios

También podemos mencionar el caso de los precios de los productos adquiridos a nuestros proveedores. Supongamos que un mismo producto lo puede proveer más de un proveedor, y un proveedor puede proveer más de un producto. Siendo importante poder conocer el precio al que comercializa cada producto un proveedor y como varía el mismo a través del tiempo. La manera de representar esta situación en un DER es la siguiente:

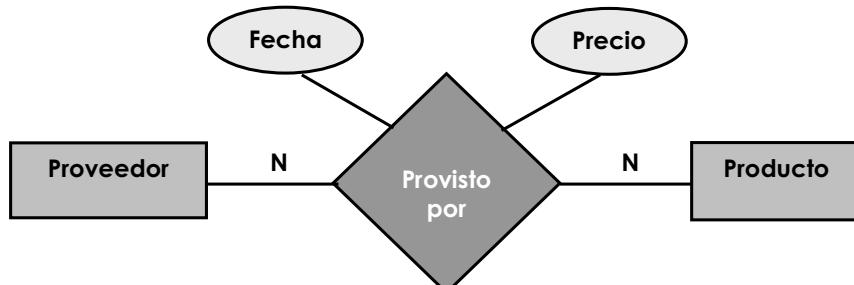
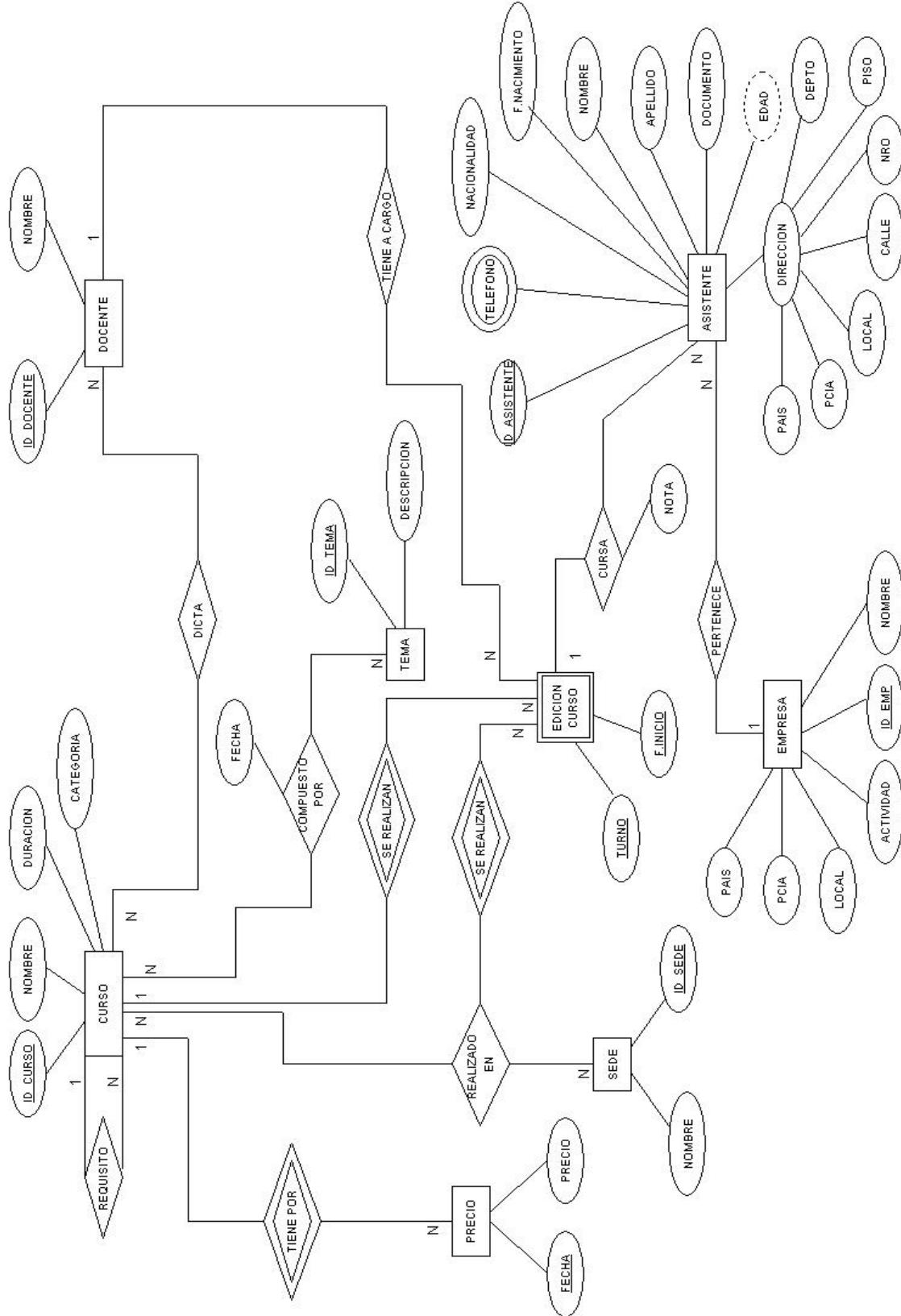


Fig. 15. Ejemplo de atributo multivaluado con respecto a precios provistos por mas de un proveedor

Aquí puede observarse a diferencia del caso anterior que ahora los atributos precio y fecha no dependen de la entidad producto sino que son atributos exclusivos de la relación “provisto por” existente entre las entidad producto y proveedor.

**Importante**

Con todos estos elementos estamos en condiciones de resolver el punto 1 del caso Aprendiendo en su totalidad utilizando la simbología propuesta por Chen, obteniendo como ya dijimos el producto de la etapa de Diseño Conceptual que es el Diagrama Entidad Relación (DER)



### Notaciones Alternativas Diagrama Entidad Relación

En relación a la simbología utilizada para expresar un Diagrama de Entidad relación no existe una notación única que sea aceptada por todos los autores.

Nuestra intención no es hacer prevalecer una notación por sobre otras sino exponer algunas de las diferentes propuestas.

Para ello tomaremos a modo de ejemplo las entidades Curso y Docente conjuntamente con sus atributos y relaciones a los fines de expresar distintos tipos de notaciones.

#### Notación según Chen

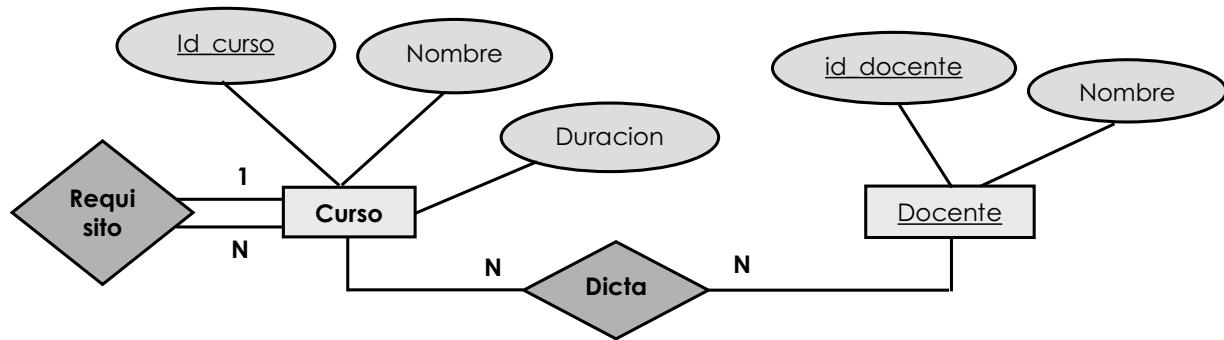


Fig. 16. Notación según Chen

#### Notación alternativa 1

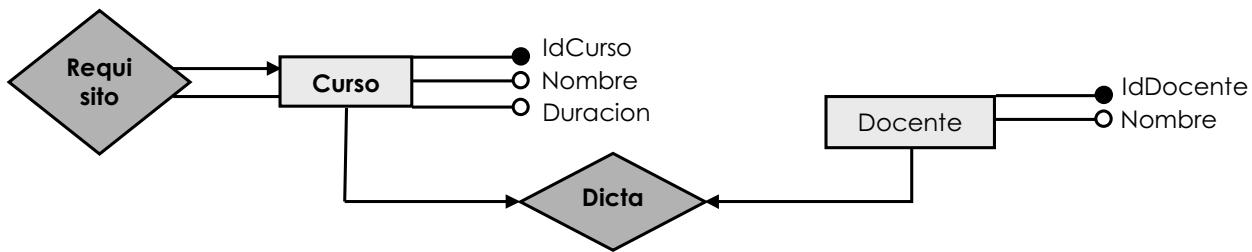


Fig. 17. Notación alternativa 1

**Atributos:** Se colocan a la derecha de la entidad remarcando con un círculo relleno aquellos campos considerados como identificador único.

**Relaciones:** En este caso se señala con una flecha para representar que un curso (el apuntado) puede ser prerequisito de muchos otros.

Para la relación **Dicta** ambos extremos están unidos por flecha lo cual representa una razón de cardinalidad de muchos a muchos.

### Notación Alternativa 2

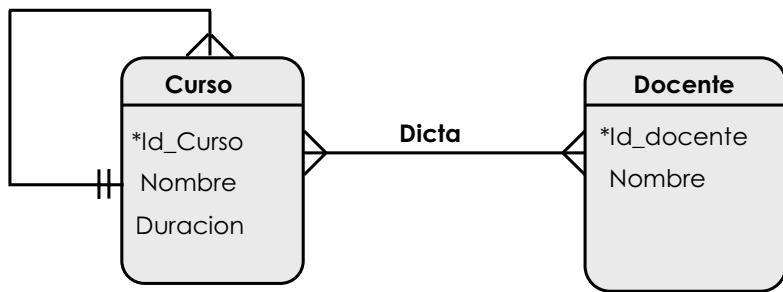


Fig. 18. Notación alternativa 2

Como se visualiza en la figura, este enfoque con diagramas mas cercanos a UML utiliza rectángulos con puntas redondeadas para representar las entidades, señalando los identificadores únicos anteponiendo asteriscos a los atributos que forman parte del mismo y las relaciones toman la forma de los diagramas de **Pata de Gallo** para representar la razón de cardinalidad.

Adicionalmente desiste de la utilización de rombos para la representación de las relaciones al utilizar simplemente rótulos sobre las líneas de las relaciones.



Puede profundizar más estos conceptos leyendo el capítulo 3 de “**Sistemas de Bases de Datos Diseño, Implementación y Administración**” 5ta Edición de Rob – Coronel. Edit. Thompson.



# E

## Modelo E-R Extendido

---

### Objetivos

- ✓ Complementar el modelo E-R.
- ✓ Dar a conocer nuevos conceptos que simplifican, o mejoran el modelado de datos.
- ✓ Lograr un acercamiento a otros modelos, tales como el diagrama de clases y otros utilizados en el modelado de aplicaciones.

### E. Modelo Entidad - Relación Extendido

La mayoría de las veces resulta suficiente con el E-R tradicional pero hay veces en donde es necesario introducir conceptos adicionales tales como la herencia, la especialización, generalización y agregación. Estos conceptos derivan de modelos evolucionados orientados a objetos. El Modelo Entidad-Relación Extendido incluye todos los conceptos del Entidad-Relación e incorpora los conceptos de Subclase y Superclase con los conceptos asociados de Especialización y Generalización. Comúnmente se denomina a este tipo de modelos ERE.

A continuación analizaremos cada uno de los conceptos nombrados.

#### Especialización

Hay casos en donde una entidad puede tener atributos compartidos y atributos propios, tal es el caso por ejemplo de la entidad persona, la cual puede compartir los atributos id\_persona, nombre y apellido, mientras que a su vez puede derivarse en otras entidades de nivel inferior que poseen sus propios atributos, entiéndase como tales entidades de nivel inferior por ejemplo a alumno, docente y secretaria. Esto se llama especialización.

De esta forma la entidad secretaria podría contener atributos propios de ella tal como experiencia\_en\_la\_posicion, estudio\_protocolar, mientras que la entidad docente podría tener los suyos propios tales como posgrado\_en\_docencia, anios\_en\_la\_docencia, etc. Lo mismo sucedería con la entidad alumno. De la misma forma se podrían obtener especializaciones sobre las mismas entidades a su vez más refinadas, tales como docentes con mas de 10 años de servicio de los que no los cumplen, o alumnos con excelentes calificaciones de aquellos cuyas calificaciones son ordinarias. Generalmente una especialización se modela con un triángulo y se suelen denominar a este tipo de relaciones como superclase-subclase, mientras que las entidades se representan mediante rectángulos. A saber:

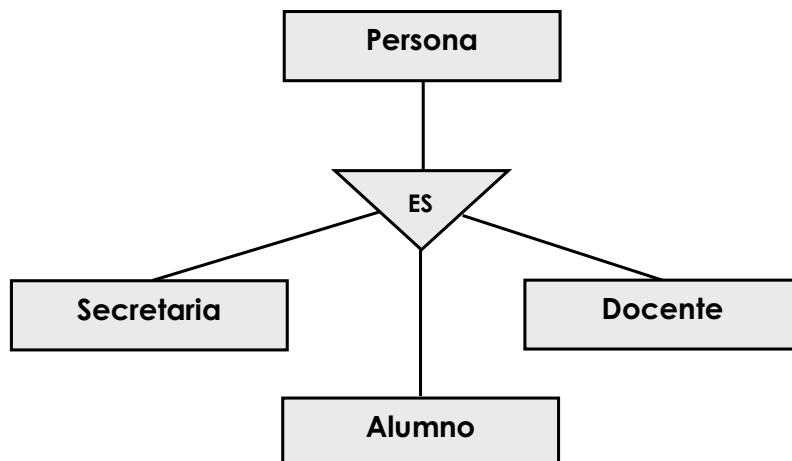


Fig. 1. Representación de una especialización

A continuación se ejemplifica el mismo caso pero agregándole los atributos propios correspondientes a cada entidad.

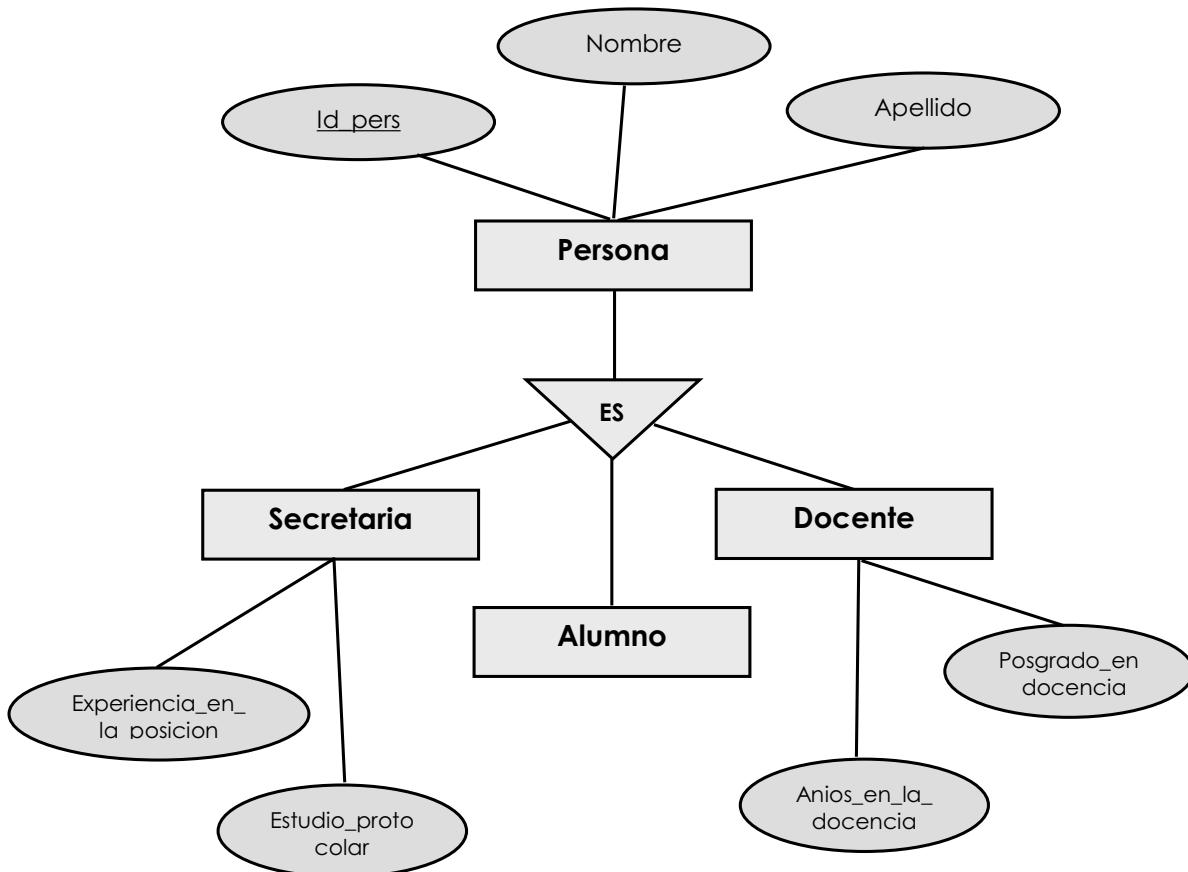


Fig. 2. Representación de una especialización con sus atributos

De esta forma se puede observar que la superclase persona tiene como subclases a las entidades alumno, docente y secretaria, conteniendo cada una de estas subclase sus atributos propios. La especialización permite manejar también el concepto de **herencia** ya que determinados atributos, o sea aquellos que pertenecen a la clase superior o superclase se definen una única vez y en un único lugar y se pueden reutilizar en las clases inferiores, mientras que los propios son definidos en el nivel inferior. La figura 2 representa un ejemplo de una jerarquía de entidades cuya herencia es única, es decir hay un único parente y tres hijos y estos últimos a su vez no son superclase de ninguna otra entidad. En este caso la entidad secretaria, alumno y docente tienen sus propios atributos y a ellos debe sumárseles los que tienen en común y son representados perteneciendo a la superclase persona.



Puede profundizar más los conceptos de herencia leyendo el capítulo 19 de **"Ingeniería de Software Un Enfoque Práctico"** 4ta Edición de Pressman. Edit. Mc Graw Hill.

### Generalización

Por el contrario a la especialización existe el concepto de generalización que busca imponer ciertas restricciones al modelo, tales como la pertenencia a entidades de niveles inferior. De esta forma una generalización es el ejemplo citado anteriormente como persona, o bien podríamos clasificar a los alumnos en Alumnos\_becado y Alumnos\_no\_becado, en este último caso la generalización estaría dada por la superclase Alumno. De esta forma especialización y generalización son dos caras de una misma moneda.

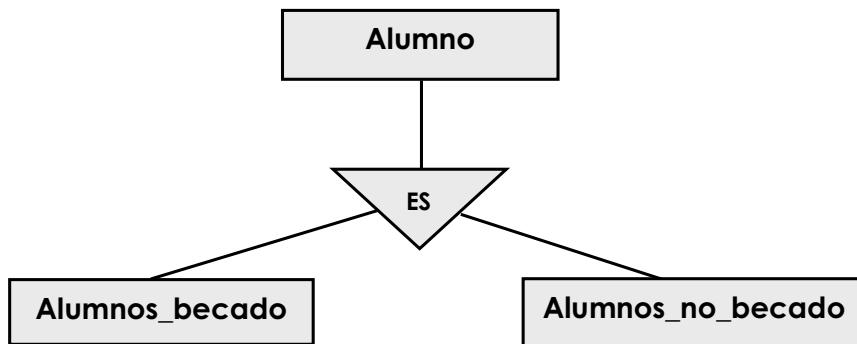


Fig. 3. Representación de una generalización

### Agregación

En el modelo E-R tradicional no es posible representar relaciones de relaciones. De tal forma si queremos representar una relación entre Alumno, materia y sede, y a su vez representar que una comisión de enseñanza tiene por objetivo la revisión de los contenidos de las materias dictados en determinada sede.

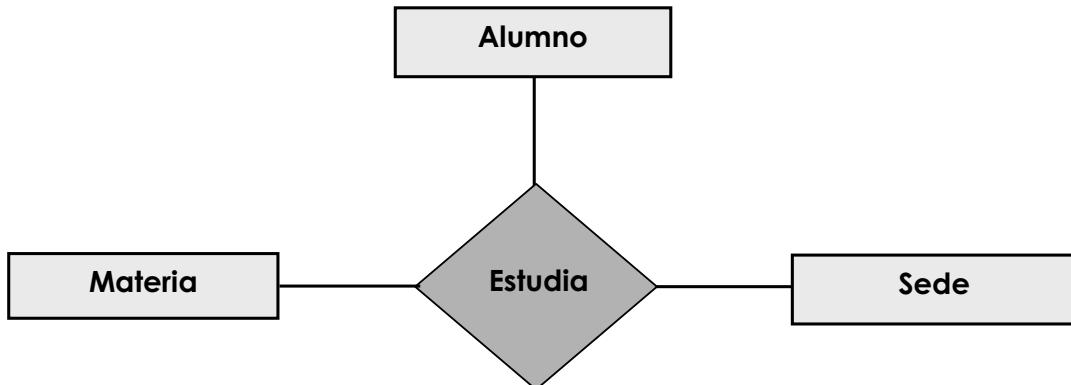


Fig. 4. Representación de una relación ternaria

Sería necesaria la aparición de una relación que vincule Materia con Sede con comision\_de\_enseñanza, de esta forma se podría crear una relación cuaternaria que permita modelar tal realidad. Quedando representado de la siguiente forma.

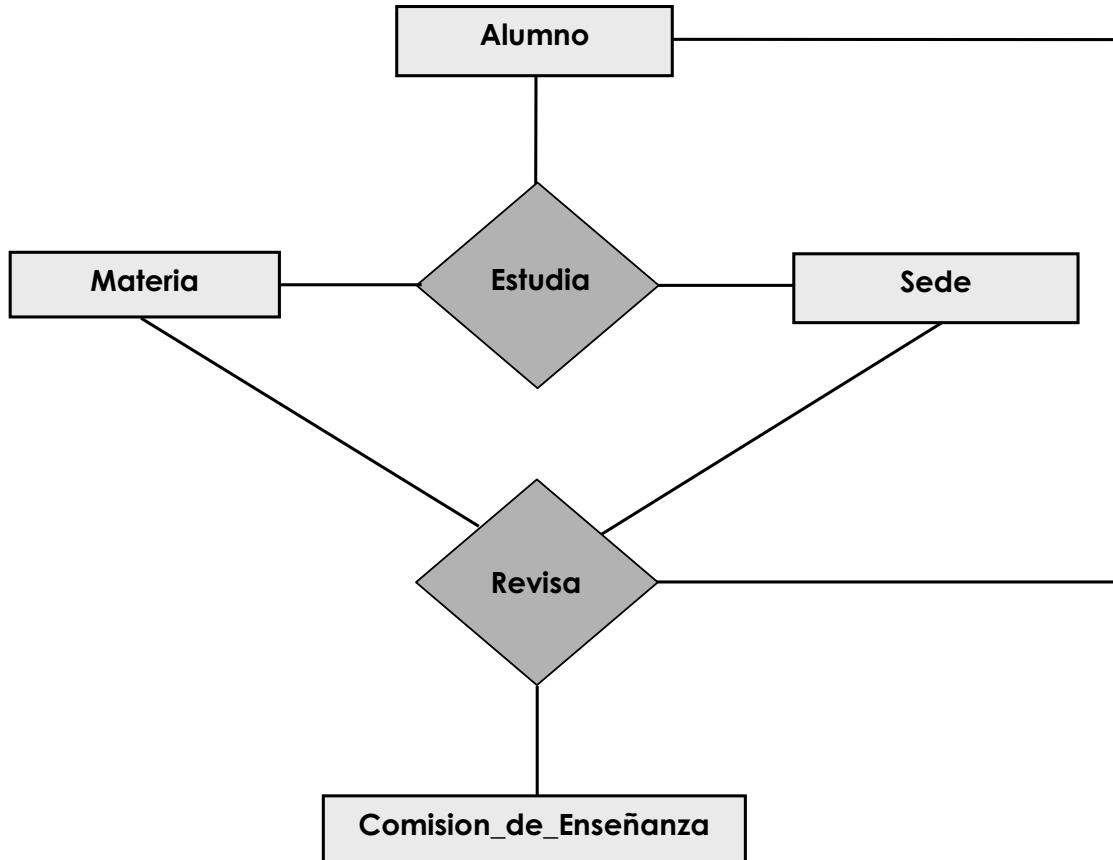


Fig. 5. Representación de una relación cuaternaria

La mejor forma de entender las relaciones anteriormente explicadas sería utilizar el concepto de agregación , que es una abstracción a través de la cual las relaciones se tratan como entidades de nivel superior.

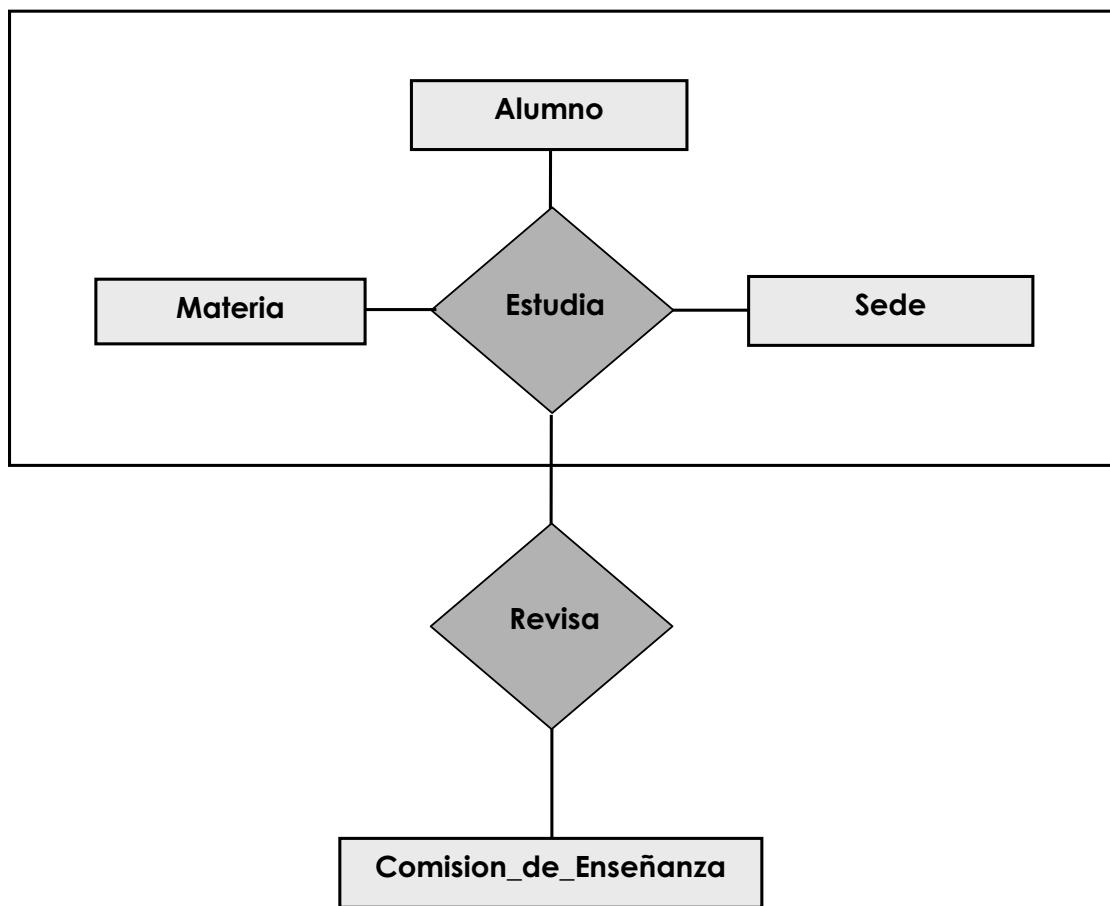


Fig. 6. Representación de una agregación

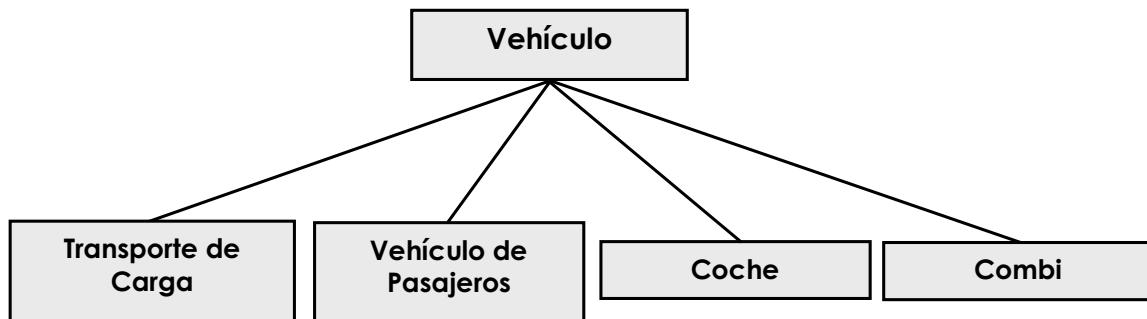


Puede profundizar más estos conceptos leyendo los capítulos 2 y 6 de "**Fundamentos de Bases de Datos**" 5ta Edición de Silberschatz – Korth y Sudarshan. Edit. Mc Graw Hill.

### Otro Ejemplo

El siguiente ejemplo tiene por objetivo terminar de comprender los conceptos vistos anteriormente y aplicarlos en un caso práctico.

Si se da el caso de que tenemos una flota de vehículos en nuestra empresa y queremos distinguirlos de otra forma podríamos recurrir a la siguiente clasificación: Por ejemplo, la entidad Vehículo puede a su vez subdividirse en Transporte de Carga, Vehículo de Pasajeros, Coche, Combi, etc. Si miramos a un determinado conjunto de ocurrencias de entidad en particular, por ejemplo, Transporte de Carga, las ocurrencias de esta entidad serán un subconjunto de la entidad Vehículo, que que todo Transporte de Carga, también es un Vehículo. A estos subconjuntos menores que dependen de uno mayor, se les suele denominar SubClases de la entidad Vehículo y a su padre, Vehículo, la SuperClase de estos subconjuntos menores.



**Fig. 7. Relaciones Clase - SubClase**

De esta forma relacionadas las Clases (en este caso una sola, llamada Vehículo), con sus SubClases, a través de relaciones Clase/Subclase. Nos quedarían entonces las siguientes relaciones de este tipo:

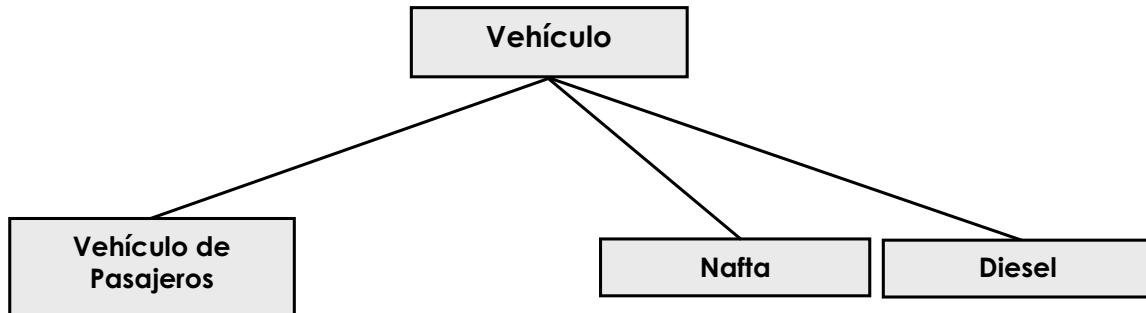
Relación Clase/SubClase
Vehículo/Transporte de Carga
Vehículo/Vehículo de Pasajeros
Vehículo/Coche
Vehículo/Combi

**Cuadro 1**

Las relaciones anteriores son conocidas como relaciones entre las SuperClase y las SubClases.

Si prestamos atención a las relaciones anteriores, y queremos hablar de una ocurrencia en particular, por ejemplo "Combi Mercedes Benz Sprinter", la misma ocurrencia aparecería en la SubClase y en la Clase, tratándose del mismo objeto real. Es decir que hay una correspondencia entre objetos reales de la SubClase y la SuperClase, siendo la relación contraria no siempre cierta. En definitiva, toda ocurrencia en una SubClase debe estar presente en la SuperClase; mientras que una ocurrencia de una SuperClase puede ser ocurrencia de una SubClase, varias SubClases o tal

vez ninguna. De esta forma "Combi Mercedes Benz Sprinter" podría estar como ocurrencia de Vehículo, y pertenecer a las SubClases Diesel y Vehículo de Pasajeros.



**Fig. 8. Relaciones SuperClase – SubClase para una ocurrencia específica**

Preste atención que se han agregado dos nuevas SubClases llamadas, Nafta y Diesel. De la misma forma, y continuando con el ejemplo anterior De esta forma "Combi Mercedes Benz Sprinter" podría estar como ocurrencia de Vehículo, y pertenecer a las SubClases Vehículo de Pasajeros, Diesel y Nafta (1 SuperClase, 3 SubClases).

Con las relaciones mencionadas anteriormente, a través de una jerarquía de SuperClases/SubClases, podemos comenzar a hablar qué ventajas tiene este tipo de representación con respecto a los atributos propios de cada entidad. A través de la **herencia**, podemos decir que una entidad de una SubClase hereda los atributos correspondientes a su SuperClase a la cual pertenece, manteniéndose tal característica para el caso de que su SuperClase se vea involucrada (podría darse el caso de que esa SuperClase sea SubClase de otra superior).

Al referirnos a un determinado grupo de SubClases, tales como Transporte de Carga, Vehículo de Pasajeros, Coche, Combi hacemos referencia a una **especialización** de la SuperClase Vehículo. Se puede dar el caso de que haya más de una especialización; en el caso anterior podríamos hablar de dos distintos tipos de especialización, una por tipo de vehículo (Transporte de Carga, Vehículo de Pasajeros, Coche, Combi) y otra dada por su tipo de combustible (Nafta o Diesel), pudiéndose utilizar de forma indistinta.

Continuando con nuestro ejemplo agregaremos atributos específicos a cada SubClase.

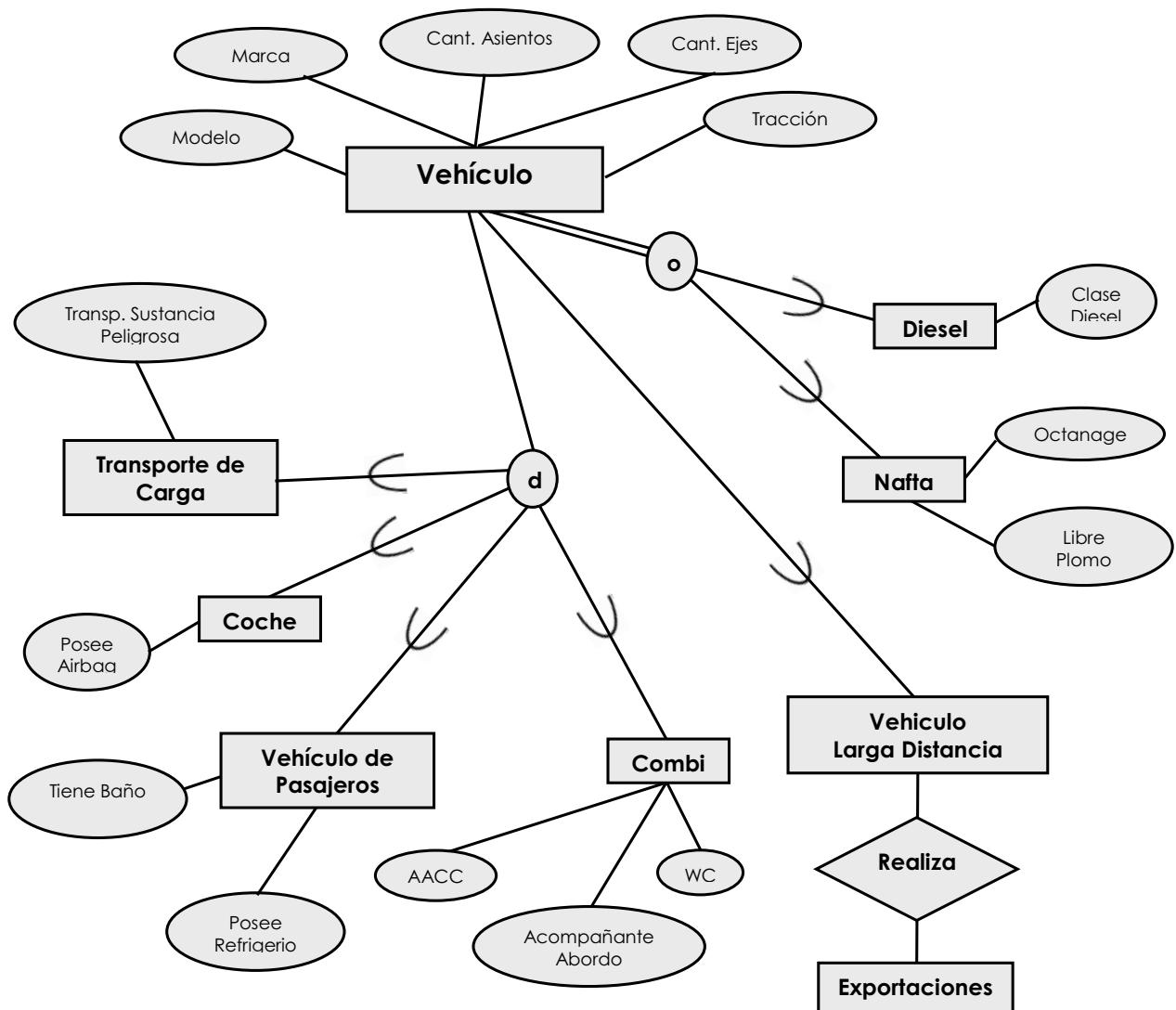


Fig. 9. Esquema de Clases con Atributos Específicos

Se pueden ver los atributos específicos de cada Entidad, tales como Posee Airbag para la Entidad Coche o los atributos Octanage y Libre Plomo para la Entidad Nafta. Pero hay otros atributos compartidos para los distintos tipos de Vehículos, tales como Marca, Modelo y Tracción.

Otra de las razones del uso de SubClases es que estas permiten establecer relaciones específicas con determinado tipo de Entidades, tal es el caso de la Entidad Vehículo Larga Distancia y la relación que se establece con Exportaciones.

De esta forma el proceso de especialización nos permite realizar las siguientes operaciones, definir conjuntos de SubClases a partir de una entidad común (SuperClase), asociar atributos específicos

a cada SubClase, volviendo muy flexible el modelo, y establecer relaciones específicas entre determinada SubClase y otras Entidades específicas, o SubClases.

Tambien se puede dar el proceso inverso desde el cual se pueden identificar características comunes y **generalizando** dichas entidades en una unica SuperClase, tal proceso se conoce como **Generalización**.

De esta forma Generalización – Especialización son procesos opuestos, o inversos.

Las SubClases y SuperClases se corresponden con entidades y se reperesentan con rectáculos.

Tanto como para especializaciones y para generalizaciones existen restricciones. En líneas generales las restricciones que aplican a ambas son comunes.

Partiendo de una SuperClase se pueden contener varias especializaciones, en el caso nuestro Tipo de Vehiculo, Tipo de Combustible y Tipo de Transportes según Distancia serian los casos.

En el caso de la pertenencia a una unica SuperClase como la especializacion de Tipo de Transporte según Distancia, no es necesario utilizar el círculo con la **d** interna.

Hay otras entidaes que requieren el uso del círculo con la **d** interna y tal es el caso de Tipo de Vehiculo. A estas SubClases se las llama definidas por **Condición**. Por ejemplo, para la SuperClase Vehiculo si tiene el atributo Tipo de Transporte se puede hablar de Condicion de pertenencia a la subclase Transporte de Carga, mediante el predicado TipoTransporte='Transporte de Carga', el cual se define como predicado de definición de la SubClase. Todas las ocurrencias de la SubClase Transporte de Carga deben satisfacer el predicado y todas las ocurrencias de la SuperClase Vehiculo, en las que el valor TipoTransporte sea el de Transporte de Carga debern pertenecer a dicha SubClase.

Si se da el caso de que todas las SubClases en una especialización tienen la condición de pertenencia al mismo atributo de la SuperClase, la especialización se denomina, **especialización definida por atributo**, llamándose este **atributo de definición de la especialización**.

Cuando no exista tal condición para determinar la pertenencia a una SuperClase, la SubClase se llamará **SubClase definida por el usuario**, en donde la pertenencia vendrá determinada por los usuarios de la Base de Datos cuando realicen una inserción de una nueva ocurrencia en la SubClase.

La letra d, dentro del círculo esta representando una restriccion de desunión, la cual especifica que las SubClases de la especialización deben estar separadas. Esto significa que una ocurrencia de la entidad puede ser miembro de como máximo una de las SubClase de la especialización.

También la letra d se utiliza para especificar que una especialización definida por el usuario debe tener la restriccción de desunión asociada, tal como puede verse en el gráfico a continuación:

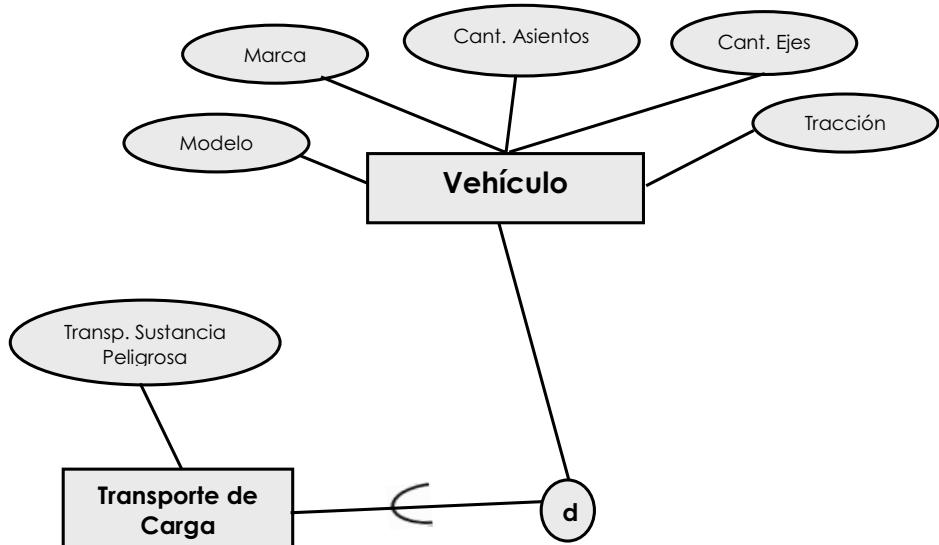


Fig. 10. Ejemplo de Desunión

Si las SubClases no son desunidas, sus conjuntos de ocurrencias se pueden solaparse, esto es, la misma ocurrencia de entidad puede ser miembro de más de una SubClase de la especialización. Este caso, que es el caso por defecto, se representa mediante una O en el círculo, como se muestra a continuación.

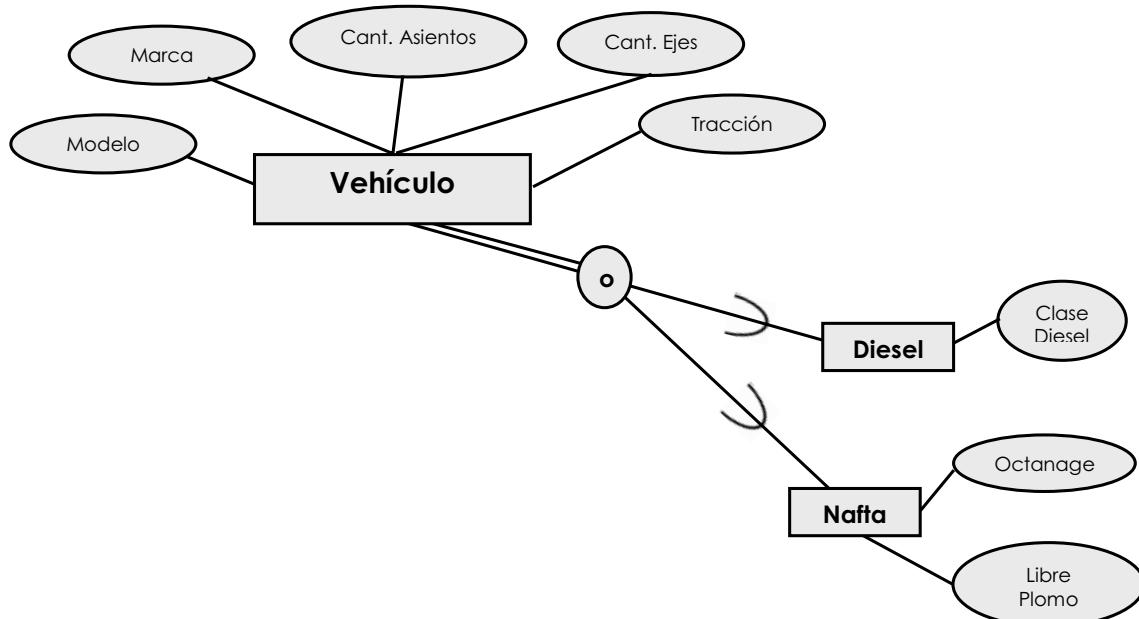


Fig. 11. Ejemplo de Unión



**F**

**Casos Prácticos Diseño  
Conceptual**

---



**Importante**

Recuerde que un mismo ejercicio puede admitir en algún punto distintas soluciones a un mismo requerimiento, por lo que a continuación se presenta una solución propuesta para cada ejercicio, pudiendo alguna de ellas diferir en algún grado de la solución desarrollada por usted. Ante esto resulta de vital importancia asegurarse de que lo diseñado por usted sea compatible con lo expresado en el relevamiento, no se olvide que el último paso de un buen diseño es testear que el modelo diseñado sea compatible con la realidad relevada.



**Importante**

Algunas de las soluciones a los ejercicios planteadas en éste capítulo se las pueden encontrar en el capítulo **G. Resoluciones Diseño Conceptual**.

**F.1. Caso 2K TRAINNING**

La empresa 2K TRAINNING ha decidido comenzar con un proceso de informatización de su sistema de manejo de cursos, el que actualmente es manejado mediante planillas de cálculo y formularios manuales. Para llevar a cabo esta tarea se ha decidido contratar a un profesional o estudiante avanzado de la carrera de sistemas de información, por lo que usted ha sido seleccionado para realizar dicha tarea.

De las reuniones mantenidas con el personal de las distintas áreas de la compañía se obtuvieron los siguientes puntos a ser contemplados por el sistema a ser desarrollado:

De los alumnos se necesita disponer de la siguiente información: DNI, Apellido y nombre, domicilio (calle y número), teléfono/s y edad. Del teléfono hay que conocer además del número su tipo (ej. Particular, Celular).

Cada carrera dictada posee su propio plan de estudios, que a su vez posee una cantidad determinada de materias.

Para cada materia se ofrecen una variedad de cursos en distintos horarios, estando cada uno a cargo de un docente. En cada curso sólo se dicta una única materia y un mismo docente puede tener a su cargo más de un curso de la misma o de diferentes materias.

Los alumnos al momento de inscribirse no eligen solamente materia, sino que eligen materia y dentro de esta el curso en el que desean cursar.

Es importante conocer las materias en las que el alumno se inscribió, la fecha de inscripción y la nota final obtenida.

De cada docente se debe conocer: DNI, apellido y nombre.

Algunos profesores tienen un supervisor (sólo uno), que es otro profesor de más antigüedad.

De cada curso es importante conocer a qué materia pertenece, quién es el docente a cargo, la carga horaria (cantidad de horas a la semana), el/los días que se dicta y el horario, tener presente que no necesariamente el horario es el mismo para todos los días en que se dicta el curso.

**Se pide**

1. Desarrollar el Modelo Conceptual, definiendo las entidades, los atributos de cada una de ellas y la cardinalidad de las relaciones.



## F.2. Caso FARMACIA LA RURAL

Usted ha sido contratado con el motivo de diseñar una base de datos para una cadena de farmacias.

De las entrevistas mantenidas con el personal de la empresa durante la etapa de relevamiento se obtuvieron los siguientes puntos a ser considerados:

Las sucursales se encuentran distribuidas en diferentes ciudades.

Cada farmacia tiene sus propios empleados y un farmacéutico a cargo de la misma.

Todo empleado se identifica con un código, guardándose además su nombre, dirección (sólo interesa calle y número) y teléfono/s.

Del personal propio además hay que poder conocer la fecha de ingreso a la empresa y su antigüedad.

Por cada ciudad hay un único farmacéutico; es decir, si en una misma ciudad existe más de una farmacia, entonces el mismo farmacéutico estaría afectado a todas las farmacias de esa ciudad.

Cada farmacia tiene a su vez su propio stock de medicamentos.

Por cada presentación del medicamento se guarda el laboratorio al que pertenece y la cantidad en existencia (stock) del mismo en cada farmacia.

Cada medicamento puede venir en muchas formas de presentación distintas pero es producido por un único laboratorio.



### Se pide

1. Desarrollar el Modelo Conceptual, definiendo las entidades, los atributos de cada una de ellas y la cardinalidad de las relaciones.

**F.3. Caso TODO LIBRO**

La empresa TODO LIBRO se dedica a la compra-venta y distribución de libros. Posee una casa central y tres sucursales, situadas en Paraná, Rosario, y la última en La Plata. La Casa Central está ubicada en la Capital Federal.

Cada sucursal cuenta con un número determinado de vendedores que brindan una atención personalizada a cada cliente.

La empresa se destaca por tener libros especializados en distintas ramas del conocimiento, tales como matemáticas, ciencias médicas, deportes, astrología, jardinería, cocina, historia, animales, teología, etc.

Los empleados pertenecen a una sucursal por un tiempo determinado, pudiéndose dar el caso de que un vendedor sea trasladado de una sucursal a otra. Es importante conocer las distintas sucursales por las que pasa un vendedor.

La empresa TODO LIBRO recibe de cada Editorial los libros que compra mensualmente.

Al final de cada mes, el gerente de la central Buenos Aires, Ernesto Zapata, realiza un ranking de ventas por empleado en cada una de las sucursales, recibiendo el mejor vendedor de cada sucursal un estímulo económico, por el desempeño alcanzado.

Los datos que se necesitan guardar de un libro son el tema, el/los autores, la fecha de publicación, la cantidad de páginas, y la editorial del mismo.

También se realiza un ranking anual de ventas por sucursal determinando la que más ventas realizó en un período de un año. Esto permite mejorar las comisiones percibidas por los vendedores.

**Consideraciones adicionales**

- ✓ Cada libro tiene un código único de identificación.
- ✓ No todos los títulos son comercializados por todas las editoriales.
- ✓ Las ventas son registradas mediante la emisión de una factura en la cual se incluyen los siguientes datos: Número de factura, fecha, cliente, sucursal, vendedor, libro, editorial, cantidad, importe.

**Se pide**

1. Desarrollar el Modelo Conceptual, definiendo las entidades, atributos de cada una de ellas y las relaciones y su cardinalidad.

**F.4. Caso CHICHO PETS**

CHICHO PETS es una veterinaria de barrio que se dedica al cuidado de mascotas.

Desde sus comienzos, la política de la empresa ha sido atender a los animales como si fueran humanos. Tal hecho nos brinda un acercamiento a la forma de pensar de su dueño.

Dado que la veterinaria esta creciendo y volviéndose cada vez más conocida, ya no es posible llevar manualmente la registración de las operaciones. Es por tal motivo que, Ayrton, dueño de CHICHO PETS, tiene intenciones de utilizar una base de datos para registrar toda su actividad comercial.

CHICHO PETS presta atención a diversos tipos de mascotas: aves, gatos y peces, pero su principal interés reside en los canes.

Entre las cuestiones pensadas para mejorar su gestión se encuentra el hecho de brindar una atención personalizada a cada mascota. Es por eso que Ayrton piensa en la idea de ofrecer una mutual y gestionar la afiliación de socios. De esta forma, cada socio abonaría un arancel mensual que le posibilitaría acceder a un set de servicios variados tales como vacunas, internaciones, belleza y cuidados intensivos.

Cada plan posee una cierta cantidad de servicios cubiertos por el mismo, por lo que es importante conocer los servicios cubiertos por cada plan.

Dependiendo de cada plan, se ofrecen diversos descuentos en los servicios prestados por CHICHO PETS. En el caso del plan Premium, este cubre todos los costos generados por un animal al momento de su atención.

El sistema debe permitir la posibilidad de llevar un historial de visitas por mascota, de tal forma que permita realizar un seguimiento completo del animal desde que fue atendido por primera vez en CHICHO PETS. Los datos que se desea guardar son: Identificador de la mascota, fecha de atención, estudios (servicios) que se le realizaron, diagnóstico y profesional que la atendió.

Entre los datos que se guardan cada vez que se da de alta un socio, se encuentran el nombre de la mascota, su fecha de nacimiento y nombre de su dueño.

En CHICHO PETS se sostiene que si a una mascota se la alimenta inadecuadamente la misma puede llegar a enfermarse. Es por tal motivo que adicionalmente también se dedica a la elaboración de dietas especiales para animales.

La empresa cuenta con proveedores de alimentos especializados en mascotas, un mismo alimento puede ser provisto por más de un proveedor.

"Compramos alimentos de la mejor calidad, y elaboramos según su mascota el alimento mas adecuado...", nos cuenta Ayrton. Cada alimento contiene una composición determinada de vitaminas y minerales.

De los alimentos comercializados es importante conocer en detalle su composición (insumos que lo forman) como así también la medida de cada uno de estos elementos. Tener presente que los elementos poseen una unidad de medida (litros, milímetro cúbico, etc.) y que no necesariamente todos se encuentran expresados en la misma unidad.

### **Consideraciones adicionales**

En la visita de cada mascota se registran datos como la fecha de la consulta, su peso, estudio realizado y el estado general de la mascota.

Los Estados Generales de cada mascota pueden ser:

- ✓ Sano.
- ✓ Enfermo.
- ✓ Comprometido.
- ✓ Delicado.

Como el monto de la cuota mensual para cada plan puede variar en el tiempo es importante conocer como varió la misma.

Los planes alimentarios son diseñados en forma personalizada para cada mascota y pueden variar con el tiempo por lo que es importante conocer los distintos planes que se armó para la mascota.

No necesariamente el número de historia clínica coincide con el de mascota.

La composición de un alimento no necesariamente es la misma para todos los proveedores de dicho alimento.



#### **Se pide**

1. Desarrollar el Modelo Conceptual, definiendo las entidades, los atributos de cada una de ellas y la cardinalidad de las relaciones.

**F.5. Caso LA CLÍNICA**

Una clínica de la ciudad presenta la necesidad de manipular y almacenar todo tipo de información que pueda estar relacionada con ella.

La clínica cuenta con una estructura física de tres torres, de las cuales dos son para hospitalización y cirugías y una para consulta. A su vez las torres de hospitalización se encuentran divididas en habitaciones mientras que la de consulta en consultorios.

Por cada habitación o consultorio es importante conocer su ubicación (torre, piso y número), longitud (ancho y largo).

A su vez por cada habitación también se necesita conocer que comodidades posee, a saber:

- ✓ Cantidad de baños.
- ✓ Si posee clóset.
- ✓ Cantidad de camas.
- ✓ Si posee ventana.

Es importante también conocer quien está ocupando cada habitación, desde cuando lo hace y por cuanto tiempo está previsto que la ocupe.

En cuanto a los consultorios, estos son utilizados para atender las consultas realizadas por los distintos servicios prestados por la clínica (cardiología, pediatría, fisioterapia, etc.).

Respecto de los pacientes cada uno es identificado con un número único y los datos guardados son su nombre, fecha de nacimiento, edad, teléfono/s y prepaga que posee.

Todo paciente posee una historia clínica (el número de historia coincide con el de paciente) y en esta se registran todas las consultas realizadas por el paciente, guardándose la fecha de consulta, el profesional que lo atendió, el motivo de la consulta (servicio) y el diagnóstico.

De cada profesional se guarda su nombre, teléfono particular, teléfono para urgencias y especialidad.

Una característica importante es que un doctor puede dirigir un área específica como ser por ejemplo pediatría o cardiología, por lo que para un jefe de área es importante saber que personal tiene a su cargo.

De la torre también es importante conocer su dirección, la cuál se compone por calle, número, piso y departamento.

**Se pide**

1. Desarrollar el Modelo Conceptual, definiendo las entidades, los atributos de cada una de ellas y la cardinalidad de las relaciones.



## **Resoluciones Casos Prácticos**

### **Diseño Conceptual**

---

#### **✓ Objetivos**

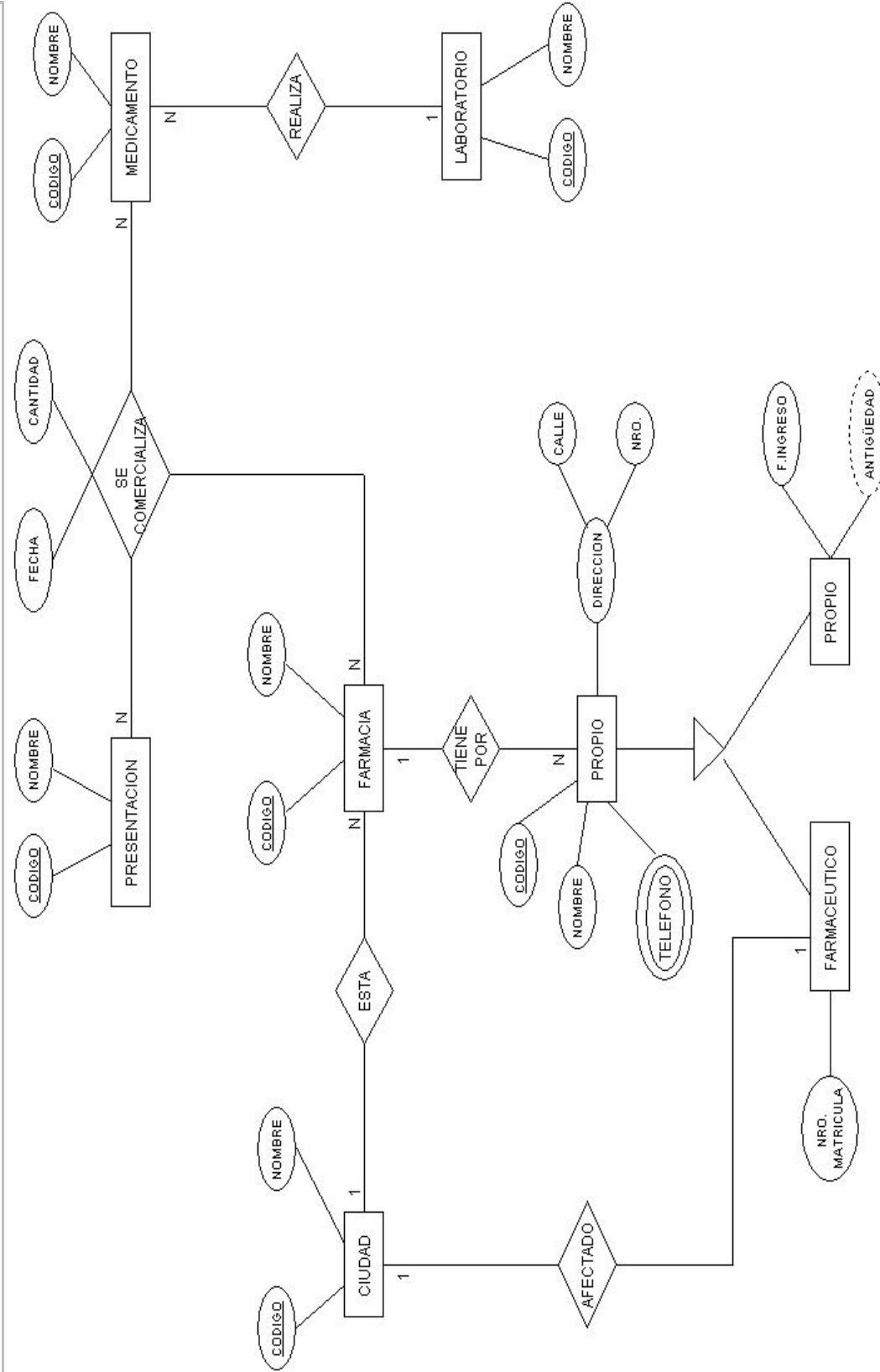
- ✓ Resolver los ejercicios que consideramos relevantes.
- ✓ Mostrar una solución posible.
- ✓ Delinear una forma de resolución y su propósito.



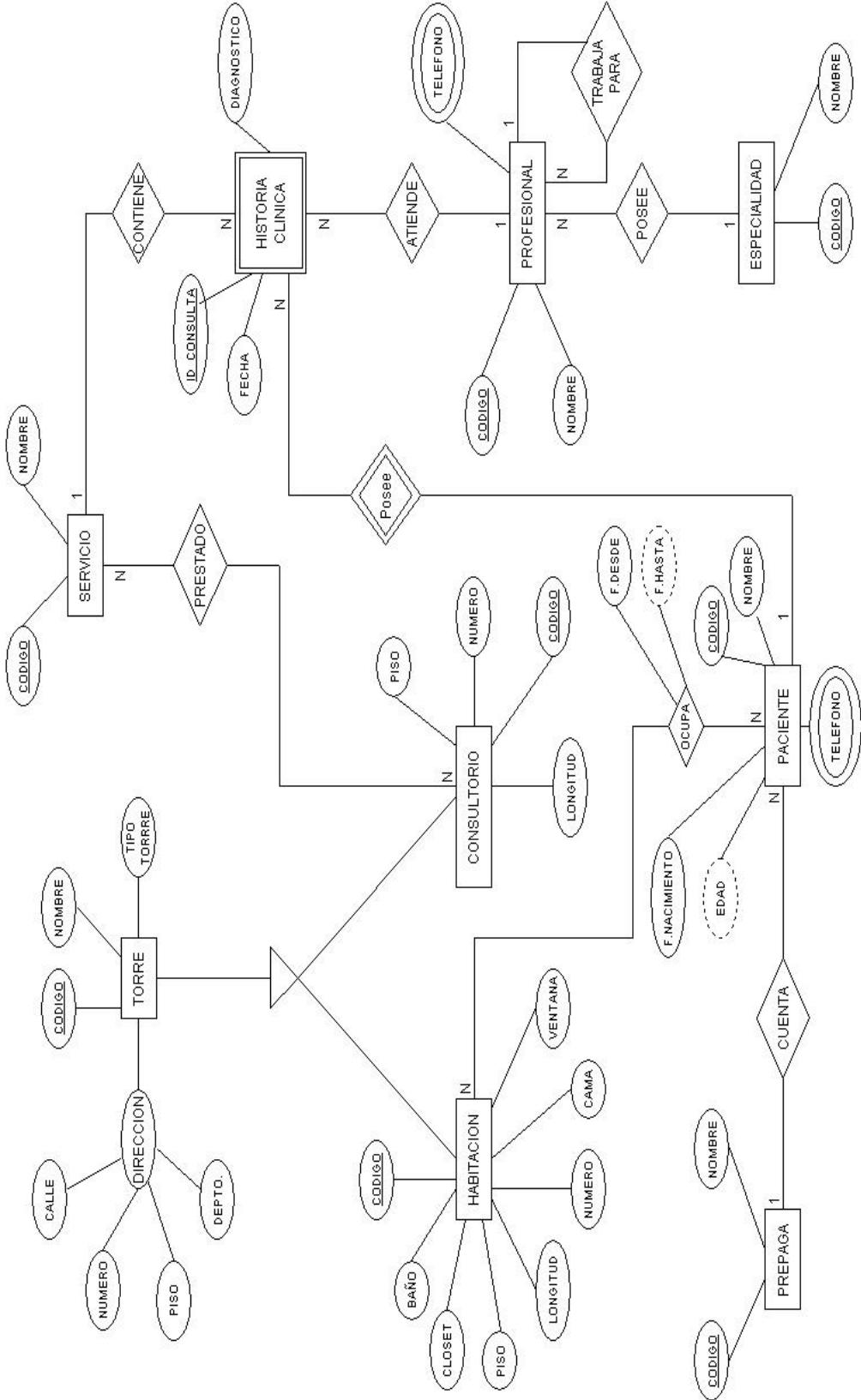
**Importante**

Recuerde que un mismo ejercicio puede admitir en algún punto distintas soluciones a un mismo requerimiento, por lo que a continuación se presenta una solución propuesta para cada ejercicio, pudiendo alguna de ellas diferir en algún grado de la solución desarrollada por usted, ante esto resulta de vital importancia asegurarse de que lo diseñado por usted sea compatible con lo expresado en el relevamiento, no se olvide que el último paso de un buen diseño es testear que el modelo diseñado sea compatible con la realidad relevada.

✓✓✓✓ G.1. Resolución Caso FARMACIA LA RURAL



✓✓✓✓✓ G.2. Resolución Caso LA CLÍNICA



# H

## Diseño Lógico

---

### Objetivos

- ✓ Comprender las diferencias respecto del modelado físico.
- ✓ Entender el proceso de normalización.
- ✓ Comprender las formas normales y su aplicación.
- ✓ Utilizar el concepto de cardinalidad según los tipos de relaciones existentes.

## H. Diseño Lógico

El modelado conceptual visto en el capítulo anterior le ha permitido al diseñador entender la semántica del problema analizado. El DER, producto de esa etapa es una herramienta que sirve al humano para comprender los elementos y relaciones que existen en el sistema. Tiene un alto nivel de abstracción y su desarrollo es independiente del sistema de gestión de base de datos que se utilice.

El paso siguiente a seguir es asociar ese modelo generado con las especificaciones que son necesarias para implementar el modelo en una computadora. En esta etapa es necesario excluir una serie de situaciones anómalas que perjudican el diseño. Esto se logra a través de un proceso denominado Normalización, que consiste en aplicar pasos sucesivos en cada uno de los cuales se elimina una situación no deseada.

En relación con cuál ha sido el nivel de detalle del DER confeccionado en la etapa anterior, estaremos más próximos o no al modelo normalizado que tenemos por objetivo en esta etapa.

Existe una relación estrecha entre los elementos analizados en la etapa del Diseño Conceptual y la de Diseño Lógico que estamos viendo. Existe un consenso generalizado en que la correlación es la que se muestra en la siguiente tabla de equivalencias.

Diseño Conceptual	Diseño Lógico
Entidad	Tabla
Relación	Relación
Atributo	Campo
Tupla	Registro / Fila

**Cuadro 1**

Se dice que una estructura de datos está normalizada cuando tiene las siguientes propiedades:

1. Cada entrada en la tabla representa un dato elemental. No hay atributos multivaluados.
2. Cada columna contiene datos que significan lo mismo.
3. A cada columna se le asigna un nombre diferente.
4. Todas las filas son distintas. No se permiten filas duplicadas. Esto se consigue a través de la generación de una clave primaria que identifica únicamente a cada fila de la tabla.

A continuación pasamos a analizar el proceso de Normalización.

### Proceso de Normalización

Partiremos del tradicional ejemplo de facturación, que con fines didácticos lo simplificaremos. Supongamos que el objetivo del sistema es emitir una factura con las siguientes características:

Datos de la Empresa		Nro. Factura: ( <b>NFact</b> ) Fecha: ( <b>Fecha</b> ) Código de Cliente: ( <b>CodCli</b> ) Nombre del Cliente: ( <b>NomCli</b> )		
Código Producto	Descripción	Cantidad	Precio Unitario	Precio Renglón
( <b>Cprod</b> )	( <b>Descri</b> )	( <b>Canti</b> )	( <b>PUnit</b> )	( <b>PcioR</b> )
Total Factura				( <b>PcioT</b> )

**Fig. 1. Ejemplo de Factura**

Las variables que tenemos que almacenar son (entre paréntesis se agregan los nombres con que van a ser reconocidos por el sistema):

- ✓ el Número de Factura (**NFact**)
- ✓ la Fecha (**Fecha**)
- ✓ el Código de Cliente que realizó la compra (**CodCli**)
- ✓ el Nombre de dicho cliente (**NomCli**)
- ✓ el Código del Producto vendido (**CProd**)
- ✓ la Descripción de ese producto (**Descri**)
- ✓ la Cantidad vendida (**Canti**)
- ✓ el Precio Unitario al que se vendió (**PUnit**)
- ✓ el Precio Total de ese renglón (**PcioR**)
- ✓ El Precio Total de la factura (**PcioT**)

Algún ansioso implementador podría sentirse tentado a crear una sola tabla con todos estos datos. Obtendría como resultado la siguiente tabla (a la que le agregamos algunos registros a modo de ejemplos):

<b>NFact</b>	<b>Fecha</b>	<b>CodCli</b>	<b>NomCli</b>	<b>CProd</b>	<b>Descri</b>	<b>Canti</b>	<b>PUnit</b>	<b>PcioR</b>	<b>PcioT</b>
1	12/01/2006	3	Carlos	35	Peras	6	3	18	59
1	12/01/2006	3	Carlos	37	Uvas	1	5	5	59
1	12/02/2006	3	Carla	39	Kiwi	5	10	50	59
1	12/01/2006	4	Carlos	40	Sandía	2	3	6	59
2	13/01/2006	4	Pablo	37	Peras	5	3	15	58
2	13/01/2006	4	Pablo	39	Kivi	3	11	33	58
2	13/01/2006	4	Pablo	44	Tomates	2	5	10	58
3	14/01/2006	3	Carlitos	35	Peras	4	3	12	12

**Cuadro 2**

Supongamos que los datos son capturados en línea renglón por renglón, sin verificar la consistencia . El hecho de cargar todos los datos en una tabla y la modalidad de captura (poco probable en la

actualidad), puede producir una serie de inconsistencias, algunas de las cuales han sido volcadas en el ejemplo.

Realicemos un comentario previo al análisis de las inconsistencias. Vemos que el dato PcioR surge de multiplicar Canti por PUnit. De manera similar, el dato PcioT surge de sumar todos los PcioR de cada factura. Estos datos son los que denominamos en la etapa de diseño Conceptual "Atributos Derivables". Desde un punto de vista estricto, estos atributos no deben ser almacenados, ya que en el momento en que se necesiten pueden ser obtenidos por cálculo a partir de los datos que los generan. Con esta definición, en el cuadro de ejemplo, se pueden eliminar las columnas PcioR y PcioT. Quedaría entonces:

NFact	Fecha	CodCli	NomCli	CProd	Descri	Canti	PUnit
1	12/01/2006	3	Carlos	35	Peras	6	3
1	12/01/2006	3	Carlos	37	Uvas	1	5
1	12/02/2006	3	Carla	39	Kiwi	5	10
1	12/01/2006	4	Carlos	40	Sandía	2	3
2	13/01/2006	4	Pablo	37	Peras	5	3
2	13/01/2006	4	Pablo	39	Kivi	3	11
2	13/01/2006	4	Pablo	44	Tomates	2	5
3	14/01/2006	3	Carlitos	35	Peras	4	3

Cuadro 3

Analicemos ahora algunas anomalías que presenta el ejemplo.

1. La factura 1 tiene 2 fechas: 12/01/2006 y 12/02/2006. Esto puede deberse a errores de tipeo.
2. La misma factura tiene 2 Códigos de Cliente diferentes (3 y 4) cuando en realidad cada compra es hecha por uno solo. No tenemos certeza sobre cual es el que corresponde aunque pudiera suponerse que es el cliente 3 porque es el que aparece en más registros.
3. El CodCli 4 tiene distintos nombres en 2 registros diferentes: Carlos y Pablo.
4. El CodCli 3 tiene distintos nombres en 2 registros diferentes: Carlos y Carlitos.
5. El CProd 37 tiene 2 descripciones diferentes: Uvas y Peras.
6. El CProd 39 tiene 2 formas de descripción diferentes: Kiwi y Kivi.
7. El CProd 39 tiene 2 PUnit diferentes: 3 y 5. Sin embargo, esta diferencia puede ser aceptada por razones propias del negocio. Por ejemplo, pueden haber precios diferentes para distintos clientes o por razones de cantidad, etc.

Estas anomalías que se mencionan son las que se tratan de eliminar con el proceso de normalización. Trataremos aquí las primeras tres etapas del proceso. Existen algunas formas normales más complejas, las cuales no van a ser analizadas en este libro.

**Primera Forma Normal (1FN)**

Las anomalías mencionadas en los puntos 1 y 2 (fechas y clientes distintos) ocurren porque la factura presenta 2 partes bien diferenciadas: por un lado la cabecera, que incluye en Número de la Factura, la Fecha y los datos del Cliente, y por el otro cada renglón que incluye los demás datos. Vemos que en una factura puede haber varios CProd, con sus cantidades y precios unitarios. Esto hace que por cada renglón de venta se deban agregar los datos de la cabecera.

Nfact	Fecha	CodCli	NomCli
1	12/01/2006	3	Carlos
2	13/01/2006	4	Pablo
3	14/01/2006	3	Carlitos

**Cuadro 4**

Desde un punto de vista estricto, esto no debe ser así, ya que la cabecera es la misma para todos los renglones de venta. Si lo cargamos más de una vez corremos el riesgo de que sucedan situaciones como las mencionadas. Al conjunto de atributos Código de Producto, Cantidad, Precio Unitario y Descripción los hemos llamado en la etapa de Diseño Conceptual Atributos Multivaluados o Grupo Repetitivo.

CProd	Descri	Canti	PUnit
35	Peras	6	3
37	Uvas	1	5
39	Kiwi	5	10
40	Sandía	2	3
37	Peras	5	3
39	Kivi	3	11
44	Tomates	2	5
35	Peras	4	3

**Cuadro 5**

La solución al problema pasa entonces por adaptar las características de la realidad (las dos partes de la factura) al modelo que se está realizando. Entonces lo que se debe hacer es descomponer la estructura de datos en dos, una de las cuales incluirá los datos atómicos (relacionados con la cabecera) y la otra los atributos multivaluados (correspondientes a los renglones). El modelo quedaría entonces conformado por las siguientes tablas:

A la primera la denominamos CabeceraFact

Y a la segunda RenglónFact

Notemos que resuelto de esta manera, hemos perdido la correspondencia entre renglones de venta y la cabecera a la que corresponden. Para evitarlo, el modelo exige incluir en la segunda tabla (RenglonFact) la clave primaria de la primera (NFact). De esta manera, el modelo en 1FN quedaría:

Nfact	Fecha	CodCli	NomCli
1	12/01/2006	3	Carlos
2	13/01/2006	4	Pablo
3	14/01/2006	3	Carlitos

NFact	CProd	Descri	Canti	PUnit
1	35	Peras	6	3
1	37	Uvas	1	5
1	39	Kiwi	5	10
1	40	Sandía	2	3
2	37	Peras	5	3
2	39	Kivi	3	11
2	44	Tomates	2	5
3	35	Peras	4	3

Cuadro 6

Con lo visto hasta aquí podemos decir que una estructura de datos está en Primera Forma Normal si no posee Atributos Multivaluados.

La solución a una estructura que los posea es separarla en dos tablas, la primera de las cuales posee todos los atributos atómicos, y la segunda todos los atributos multivaluados, a la que debe agregarse la clave primaria de la primera tabla.

### Segunda Forma Normal (2FN)

Analicemos ahora las anomalías mencionadas en los puntos 5 y 6 (descripciones de Productos diferentes). Se dan porque se almacena la Descripción del producto con los datos de la factura, cuando en realidad este es un atributo exclusivo del producto, independientemente de las facturas en las que se venda.

Nuevamente, la solución es adaptar la realidad al modelo, y generar una tabla específica de Productos (a la que llamaremos Prod), en los que se almacenen sus atributos. El nuevo modelo en 2FN quedaría:

Fact	CProd	Canti	PUnit		CProd	Descri
1	35	6	3		35	Peras
1	37	1	5		37	Uvas
1	39	5	10		39	Kiwi
1	40	2	3		40	Sandía
2	37	5	3		44	Tomates
2	39	3	11			
2	44	2	5			
3	35	4	3			

Cuadro 7

Podría pensarse que el campo precio unitario es también un atributo que depende del producto. La realidad es un poco más compleja, ya que no solo depende del producto sino de otras circunstancias, como por ejemplo el lapso en el que se mantiene el mismo precio, los valores diferenciales de acuerdo a ciertos tipos de clientes (precios mayoristas y minoristas), etc. En el ejemplo, los productos 37 y 39 se vendieron a dos valores diferentes; las circunstancias de esa diferencia puede ser porque al cliente 3 se le cobró un precio y al 4 otro, o porque el 12 de enero costaba 10 pesos y el 13 de enero 11. Esto amerita que el tema precio unitario lo tratemos de manera separada mas adelante.

Retomando el eje del apartado, diremos que una estructura de datos está en Segunda Forma Normal si está en Primera Forma Normal y todos sus campos no clave dependen íntegramente de la clave primaria.

Una estructura que está en 1FN y no en 2FN se corrige dividiendo la tabla que tiene campos dependientes parcialmente de la clave primaria en dos, la primera de las cuales tiene los campos que dependen totalmente de la clave primaria, y la segunda el campo dependiente parcialmente de la clave primaria, con la parte de la clave de la cual depende.

Analicemos estos dos últimos párrafos. Decimos en el primero: "una estructura de datos está en Segunda Forma Normal (...) y todos sus campos no clave dependen íntegramente de la clave primaria.". En la estructura original de la Tabla RenglonFacto existe un campo (Descri), que no depende de toda la clave, sino exclusivamente del campo CProd, ya que el producto se llama de la misma manera si el CProd es el mismo, independientemente de que se venda en la Factura 1 o la 2. A esto es lo que llamamos Dependencia parcial de la clave.

En el segundo párrafo decimos: "Una estructura que está en 1FN y no en 2FN se corrige dividiendo la tabla que tiene campos dependientes parcialmente de la clave primaria en dos, la primera de las cuales tiene los campos que dependen totalmente de la clave, y en la segunda el campo dependiente parcialmente de la clave, con la parte de la clave de la cual depende".

La tabla que tiene un campo dependiente parcialmente de la clave es RenglonFact original (Descri); la tabla resultado que tiene los campos que dependen totalmente de la clave es la nueva versión de RenglonFact (sin Descri), y la tabla resultado que tiene el campo dependiente parcialmente de la clave, con la parte de la clave de la cual depende, es la tabla Prod.

### Tercera Forma Normal (3FN)

Nos resta aún analizar el problema que se menciona en los puntos 3 y 4 (problemas en el Código y Nombre de Cliente). Es similar al caso del producto, ya que se produce al almacenar el nombre del cliente con los datos de la factura, cuando en realidad este es un atributo exclusivo del Cliente, independientemente de las facturas que se le emitan.

Como intuirán, la solución es adaptar el modelo a la realidad y generar una tabla específica de Clientes (a la que llamaremos Cli), en la que se almacenen sus atributos.

El nuevo modelo en 3FN quedaría:

Nfact	Fecha	CodCli	CodCli	NomCli
1	12/01/2006	3	3	Carlos
2	13/01/2006	4	4	Pablo
3	14/01/2006	3		

**Cuadro 8**

Diremos entonces que una estructura de datos está en Tercera Forma Normal si se encuentra en 2FN y no existe dependencia transitiva.

Una estructura que está en 2FN y no en 3FN se corrige dividiendo la tabla que tiene campos con dependencia transitiva en dos nuevas tablas, la primera de las cuales tiene los campos que dependen totalmente de la clave primaria, y la segunda el campo que tenía dependencia transitiva, con el atributo del cual depende.

En nuestro ejemplo, el campo con dependencia transitiva es el NomCli, ya que este depende del CodCli y no del NFact, que recordemos, es la clave primaria de la tabla. Se genera entonces una nueva tabla Cli, que tiene los datos relacionados con el cliente, y su clave primaria será entonces CodCli.

Luego de aplicar los tres primeros pasos del proceso de Normalización, el modelo definitivo es:

CabeceraFact (#NFact, Fecha, CodCli)  
 RenglonFact (#NFact, #CProd, Canti, Punit)  
 Prod (#CProd, Descri)  
 Cli (#CodCli, NomCli)



Los campos con # y **negrita** representan la clave primaria de la tabla.

Resumiendo los tres primeros pasos del proceso de normalización, diremos que:

- ✓ Una estructura de datos se encuentra en 1FN si no poseen atributos que puedan repetirse para la misma clave primaria.
- ✓ Una estructura de datos se encuentra en 2FN si se encuentra en 1FN y todos sus atributos dependen funcionalmente de la totalidad de la clave primaria.
- ✓ Una estructura de datos se encuentra en 3FN si se encuentra en 2FN y no existe dependencia transitiva (ninguno de sus atributos dependen de un campo no clave).

#### Cuadro Resumen FN

Además de los pasos enunciados anteriormente que forman propiamente dicho el proceso de normalización, durante la etapa de diseño lógico ocurre que el diseñador además puede tener la necesidad de modelar situaciones que pueden llevar un tratamiento diferente según la realidad observada, dentro de estas situaciones podemos encontrar la representación de históricos, de relaciones recursivas y de relaciones definidas en el DER con cardinalidad N a N. Recuerde que estos temas ya fueron introducidos en el capítulo anterior (cuando hablamos de diseño conceptual) por lo que no realizaremos una introducción a los mismos sino que sólo veremos la representación de cada uno de estos puntos en un modelo normalizado.

#### Relaciones N a N

En el modelo normalizado las relaciones muchos a muchos NO existen, sino que este tipo de relaciones que sí existen en el DER producto del diseño conceptual, en un modelo normalizado se transforman en una nueva tabla que se conoce con el nombre de tabla asociativa o relacional, y cuya finalidad es la de vincular a las dos tablas que dieron origen a la relación de tipo N a N.

Ahora veamos esto mediante un ejemplo, para lo cuál utilizaremos el caso de la factura utilizado anteriormente con la siguiente modificación: No todos los productos son vendidos a todos los clientes, esto significa que pueden existir productos que son vendidos a todos los clientes y otros

que quizás por una cuestión de exclusividad son vendidos únicamente a un cliente o grupo de clientes.

En la etapa de diseño conceptual se detectó la existencia de una relación entre el cliente y los productos definiéndose la cardinalidad de la misma en muchos a muchos, ya que un mismo producto puede ser vendido a más de un cliente y más de un cliente puede adquirir el mismo producto.



**Fig. 2. Representación en el DER**

Como en el modelo normalizado no existen las relaciones N a N es que esta relación se transforma en una nueva tabla relacional o asociativa que vincula a la tabla cliente con la de producto y a su vez permite cumplir con la restricción de que no necesariamente un producto sea vendido a todos los clientes. Los campos de esta nueva tabla serán como mínimo los campos que forman la clave primaria de cada una de las dos tablas y cualquier otro campo propio de dicha relación, siendo la clave primaria de la nueva tabla la conjunción de todos los campos que forman la clave primaria de las dos tablas.

La estructura de datos normalizada será entonces:

CProd	Descri	CodCli	NomCli	CProd	CodCli
35	Peras	3	Carlos	35	3
37	Uvas	4	Pablo	35	4
39	Kivi			37	3
40	Sandía			39	4
44	Tomates			40	4
				44	3
				44	4

**Cuadro 9**

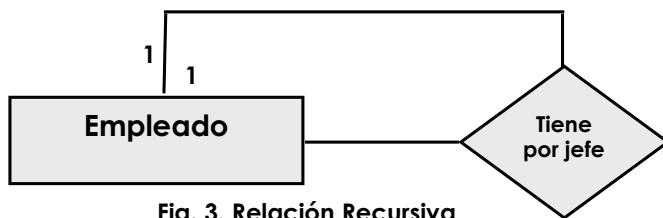
Como se puede observar la nueva tabla a la que llamaremos ProductoCliente, nos permite definir a qué cliente o grupo de clientes se le puede vender qué producto o conjunto de productos.

## Relaciones Recursivas

Cuando hablamos de relaciones recursivas en el capítulo de diseño conceptual vimos que la cardinalidad de este tipo de relaciones puede ser de 1 a 1 o de 1 a N. En el diseño lógico la cardinalidad de la relación determina la manera en la que la misma debe ser modelada, es decir la estructura de datos modelada para representar relaciones con cardinalidad 1 a 1 no es la misma que las relaciones con cardinalidad 1 a N. Veamos cada una de las situaciones por separado.

## **Cardinalidad 1 a 1**

Para poder explicar este tema supongamos que se nos presenta la siguiente realidad, usted se encuentra diseñando un sistema para el área de recursos humanos y uno de los puntos a ser cubiertos por el sistema es la necesidad de poder conocer quién es el jefe de cada uno de los empleados que posee la compañía, interesando solamente conocer el jefe actual de cada empleado, también sabe que un empleado sólo tiene un único jefe directo. Del diseño conceptual surge lo siguiente:



**Fig. 3. Relación Recursiva**

Una alternativa que el diseñador podría plantearse es al de generar la estructura de datos es:

## Empleado

<u>CodEmp</u>	Nombre
1	Luis
2	Gerardo
3	Luciana
4	Luisa
5	Fernando

## Cuadro 10

**Jefe**

<u>CodJefe</u>	<u>Nombre</u>
1	Luis
4	Luisa
2	Gerardo

**Cuadro 11****Relación Jefe – Empleado**

<u>CodJefe</u>	<u>CodEmp</u>
1	2
2	3
2	4
4	5

**Cuadro 12**

Observemos que el modelo generado posee ciertas inconsistencias, analicemos ahora algunas anomalías que presenta el ejemplo:

1. Los jefes a su vez son empleados por lo que ya existe un registro con sus datos en la tabla empleado, lo que hace que la tabla Jefe sea redundante. Recuerde que uno normaliza con el fin de eliminar la redundancia innecesaria.
2. Todo empleado tiene un único jefe directo por lo que la tabla de relación jefe – empleado no es necesaria, ya que de existir se estaría permitiendo la posibilidad de que un empleado tenga mas de un jefe, lo cuál se contradice con la realidad observada.

La solución correcta al problema planteado es agregar en la tabla empleado un nuevo campo llamado CodJefe, el cuál contenga el código de empleado correspondiente al jefe del cuál depende el empleado.

**Empleado**

<u>CodEmp</u>	<u>Nombre</u>	<u>CodJefe</u>
1	Luis	
2	Gerardo	1
3	Luciana	2
4	Luisa	2
5	Fernando	4

**Cuadro 13**

En este caso no existe redundancia y se cumple con la consigna de que un empleado tenga únicamente un jefe directo.

### Cardinalidad 1 a N

Para poder explicar este tipo de relación supongamos que se nos presenta una situación como la detallada a continuación:

Se está diseñando un sistema para posicionamiento geográfico y de la realidad observada surge que existen países los cuales limitan con otro/s país/es. En el DER definido en la etapa conceptual esta situación se representó de la siguiente manera:

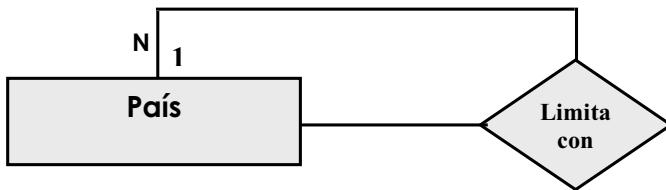


Fig. 4. Relación Recursiva 1 a N

En la etapa de diseño lógico la solución para modelar este tipo de relaciones es crear una segunda tabla que contenga el grupo repetitivo, en nuestro ejemplo crearemos una nueva tabla a la cual llamaremos PaísLimita y la que contendrá todos los países que limitan con un determinado país.

La estructura de datos entonces será:

### País

CodPaís	Nombre
1	Argentina
2	Uruguay
3	Chile
4	Paraguay
5	Bolivia
6	Perú
7	Brasil

Cuadro 14

**PaisLimita**

<u>CodPais</u>	<u>CodPaisLimita</u>
1	2
1	3
1	4
1	5
1	6
1	7

**Cuadro 15****Histórico**

Cuando hablamos de la 2FN en uno de los párrafos decíamos: “podría pensarse que el campo precio unitario es también un atributo que depende del producto. La realidad es un poco más compleja, ya que no solo depende del producto sino de otras circunstancias, como por ejemplo el lapso en el que se mantiene el mismo precio, los valores diferenciales de acuerdo a ciertos tipos de clientes (precios mayoristas y minoristas), etc. (...)”.

Del párrafo anterior se deduce que pueden presentarse diferentes realidades, lo cuál tendrá como consecuencia quizás la necesidad definir distintas estructuras de datos que sean capaces de contemplar la realidad observada. A fines prácticos y a modo de ejemplo para explicar el tratamiento de históricos es que sólo consideraremos dos posibles situaciones.

Situación 1: Los clientes no se encuentran categorizados en mayoristas ni minoristas por lo que se maneja un único precio para cada producto. De encontrarse con esta situación y en un primer análisis el diseñador preguntará si se desea guardar los distintos precios a los que se vendió cada producto en el tiempo o si sólo se desea conservar el último precio, o sea el vigente actualmente. Esta pregunta y su correspondiente respuesta no corresponde al proceso de normalización, ya que la misma debió ser efectuada en la etapa de diseño conceptual y su respuesta reflejada en DER (que como ya sabemos es el resultado obtenido de la etapa de diseño conceptual) por lo que en esta etapa partimos de que ya conocemos la realidad a modelar.

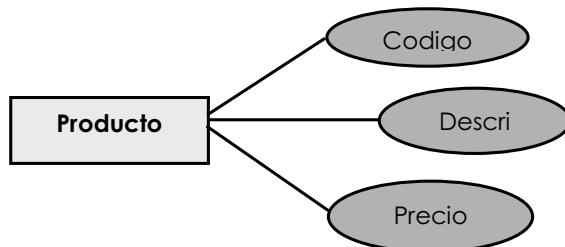
Prod:

<u>CProd</u>	<u>Descri</u>	<u>Precio</u>
35	Peras	3
37	Uvas	3
39	Kivi	11
40	Sandía	3
44	Tomates	5

**Cuadro 16**

Situación 1A: De la realidad observada se detectó que sólo interesa guardar el precio actual de cada producto, es decir el último precio.

Representación en el DER:

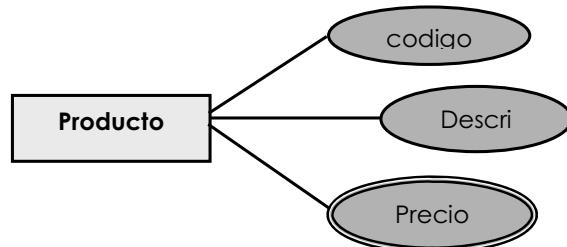


**Fig. 5. Prod y sus atributos**

En este contexto el precio del producto es un campo más de la tabla de productos, para poder ilustrar como sería esto utilicemos la tabla Prod correspondiente al ejemplo de la factura.

Situación 1B: De la realidad observada se detectó que interesa guardar los distintos precios a los que se comercializó cada producto a través del tiempo.

Representación en el DER:



**Fig. 6. Prod y sus atributos**

En este contexto se presenta la necesidad de conservar un histórico de precios para cada producto, por lo que debemos generar una nueva tabla que nos permita almacenar la variación del precio de cada producto a través del tiempo; siguiendo con el ejemplo de la factura ahora la estructura de datos ahora quedaría de la siguiente manera:

<u>CProd</u>	<u>Descri</u>	<u>CProd</u>	<u>Fecha Desde</u>	<u>Precio</u>
35	Peras	35	12/01/2006	3
37	Uvas	37	12/01/2006	5
39	Kivi	37	13/01/2006	3
40	Sandía	39	12/01/2006	10
44	Tomates	39	13/01/2006	11
		40	12/01/2006	3
		44	14/01/2006	5

Cuadro 17

Situación 2: Los clientes se encuentran categorizados en mayoristas y minoristas por lo que se maneja un precio para cada producto y tipo de cliente. La solución a esta situación no será planteada, sino que le dejaremos a usted la tarea de modelar como quedaría la estructura de datos para poder reflejar esta realidad.

### Diagrama Lógico

Toda estructura de datos normalizada, conjuntamente con las dependencias de clave primaria (recuerde que esta asegura la unicidad de cada registro dentro de una tabla) y clave externa (siendo esta la que permite asegurar que el valor de un campo o conjunto de campos en una tabla existan previamente en otra), puede ser representada mediante un diagrama lógico.

En este diagrama las tablas son representadas como un cuadro y su correspondiente nombre sobre el mismo, adicionalmente se puede incluir para cada tabla el conjunto de campos que forman parte de la misma y subrayarse su clave primaria. Las dependencias de clave externa (también denominada foránea) son representadas mediante flechas que van desde la tabla referente hacia la tabla referenciada, adicionalmente la flecha puede partir desde el campo o conjunto de campos que forma/n la clave externa de la tabla hacia la clave primaria de la tabla referenciada, indicándose de esta manera el campo o conjunto de campos que forman la clave externa. Recuerde que la clave externa de una tabla siempre referencia en forma completa a la clave primaria de la tabla referenciada.

Es importante no confundir el diagrama lógico con el DER, ya que NO son lo mismo, dado que en el diagrama lógico se incluyen los campos que forman la clave externa de una tabla mientras que en el DER no.

Siguiendo con el ejemplo de la factura, a continuación se presenta como quedaría representado el correspondiente diagrama lógico para la estructura de datos normalizada:

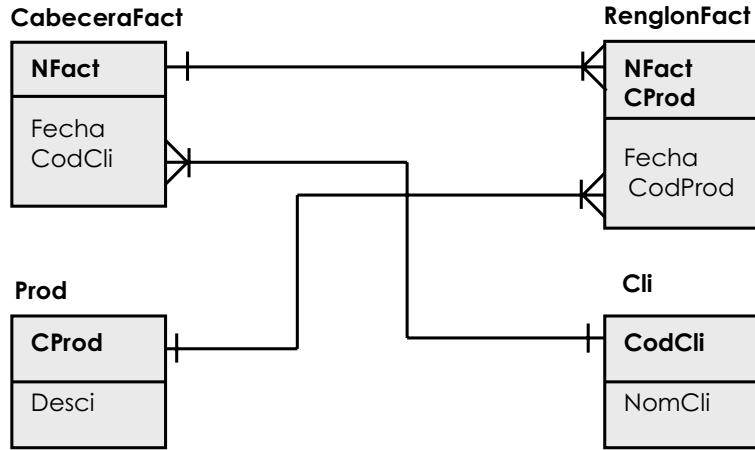


Fig. 7. Diagrama lógico de Factura

Recuerde que la inclusión de los campos que forman parte de cada tabla como así también la indicación de su correspondiente clave primaria es opcional y su inserción dependerá del grado de detalle que se desee mostrar en el diagrama lógico.

Con lo visto estamos en condiciones de resolver el punto 2 del caso Aprendiendo:

CATEGORIA (#ID\_CATEGORIA, DESCRIPCION)

CURSO (#ID\_CURSO, NOMBRE, DESCRIPCION, DURACION,  
ID\_CATEGORIA, ID\_CURSO\_REQUSITO)

SEDE (#ID\_SEDE, NOMBRE)

CURSO\_SEDE (#ID\_CURSO, #ID\_SEDE)

DOCENTE (#ID\_DOCENTE, NOMBRE)

CURSO\_DOCENTE (#ID\_CURSO, #ID\_DOCENTE)

TEMA (#ID\_TEMA, DESCRIPCION)

TEMA\_CURSO (#ID\_CURSO, #ID\_TEMA, #FECHA)

PRECIO\_CURSO (#ID\_CURSO, #FECHA, PRECIO)

PAIS (#ID\_PAIS, NOMBRE)

PROVINCIA (#ID\_PAIS, #ID\_PCIA, NOMBRE)

LOCALIDAD (#ID\_PAIS, #ID\_PCIA, #ID\_LOCAL, NOMBRE)

EMPRESA (#ID\_EMPRESA, NOMBRE, ID\_PAIS, ID\_PCIA, ID\_LOCAL,  
ID\_ACTIVIDAD)

ASISTENTE (#ID\_ASISTENTE, NOMBRE, APELLIDO, NRO\_DOC,  
ID\_NACIONALIDAD, FECHA\_NACIMIENTO, ID\_SEXO, ID\_PAIS, ID\_PCIA, ID\_LOCAL,  
CALLE, NUMERO, PISO, DEPTO, ID\_EMPRESA)

ASISTENTE\_TELEFONO (#ID\_ASISTENTE, #TELEFONO)

ASISTENTE\_EDICION\_CURSO (#ID\_CURSO, #ID\_SEDE,  
#FECHA\_INICIO, #TURNO,  
#ID\_ASISTENTE, NOTA\_FINAL)

EDICION\_CURSO (#ID\_CURSO, #ID\_SEDE, #FECHA\_INICIO,  
#ID\_TURNO, ID\_DOCENTE)

ACTIVIDAD (#ID\_ACTIVIDAD, DESCRIPCION)

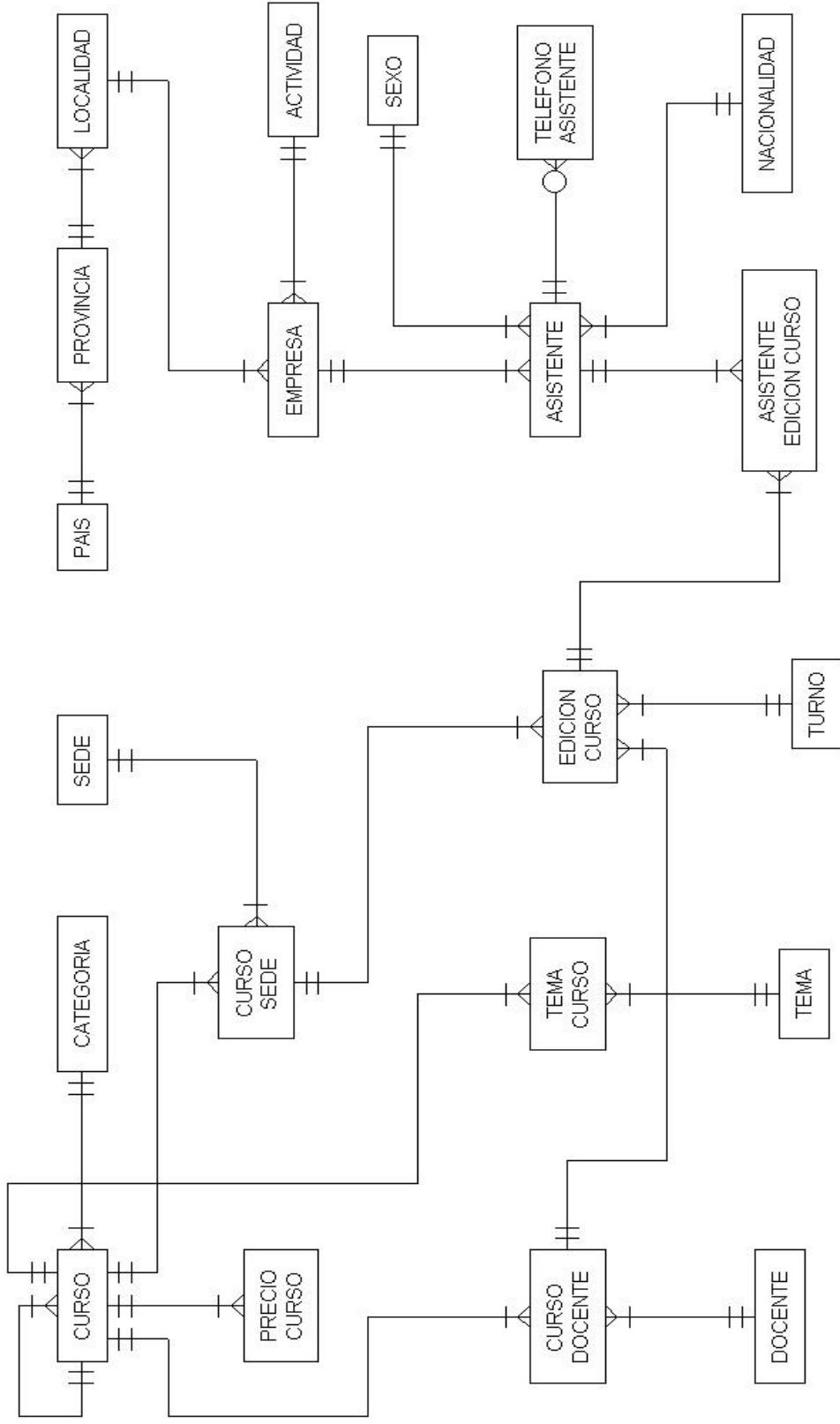
TURNO (#ID\_TURNO, DESCRIPCION)

NACIONALIDAD (#ID\_NACIONALIDAD, DESCRIPCION)

SEXO (#ID\_SEXO, DESCRIPCION)



Los campos con # y **negrita** representan la clave primaria de la tabla.





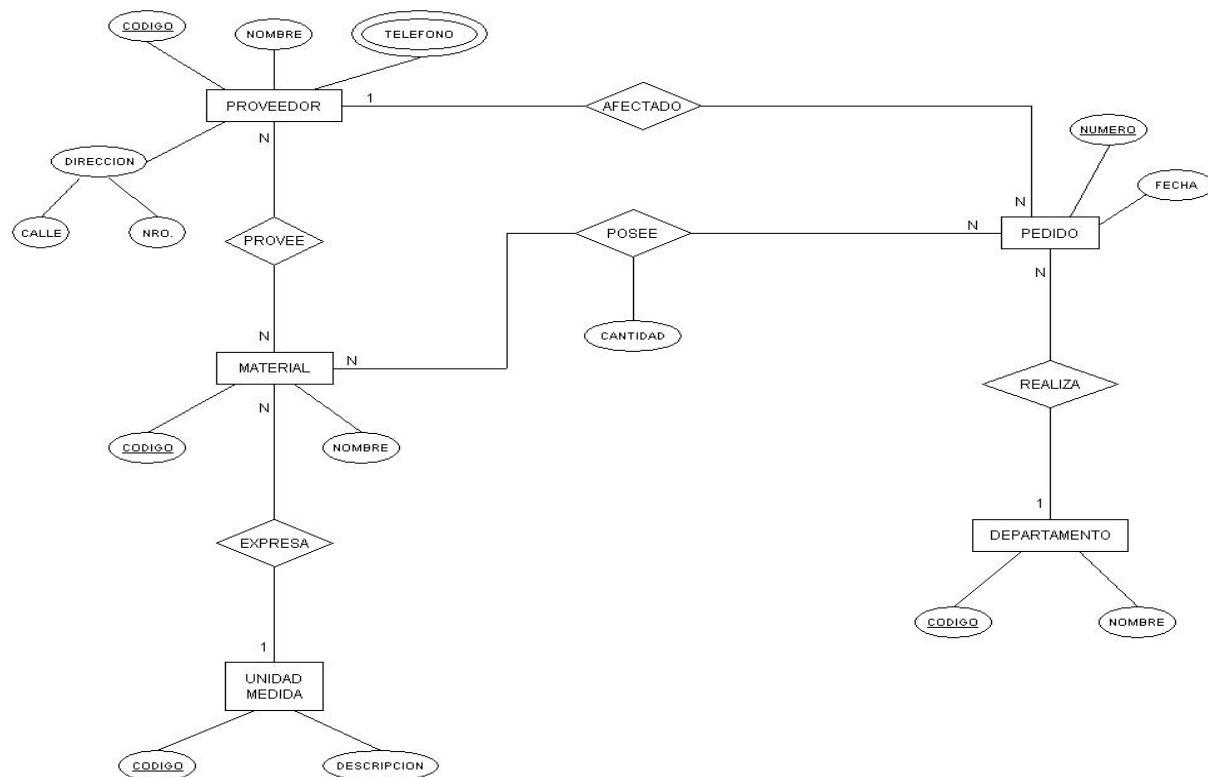
|

# **Casos Prácticos Diseño Lógico**



### I.1. Caso COMERCOM

Convertir el Modelo Conceptual presentado a continuación en un Modelo Relacional expresado en 3FN:



#### Importante

Algunas de las soluciones a los ejercicios planteados en éste capítulo se las pueden encontrar en el capítulo **J. Resoluciones Diseño Lógico**.



## I.2. Caso MAGIC FUN

La misión de MAGIC FUN es que la totalidad de sus clientes se entretengan jugando como niños. Para lograrlo comercializa software para Play-Stations y PCs.

Esperanza Pagadizabal, encargada de la gestión comercial de MAGIC FUN, nos explica la importancia lúdica en adultos y niños, "...el juego desencadena una importante fuerza vital que puede provocar cambios a favor en la conducta de adultos y niños...". MAGIC FUN cree que debe conducir estos cambios.

La empresa en su local comercial tiene una gran diversidad de juegos. Los mismos vienen presentados en distinto tipo de soportes, CDs, Diskettes, Cartuchos, Cintas, etc. Un mismo juego puede venir en más de un soporte.

Cuando Esperanza decide comprar juegos, los busca en primer lugar en el mercado local. Si éste no satisface sus demandas, entonces los importa. La empresa tiene varios proveedores de los cuales obtiene los juegos que posteriormente comercializa.

Un proveedor puede proveer todos los juegos o solamente algunos. No necesariamente comercializa todos los soportes en los que se presenta un juego. Un mismo proveedor puede vendernos el juego para todas las plataformas en las que viene el juego o solamente alguna de ellas.

Los proveedores son identificados con un código y los datos que nos interesa almacenar de ellos son: Nombre o Razón Social, País, Provincia, Localidad y teléfonos.

Entre las plataformas más comunes encontradas en el mercado están:

- ✓ PCs, plataforma PC,
- ✓ XBOX, Microsoft Xbox,
- ✓ PSX, Sony Play Station 1,
- ✓ DC, Sega Dreamcast,
- ✓ GC, Nintendo Gamecube,
- ✓ PS2, Sony Playstation 2.

Los juegos se encuentran clasificados en categorías. Un juego posee una única categoría.

Las categorías más significativas son:

- ✓ Acción.
- ✓ Estrategia.
- ✓ Aventuras Gráficas.
- ✓ Carreras.
- ✓ Combate.
- ✓ Juegos de Mesa.

Los juegos tienen un plazo de seis meses de garantía, nos comenta Esperanza. Sólo pueden ser cambiados ante causas justificables. Esto es posible únicamente si el cliente conservó la factura de compra.

### Consideraciones adicionales

- ✓ En una misma factura un cliente puede comprar más de un juego.
- ✓ Cuando se realiza una venta se registra la factura, la fecha, el cliente, el título del juego, el soporte, la plataforma, el proveedor, la cantidad comprada y el precio del mismo.
- ✓ La empresa maneja una lista de precios, dependiendo éste último del Juego, del Proveedor, del Soporte y de la Plataforma. Es importante poder conocer como varió el precio con el transcurso del tiempo.
- ✓ Un mismo título puede comercializarse para distintas plataformas.
- ✓ Los juegos se convierten en clásicos sólo si superan las 1500 unidades vendidas.
- ✓ La forma de identificar si un juego fue comprado en el exterior o en el mercado local es a partir del país de origen del proveedor del juego.
- ✓ Con el objetivo de minimizar la posibilidad de error por parte de quien emite la factura de venta al momento de seleccionar la combinación juego-plataforma-soporte-proveedor es que se decidió tener almacenadas las posibles combinaciones que pudieran existir restringiendo de esta forma la tarea del operador de facturación a la selección de una opción válida.
- ✓ A su vez para restringir las combinaciones posibles a ser definidas para juego-plataforma-soporte-proveedor es que se ha decidido almacenar también las siguientes combinaciones:
  - Las distintas plataformas en las que viene un juego.
  - Los distintos soportes en los que se comercializa cada juego.
  - Cada uno de los juegos que un proveedor comercializa.



#### Se pide

1. Realizar un modelo de datos en tercera forma normal que cubra los requisitos planteados en el enunciado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándonos en un capítulo aparte llamado "Supuestos"



### I.3. Caso SOLAR COLLEGE

SOLAR COLLEGE es una universidad reconocida en el ámbito educativo a nivel nacional. Es una institución que por su tamaño no ha tenido hasta el momento inconvenientes con la registración de sus operaciones con planillas de cálculo.

Actualmente las limitaciones presentadas por el uso de planillas y documentos se han reflejado en la incapacidad de recuperar archivos infectados, y otras veces directamente se han perdido documentos por haber sido borrados, al desconocerse su utilidad.

Es por eso que autoridades de SOLAR COLLEGE han decidido invertir en tecnología más confiable y segura, y están pensando en una BD centralizada y varias estaciones clientes.

Sebastián Boeri, Director de SOLAR COLLEGE está al tanto de las Bases de Datos que se utilizan y se ha inclinado a emplear una relacional, por su amplia utilización en el mercado.

Entre los datos que se desea guardar se encuentran:

Los datos personales de los alumnos, estos son: Apellido, Nombre, Fecha de Nacimiento, E-mail, y Teléfonos.

Además los alumnos tienen un Legajo que los diferencia unos de otros.

Cada alumno cursa una o más materias a las cuales se debe inscribir cada cuatrimestre.

La institución tiene distintos cursos para las distintas materias que se dictan, en distintos días y horarios. Los números de cursos pueden repetirse, pero son únicos para cada materia. Todo curso posee un único docente a cargo.

Los datos que se guardan de los docentes son el Nombre, Apellido, Dirección, Fecha de Nacimiento, Teléfonos, Cargo, y Fecha de Ingreso a la institución.

Todo docente posee un jefe, por lo que es importante conocer quién es el jefe de cada docente. También se sabe que el jefe es otro docente.

#### **Consideraciones adicionales**

- ✓ Un curso sólo se puede tener asignado un único docente.
- ✓ Se desea guardar los cursos en los que se inscribió cada alumno en cada cuatrimestre y la nota obtenida para poder calcular un promedio de notas del alumno para un determinado cuatrimestre.

- ✓ Para evitar posibles errores al momento de asignar un día y un horario a cada curso es que resulta importante tener guardado por un lado los días de la semana en los que se pueden dictar clases, por otro lado los horarios en los que se pueden dictar cursos y por último las combinaciones de día-horario que se pueden asociar a los distintos cursos.



**Se pide**

1. Realizar un modelo de datos en tercera forma normal que cubra los requisitos planteados en el enunciado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos"



#### I.4. Caso TODO LIBRO



##### Se pide

1. Se pide realizar el pasaje del modelo conceptual desarrollado en el capítulo anterior a un modelo de datos expresado en tercera forma normal y adicionalmente realizar el diagrama lógico correspondiente al modelo desarrollado.



### 1.5. Caso SICON

"...Pensamos en desarrollar un sistema robusto, que nos permita gestionar las futuras empresas que se desprendan de esta..." nos dice Germán Pello, Gerente Financiero de SICON.

La empresa SICON necesita un sistema contable para mejorar su gestión interna.

Este sistema debe permitir:

- ✓ Manejar más de una empresa, es decir debe ser multiempresa.
- ✓ Llevar un plan de cuentas.
- ✓ Emitir balances de sumas y saldos.
- ✓ Llevar un control del estado financiero de la empresa.
- ✓ Determinar los períodos contables de la empresa.

Además el sistema nuevo nos debe solucionar los siguientes requerimientos:

- ✓ El acceso a varios períodos contables en distintos momentos.
- ✓ El manejo de un plan de cuentas específico para cada una de las empresas. Dado que quizás muchas cuentas sean comunes a varias empresas es que se debe contemplar la posibilidad de que existan números de cuenta (por ej. Caja \$) que sean utilizados por mas de una empresa, por lo que se debe tener un plan de cuentas general y luego de alguna manera determinar con que cuentas operará cada una de las empresas administradas bajo este sistema.
- ✓ La carga de asientos por pantalla debe realizarse de la manera más automática posible y poder imprimirlos.

Al dar de alta una nueva empresa en el sistema se procederá de la siguiente manera: se ingresará el nombre de la empresa en cuestión y el rubro al que se dedica.

Una vez dada de alta la empresa, se deben dar de alta los períodos contables. Esto servirá para establecer los ejercicios económicos o períodos contables de la empresa a analizar. Tener presente que no necesariamente para todas las empresas el ejercicio económico tiene la misma fecha de inicio y fin.

Al dar de alta un asiento contable para cada operación en particular se cargan los siguientes datos: fecha carga, usuario de carga, fecha de registro contable, número de asiento, cuenta contable y el monto.

Para minimizar los errores durante la carga de datos es que se decidió definir tipos de asiento y a estos asociarles las cuentas contables que podrán ser usadas, evitándose de esta manera por ejemplo que en un asiento cuyo tipo es Amortización se realice una imputación sobre la cuenta Caja en Pesos.

Como norma de seguridad y para garantizar la división de tareas es que no todos los usuarios podrán tener acceso a todos los tipos de asiento, sino que se deben poder parametrizar los tipos de asiento que pueden llegar a utilizar cada uno de los usuarios del sistema. En la reunión mantenida con Germán Pello en la cual se discutió la forma de implementar esta restricción se decidió que la forma de hacerlo será restringiendo los tipos de asiento según el perfil, es decir que según el perfil que posea el usuario serán entonces los tipos de asientos que puede utilizar.

### **Consideraciones adicionales**

- ✓ Cada cuenta contable tendrá asociado un identificador que indica si se debita o se acredita.
- ✓ Los tipos de asiento son comunes para todas las empresas.
- ✓ Las cuentas contables que forman parte de cada tipo de asiento no necesariamente son las mismas para todas las empresas.
- ✓ Todo asiento está conformado como mínimo por dos cuentas, una con signo deudor y otra con signo acreedor.
- ✓ No todos los usuarios pueden operar con todas las empresas por lo que debe poder definirse qué usuarios pueden trabajar con cada una de las empresas.
- ✓ Todo usuario tiene asociado un perfil, a saber:
  - Operador
  - Gran Operador
  - Supervisor
  - Administrador
- ✓ SICON ya cuenta con un modelo desarrollado para el sistema en cuestión, pero Germán Pello no confía mucho en él. A continuación se presenta dicho modelo:

EMPRESA (#ID\_EMPRESA, NOMBRE, NOMBRE\_RUBRO)

EJERCICIO\_CONTABLE (#ID\_EMPRESA, #FECHA\_INICIO,  
#FECHA\_FIN)

PLAN CUENTA (#ID CUENTA, ID\_EMPRESA)

PLAN CUENTA\_EMPRESA (#ID\_EMPRESA, #ID CUENTA)

TIPO\_ASIENTO (#ID TIPO\_ASIENTO, NOMBRE)

PERFIL (#ID\_PERFIL, DESCRIPCION)

USUARIO (#ID\_USUARIO, NOMBRE)

USUARIO\_TIPO\_ASIENTO (#ID\_USUARIO, #ID\_TIPO\_ASIENTO)

ASIENTO (#ID\_EMPRESA, #ID\_TIPO\_ASIENTO, #NRO\_ASIENTO,  
#ID CUENTA, FECHA\_CARGA,  
FECHA\_REGISTRO\_CONTABLE, IMPORTE)



**Aclaración**

Los campos con # y negrita son la clave primaria de la tabla.



**Se pide**

1. Corregir y finalizar el proceso de normalización. Identificar los posibles errores en el proceso de normalización, las entidades y atributos faltantes o inadecuadamente diseñados. Adicionalmente realizar el diagrama lógico correspondiente al modelo normalizado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos".



### I.6. Caso SOLARIUM TOSTADITOS

SOLARIUM TOSTADITOS tiene como actividad principal la de lograr broncear de forma única la piel de sus clientes.

Cuando una persona visita TOSTADITOS, lo primero que hace es ser atendido en la recepción, donde se le da un turno para una cama solar. En ese momento lo que hace es comprar fichas por el tiempo que desee.

Las fichas tienen una duración de 10, 12, 15, 20 y 25 minutos. No todas las fichas sirven para todas las máquinas, sino que depende del tipo de máquina.

TOSTADITOS dispone de diversas máquinas, que permiten un mejor bronceado y hacen menor daño a la piel.

Entre los equipos que posee están:

- ✓ StarPower 52 G.
- ✓ Solaris Plus 42 3F.
- ✓ Sunstar ZX32 3F Series.
- ✓ Sunstar ZX30 3F Series.
- ✓ Pro28LX Series.
- ✓ Pro 24S Series.

La cama Mega bronceadora StarPower, una de las más solicitadas, utiliza 48 bombillos reflectores de 160 watts, mientras que la Sunstar ZX30 3F Series utiliza 40 bombillos reflectores de 160 watts y la Pro 24S Series utiliza 48 bombillos reflectores de 120 watts.

Cada cama tiene una cantidad determinada de bombillos reflectores. Cada bombillo reflector tiene una determinada cantidad de watts.

"La gente viene a buscar color....", nos cuenta Raquel Luraschi, Gerente Administrativa de TOSTADITOS, "y eso es lo que nosotros le damos...".

TOSTADITOS además ofrece la posibilidad de comprar abonos que van desde los \$20. La ventaja de dichos abonos es que el valor de la ficha es un 10% menos.

Cada vez que se utiliza el abono se entregan fichas por el valor que se desea consumir, registrándose la fecha, el importe consumido y el saldo.

Las máquinas no son propias sino que se alquilan a distintos proveedores de camas solares, pagándose en concepto de alquiler un monto fijo mensual pudiendo cambiar este a lo largo del tiempo, por lo que se debe poder conservar cuál fue el valor del alquiler a través del paso del tiempo.

De cada una de las máquinas es importante conocer la cantidad de veces que la misma ha sido utilizada en un determinado período de tiempo, como así también lo recaudado por esta en un determinado mes pudiéndose entonces determinar de esta forma la utilidad devengada por cada una de las máquinas.

### Consideraciones adicionales

- ✓ Los abonos tienen una fecha de vencimiento.
- ✓ De un proveedor es importante conocer su nombre y el alquiler cobrado por máquina.
- ✓ Se considera que la cantidad de veces que se uso una máquina es equivalente a la cantidad de fichas extraídas de la misma al final del día.



#### Se pide

1. Corregir y finalizar el proceso de normalización. Identificar los posibles errores en el proceso de normalización, las entidades y atributos faltantes o inadecuadamente diseñados.
2. Corregir el Diagrama lógico correspondiente al modelo.

A continuación se adjunta la documentación entregada por María Laura Luna, quién actualmente se encuentra de licencia:

FICHA (#ID\_FICHA, DESCRIPCION, DURACION)

PROVEEDOR (#ID\_PROVEEDOR, NOMBRE)

MAQUINA (#ID\_MAQUINA, NOMBRE, ID\_TIPO\_MAQUINA, ID\_FICHA)

MAQUINA\_PROVEEDOR (#ID\_MAQUINA, #ID\_PROVEEDOR)

PRECIO\_ALQUILER\_MAQUINA (#ID\_MAQUINA, #AÑO, #MES, PRECIO)

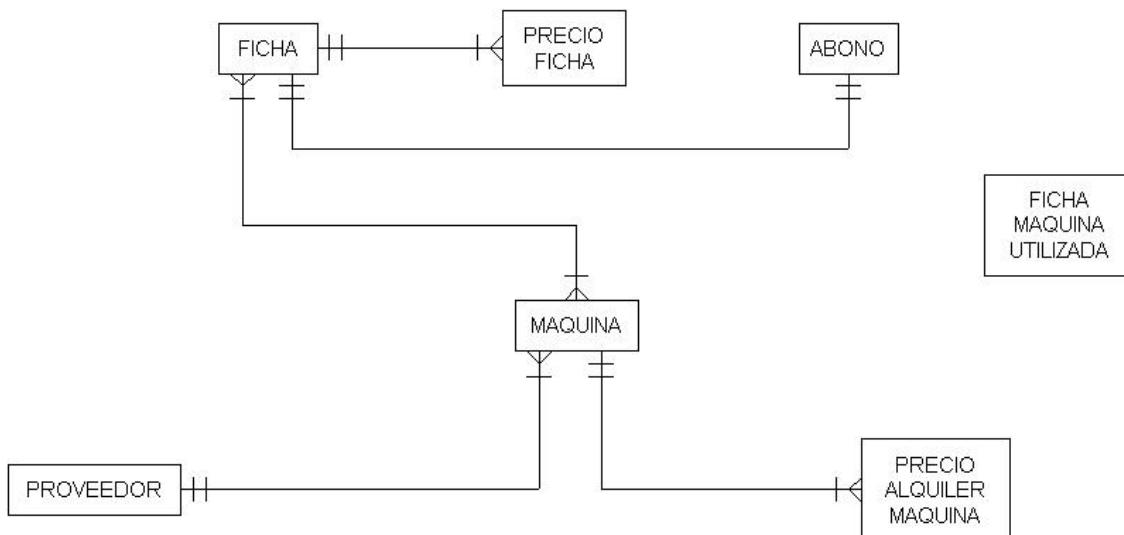
ABONO (#ID\_ABONO, FECHA\_VENCIMIENTO, IMPORTE)

CONSUMO\_ABONO (#ID\_ABONO, #ID\_FICHA, #FECHA,  
CANTIDAD\_FICHAS)

FICHA\_UTILIZADA\_MAQUINA (#ID\_MAQUINA, #ID\_FICHA, FECHA, CANTIDAD)

**Aclaración**

Los campos con # y negrita son la clave primaria de la tabla.

**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos".



### 1.7. Caso SURTIDITO S.A.

La empresa SURTIDITO S.A. es un supermercado mayorista dedicado a vender distintos productos en la Capital Federal.

Dada la creciente competencia en este rubro la gerencia de SURTIDITO S.A. ha decidido implementar un sistema que le permita controlar la satisfacción de sus clientes.

En una entrevista mantenida con el jefe de Atención a Clientes, Saúl Jauregui, le ha manifestado su interés por implementar un sistema que cuente con una base de datos que permita hacer el seguimiento de los clientes y sus quejas.

Saúl, pretende con este sistema lograr mejorar la satisfacción del cliente en cuanto a productos y atención dispensada.

En reuniones posteriores celebradas en la empresa con Saúl y su gente, Ud. ha detectado lo siguiente:

Un cliente que realiza una compra en alguna de las sucursales de SURTIDITO S.A. deja sus datos por primera vez llenando un cupón, en el cual los datos relevantes identificados son:

- ✓ Nombre.
- ✓ Apellido.
- ✓ Documento.
- ✓ Teléfono/s.
- ✓ Dirección.

Para que un cliente de SURTIDITO pueda realizar una llamada telefónica para quejarse debe previamente haber completado el formulario con todos sus datos personales, y además haber realizado una compra y conservar el ticket que avale la misma.

Cualquier cliente puede realizar posteriormente a una compra un reclamo telefónico. Para esto la empresa cuenta con una línea 0-800 que presta servicio de atención al cliente.

Los inconvenientes que originan un llamado pueden ser diversos: mala atención, suciedad del local, productos dañados, productos que se hallan en mal estado, escasez de marcas, etc.

Al momento de llamar, un operador ingresa los siguientes datos: Cliente, Fecha y Hora del incidente, la Fecha Actual, Sucursal donde efectuó la compra, Motivo de la queja, número de ticket, producto (de estar relacionado el reclamo con algún/os producto/s) y comentarios adicionales que el cliente efectúe.

Si el llamado es a consecuencia de problemas con alguno/s de los productos comercializados por Surtidito entonces debe guardarse un detalle de los productos a los que hace referencia el cliente. La empresa atiende reclamos las veinticuatro horas del día los 365 días del año.

Asentada la queja, la misma tiene un curso de acción inicial hasta ser resuelta.

Un reclamo puede tener dos acciones principales opuestas: No Corresponde Solución o Resuelta. Este es el resultado final de una acción, pudiéndose dar estados intermedios entre cada uno de estos dos estados: En curso, Pendiente de Aprobación Dpto. CC, Aprobado Dpto. CC, Pendiente Aprobación MKT, Aprobado MKT, pero finalmente, todas las quejas llegan al estado Resuelta o No Corresponde Solución.

El sistema debe permitir llevar un control detallado de todos los estados por los que pasó un reclamo, debiéndose guardar las fechas y los estados a través de los cuales pasó.

Un llamado concluye con una única acción determinada.

Hay determinados niveles de prioridades ante los reclamos:

- ✓ Muy Urgente.
- ✓ Urgente.
- ✓ Normal.
- ✓ Medio.
- ✓ Bajo.

### **Consideraciones adicionales**

- ✓ Las sucursales tienen un responsable a cargo.
- ✓ Todo responsable de sucursal responde a un gerente. Es importante conocer de quien depende cada responsable de sucursal, como así también a quien reporta cada uno de los gerentes.
- ✓ Cada producto comercializado es provisto por un único proveedor, es decir un mismo producto nunca tiene dos proveedores distintos.
- ✓ Dos o más sucursales no pueden tener como responsable a la misma persona, o sea un responsable solo puede tener a su cargo una única sucursal en un mismo momento.



**Se pide**

1. Realizar el Diagrama de Entidad Relación correspondiente..
2. Realizar el pasaje del DER desarrollado en el punto anterior a un modelo de datos expresado en tercera forma normal y adicionalmente realizar el diagrama lógico correspondiente al modelo desarrollado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos".



### I.8. Caso CHICO PETS



#### Se pide

1. Realizar el pasaje del modelo conceptual desarrollado en el capítulo anterior a un modelo de datos expresado en tercera forma normal y adicionalmente realizar el diagrama lógico correspondiente al modelo desarrollado.



### I.9. Caso RENT A KAB

RENT A KAB es una empresa dedicada al transporte de pasajeros, la cual realiza viajes a pedido de sus clientes a diversos puntos de Capital Federal y provincia de Buenos Aires.

La empresa recibe llamados de sus clientes y los registra enviando inmediatamente un coche a su domicilio.

El operador de la base es quien recibe los llamados de los clientes.

La primera vez que llama un cliente se guardan los siguientes datos: Nombre del Cliente, Dirección, Teléfono y Fecha de Nacimiento.

Por cada llamado el operador pregunta la cantidad de pasajeros a transportar, Destino y Forma de Pago.

Las formas de pago pueden ser:

- ✓ Efectivo
- ✓ Cuenta Corriente

Los choferes que se encuentran en la calle trabajando reciben un pedido de la base para proceder al transporte del pasajero, siendo el operador quien determina que chofer se hace cargo del cliente dando los datos precisos del lugar de destino.

Anteriormente, nos comenta Marco Boces (encargado de los Choferes de RENT A KAB), que cualquier chofer tomaba el pedido, con tal de asegurarse el pasajero, y se producían retrasos que derivaban en quejas por parte del cliente respecto de la impuntualidad del servicio.

Por este motivo Máximo decidió implementar un sistema de premios y castigos que sanciona con multas a los choferes que incurren en retrasos.

Marco sostiene que, con el nuevo sistema, las llamadas por quejas por impuntualidad de los clientes se verán disminuidas en un 84%. De esta forma se cuida a los choferes y la clientela.

Cada vez que un cliente llama por una queja de impuntualidad se asocia a esta con el viaje por el cuál se está llamando, registrándose la fecha y hora y el tiempo de la llamada, el cliente que llama, el destino del viaje, el chofer que fue asignado a dicho viaje y la demora acusada por el cliente.

Los datos que se guardan de los choferes son:

- ✓ Nombre.
- ✓ Apellido.
- ✓ Dirección de Residencia.
- ✓ Teléfonos.
- ✓ DNI.

- ✓ Nro. de Registro.

Las severidades de las multas aplicadas a los choferes por retraso son directamente proporcionales a los tiempos de retraso, medidos en minutos:

- ✓ 0-5 .....Leve.
- ✓ 5-10 .....Grave.
- ✓ +10 .....Muy Grave.

Según el tipo de severidad se establecen las multas que van desde un día hasta una semana de suspensión del servicio radial. Por ejemplo un tipo de multa leve equivale a un día de suspensión.

Cuando los choferes son suspendidos se debe asentar el período de tiempo de suspensión del servicio, es decir a partir de que día y hasta que día se encuentra suspendido para la utilización del servicio.

#### **Consideraciones adicionales**

- ✓ Por cada llamado de solicitud de un vehículo el operador informa al cliente la demora aproximada, siendo importante poder almacenar este dato.
- ✓ Los clientes pueden solicitar más de un viaje por día.
- ✓ Los coches que operan con RENT A KAB tienen que tener una antigüedad máxima de 5 años.
- ✓ Cuando un cliente se registra por primera vez se le da un número de socio.
- ✓ De los autos se guarda la marca, modelo, color, año de fabricación, titular y chofer asignado.
- ✓ Los autos se asignan a dos choferes que realizan un turno de 12 horas cada uno, a saber: turno día y turno noche. Dado que un chofer puede renunciar, cambiar a otro vehículo o rotar de turno es que resulta importante poder conservar un histórico de que chofer tuvo a su cargo un auto y en qué turno durante un determinado período.



#### **Se pide**

1. Realizar el Diagrama de Entidad Relación correspondiente.,
2. Realizar el pasaje del DER desarrollado en el punto anterior a un modelo de datos expresado en tercera forma normal y adicionalmente realizar el diagrama lógico correspondiente al modelo desarrollado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos".

**I.10. Caso ABC MOVIL**

"Comunicarse es una necesidad" sostiene, Damián Findling, gerente de ABC MOVIL, quien cree que en los próximos diez años todos estarán haciendo uso de algún tipo de tecnología móvil.

La empresa que lidera Findling se encuentra en un turbulento mercado. Damián sabe que la única forma posible de incrementar la participación en el mercado es contar con una estructura importante de ventas. Para poder realizar un seguimiento de las ventas como así también del rendimiento de los vendedores es que Findling ha decidido invertir en el desarrollo de un nuevo módulo de ventas, por lo que usted ha sido contratado para diseñar el modelo de datos para dicho sistema.

Del relevamiento realizado se obtuvieron los siguientes puntos a ser considerados en el modelo:

Cada vendedor recibe una comisión en base al desempeño alcanzado. La estructura con que cuenta la empresa es: Vendedor – Coordinador – Gerente. Los coordinadores tienen a su cargo un número reducido de vendedores. Cada gerente puede tener uno o más coordinadores a su cargo.

Los equipos que comercializa la empresa son provistos por diferentes fabricantes, tales como Motorola, Ericcson, Samsung, etc.

Cada una de estas marcas tiene distintos modelos, así por ejemplo, la Marca Motorola tiene entre sus modelos al Motorola V60, Motorola V120, y el Audiovox 8000 y Audiovox 9000, entre otros.

Entre los datos de los equipos que se desean guardar se encuentran los de: Nro. de Serie, Comprador y Fecha de compra.

Los equipos son vendidos a través de una Operación de venta y es registrada mediante la emisión de una factura. Cada factura tiene un número único que no se repite, una forma de pago, pudiendo ser el equipo abonado en una o más cuotas y en un tipo de moneda determinada. La empresa trabaja con Pesos y Dólares. En el caso de ésta última moneda, se toma como valor del dólar el valor de cierre de la fecha más cercana a la operación, los tipos de cambio deben poder también ser registrados en el sistema.

Dado que la financiación de los equipos para el caso de venta en cuotas es realizada directamente por ABC MOVIL es que resulta importante poder conocer la cantidad de cuotas en las que se efectuó la venta, el importe de cada una y la fecha de vencimiento.

Cada vendedor es dado de alta por la empresa guardándose su Número de Legajo, su Nombre, Nro de Documento y dirección.

Cuando un vendedor es desvinculado de la empresa se guarda la fecha de baja del mismo. Cada vendedor percibe un sueldo básico mensual y una comisión sobre la venta de cada equipo, siendo esta no necesariamente la misma para todos los equipos, es decir por ejemplo sobre la

venta de un equipo Motorola V300 la comisión es del 1% mientras que sobre un Nokia 6125 es del 1,5%. Dado que esta comisión puede variar en el tiempo es que resulta importante tener un detalle de cómo fue cambiando la misma.

Los equipos pueden ser vendidos en distintos puntos de venta que posee la empresa. Estos puntos de venta se denominan stands y están ubicados en distintos lugares del país. De cada stand es importante conocer la provincia, localidad, calle y número donde se encuentra ubicado, además de sus respectivos teléfonos.

Los vendedores son asignados a un determinado punto de venta, pudiendo ser rotados de manera frecuente, para liquidar las comisiones por ventas en forma correcta es necesario llevar un histórico de donde estuvo cada vendedor y el período en el que estuvo.

En cada punto de venta puede haber uno o muchos vendedores y un único coordinador. Un vendedor solo trabaja en un único punto de venta a la vez.

Todos los equipos vendidos son acompañados por un plan de llamadas acorde a las necesidades del cliente, para esto la compañía posee una serie de planes posibles, dándole al cliente al momento de la compra la posibilidad de seleccionar cuál es el que mejor se adapta a sus necesidades de comunicación.

Cada plan incluye una cantidad determinada de minutos libres para hablar y una cantidad de mensajes de texto libres al mes. A cambio de esta prestación la empresa ABC MOVIL cobra un abono mensual fijo que varía según el plan.

Como el cliente puede en cualquier momento decidir cambiar de plan asociado a su teléfono es que resulta necesario guardar el plan que posee actualmente como así también los que tuvo en el pasado.

Dentro de las 48 hs. posteriores a la compra el cliente es contactado para darle la bienvenida y realizarle una encuesta para evaluar su nivel de satisfacción respecto de la atención recibida por parte del vendedor.

Las preguntas efectuadas por el encuestador son entre otras:

- ✓ ¿Cuál es su nivel de satisfacción respecto de la atención recibida?
- ✓ ¿Cómo calificaría la predisposición del vendedor?
- ✓ ¿En qué nivel han sido respondidas todas sus dudas?

Los datos recolectados y almacenados en una encuesta son: Fecha de la encuesta, cliente, legajo del encuestador, pregunta realizada y calificación dada por el cliente ante la pregunta.

Las preguntas a ser efectuadas por parte del encuestador se encuentran estandarizadas y resulta importante tenerlas almacenada.

La respuesta del cliente a todas las debe estar comprendida en una escalada de 0 a 10, siendo importante también tener almacenada en el sistema dicha escala para evitar de esta forma posibles errores de tipeo por parte del encuestador al momento de la carga.

**Consideraciones adicionales**

- ✓ Todos los Vendedores tienen solo un Coordinador y un Gerente.
- ✓ El número de serie de los equipos es único para cada uno de los equipos provistos por un mismo fabricante, pero puede llegar a repetirse entre equipos de distintos fabricantes.
- ✓ Existe una lista de precios única por equipo.
- ✓ Un coordinador solo puede tener a cargo un único punto de venta.

**Se pide**

3. Realizar el Diagrama de Entidad Relación correspondiente.,
4. Realizar el pasaje del DER desarrollado en el punto anterior a un modelo de datos expresado en tercera forma normal y adicionalmente realizar el diagrama lógico correspondiente al modelo desarrollado.

**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos".



### I.11. Caso TERRA CONTROL

TERRA CONTROL es una empresa que se dedica al seguimiento de vehículos terrestres a través de sistemas GPS.

"Los sistemas GPS son sistemas satelitales que permiten la localización al instante de un objeto que se mueve por la superficie de la tierra...", nos explica Antonio Bonifaccino, gerente de tecnología.

TERRA-CONTROL brinda además, otros servicios tales como el disparo automático de alarmas ante una frase en clave y servicio de custodias de vehículos.

Con el nuevo sistema se pretende ofrecer lo siguiente:

- ✓ Que los clientes puedan solicitar un presupuesto de cada servicio mediante la Web o en forma personal.
- ✓ Que los clientes puedan dejar sus datos para que la empresa los contacte a través de la Web o en forma personal.

Cada vez que un cliente entra a la Web de la empresa o se acerca personalmente debe completar un formulario con los siguientes datos:

Si es la primera vez entonces debe proporcionar la información de contacto de un cliente que consiste en sus datos personales: Número de Documento, Números de teléfono, Dirección de Mail y Dirección (provincia, localidad, calle, número, piso, departamento).

Una vez llenado estos datos el cliente puede solicitar información respecto de un servicio determinado. Para hacerlo debe ingresar la Marca del vehículo (ej. Ford, Chevrolet, etc.), el Modelo (ej. Gol, Corsa, etc.), el año de fabricación y el/los servicio/s por el/los que desea averiguar.

El presupuesto (que incluye costo del servicio y tiempo que lleva la realización del trabajo) es respondido a la dirección de mail informada por el cliente o entregado en forma personal si no posee mail.

Cada cliente cuenta con su propia cuenta, la cual puede administrar personalmente o bien, a través del personal de la empresa, para lo cual se le suministra al usuario un id (que coincide con su DNI) y una contraseña.

#### Consideraciones adicionales

- ✓ Los servicios prestados por la empresa TERRA CONTROL se encuentran codificados (por ej. Instalación equipo, Servicio de Custodia, etc.).
- ✓ Los presupuestos son identificados en forma única mediante un número de presupuesto.

- ✓ Dado que el costo de un servicio puede variar en el transcurso del tiempo es necesario poder guardar un histórico del precio del mismo.
- ✓ Si el cliente ya se encuentra registrado en el sistema entonces con ingresar su DNI podrá solicitar cualquier información que desee respecto de los servicios prestados por la empresa.
- ✓ Uno de los empleados de TERRA comenzó con el diseño del modelo pero por falta de conocimiento no ha podido finalizarlo como así tampoco está seguro de que el mismo esté en 3FN y que cumpla con los requisitos del negocio. A continuación se presenta el modelo desarrollado:

CLIENTE (#DNI, CLAVE, TELEFONOS)

CLIENTE\_DIRECCION (#DNI, ID\_PCIA, ID\_LOCAL, CALLE, NUMERO,  
PISO, DEPARTAMENTO)

CLAVE\_CLIENTE (#DNI, #CLAVE)

LOCALIDAD (#ID\_PCIA, #ID\_LOCAL, NOMBRE\_PROVINCIA,  
NOMBRE\_LOCALIDAD)

SERVICIO (#ID\_SERVICIO, #DESCRIPCION,  
CANTIDAD\_DIAS\_TRABAJO, PRECIO)

MARCA\_MODELO (#ID\_MARCA, #ID\_MODELO, NOMBRE\_MARCA)

CABECERA\_PRESUPUESTO (#ID\_PRESUPUESTO, FECHA, DNI,  
ID\_MARCA, ID\_MODELO, AÑO)

RENGLON\_PRESUPUESTO (#ID\_PRESUPUESTO, #ID\_SERVICIO,  
FECHA, DNI, CLAVE)



#### Aclaración

Los campos con # y negrita son la clave primaria de la tabla.



**Se pide**

1. Corregir y finalizar el proceso de normalización. Identificar los posibles errores en el proceso de normalización, las tablas y campos faltantes o inadecuadamente diseñados. Adicionalmente realizar el diagrama lógico correspondiente al modelo normalizado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos".

**I.12. Caso E-BOLSA**

E-Bolsa es una empresa argentina, cuya actividad principal es la compra-venta de activos financieros.

E-Bolsa opera sólo de forma electrónica, a través de su sitio Web [www.ebolsa.com.ar](http://www.ebolsa.com.ar).

Debido al aumento de su cartera de clientes es que E-BOLSA ha decidido retomar el desarrollo de un nuevo sistema que por problemas financieros había sido suspendido. Luis Álvarez, Socio Gerente, lo ha contratado a usted para que concluya el desarrollo del modelo de datos que utilizará el sistema.

En las reuniones de relevamiento por usted realizadas se obtuvo la siguiente información:

Los activos financieros incluyen acciones con cotización pública y títulos públicos. Tanto las acciones como los títulos públicos que se pueden adquirir o vender, cotizan en las respectivas bolsas internacionales.

Cada acción o título cotiza en una moneda determinada y en un único mercado.

Los ingresos de E-Bolsa derivan exclusivamente de las comisiones cobradas a sus usuarios a través de la intermediación en operaciones de compra o de venta.

Toda operación de compra o venta debe poder ser registrada para luego a fin de mes poder efectuar la liquidación por comisiones a ser cobradas a los respectivos clientes.

De cada operación es importante conocer: La fecha de realización de la misma, el cliente, si se trata de una transacción de compra o venta, el operador de bolsa que efectuó la transacción, el/los papel/es tranzado/s, la cantidad comprada/vendida, el importe total de la operación y la comisión a ser liquidada.

La empresa opera tanto con clientes residentes en el país como en el exterior, y los datos a guardar de los mismos es: Nombre o Razón Social, Mail, País, Provincia y Localidad.

"...E-Bolsa ofrece la posibilidad de poder realizar todas las operaciones durante los 365 días del año, las 24 horas, siendo el único requisito ser titular de una cuenta corriente que permita depositar o debitar los valores correspondientes a la operación...." nos explica Ronaldo Sosa, Gerente de Inversiones de E-Bolsa.

Los datos de una cuenta corriente son: Banco, Sucursal y Número de Cuenta.

Los clientes deben registrarse para poder operar, siendo este registro necesario realizarlo antes de la primera vez que se desee operar. Una vez confirmada la registración el sistema asigna un número de cliente (el cuál es necesario para poder operar con el sistema) y solicita el ingreso de una password.

### Consideraciones adicionales

- ✓ Una transacción sólo puede ser de compra o de venta, nunca por ambos motivos.
- ✓ El porcentaje de comisión a ser cobrado es un porcentaje fijo que depende del cliente.
- ✓ Cada usuario puede poseer una o más cuentas corrientes, por lo que al momento de realizar la operación debe informar la cuenta en la cual se le debitárán o acreditarán los fondos.
- ✓ Se debe mantener información de la última cotización del día para cada moneda.
- ✓ Es necesario mantener un histórico de la cotización de cada papel, como así también resulta importante poder conocer los valores de apertura, cierre, el máximo y el mínimo de cada día, de manera de poder brindar información estadística a los clientes.
- ✓ A continuación se adjunta la documentación entregada por Luis Álvarez:

TIPO\_ACTIVO (#ID\_TIPO\_ACTIVO, #MONEDA, NOMBRE)

ACTIVO (#ID\_ACTIVO, NOMBRE, ID\_TIPO\_ACTIVO, ID\_MERCADO)

TIPO\_CAMBIO\_MONEDA (#ID\_MONEDA, #FECHA, #VALOR)

COTIZACION\_ACTIVO (#ID\_ACTIVO, #FECHA, VALOR)

PROVINCIA (#ID\_PAIS, #ID\_Pcia, NOMBRE)

CLIENTE (#ID\_CLIENTE, NOMBRE, ID\_PAIS, ID\_Pcia, ID\_LOCAL,  
PORCENTAJE\_COMISION, PASSWORD)

BANCO (#ID\_BANCO, NOMBRE)

SUCURSAL (#ID\_BANCO, ID\_SUCURSAL, NOMBRE)

CUENTA\_CORRIENTE\_CLIENTE (#ID\_CLIENTE, ID\_BANCO, ID\_SUCURSAL,  
NRO CUENTA)

TIPO\_OPERACION (#ID\_TIPO\_OPERACION, DESCRIPCION)

CABECERA\_OPERACION (#ID\_OPERACION, FECHA,  
ID\_TIPO\_OPERACION, ID\_OPERADOR)

RENGLON\_OPERACION (#ID\_OPERACION, #ID\_ACTIVO,  
CANTIDAD)



**Aclaración**

Los campos con # y **negrita** son la clave primaria de la tabla.



**Se pide**

1. Corregir y finalizar el proceso de normalización. Identificar los posibles errores en el proceso de normalización, las tablas y campos faltantes o inadecuadamente diseñados. Adicionalmente realizar el diagrama lógico correspondiente al modelo normalizado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos".



### I.13. Caso BAUXITA & CO

En Argentina una de las actividades más rentables es la minera. Dado el panorama económico actual, es un momento ideal para desarrollar la actividad minera. Es por tal motivo que José Rodríguez, uno de los socios fundadores ha decidido incursionar en este rubro, y lo ha propuesto a la comisión directiva de BAUXITA & CO.

Anteriormente BAUXITA & CO. se venía dedicando exclusivamente a la comercialización de minerales, pero a partir de las sucesivas reuniones se decidió incursionar en dos rubros adicionales, la extracción, por un lado y el fraccionado de minerales, por el otro.

BAUXITA & CO. cuenta con una oficina en la Capital Federal y varias en el interior. Cada una de estas oficinas se denomina establecimientos, y realizan distintos tipos de actividades.

El establecimiento de Capital Federal tiene por actividad principal la de administración, pero además realiza las actividades de depósito y comercialización de minerales.

Para hacer las cosas bien, Aníbal Kaddar, Gerente de Sistemas, propuso implementar algún tipo de software que permita gestionar adecuadamente el desarrollo de las nuevas actividades.

Ante la decisión de explorar una zona determinada se envía a un grupo de gente experimentada, tales como geólogos, agrimensores, técnicos y mineros a investigar el suelo. De los estudios realizados puede surgir una diversidad de minerales encontrados, pero a BAUXITA & CO. le interesa aquellos que considere más útiles. Así por ejemplo, si encuentra cal, cuando la empresa estaba buscando cemento, entonces para la empresa el resultado del estudio fue No Satisfactorio.

BAUXITA & CO., dedica gran parte de sus ingresos a estudiar el suelo, los minerales que éste contiene y a determinar la ubicación de los mismos. Los estudios del suelo tienen un costo que se debe registrar y que finaliza con un resultado determinado: Satisfactorio o No Satisfactorio. Los mismos se realizan en una determinada provincia y localidad, por un tiempo determinado. Nunca un estudio se prolonga más allá de los 27 días.

En caso de ser satisfactorio el resultado, es decir, se halló el mineral buscado, se solicita a la Dirección General de Minería la concesión de explotación. La misma es concedida a través de una Resolución, instrumento que avala la autorización para explotar un yacimiento determinado.

La Concesión es emitida a una fecha cierta, y cuenta además con un Número de Resolución, una Vigencia Desde y Hasta, una Ubicación en determinada Provincia, Localidad, Nro. de Catastro, y un Canon a percibir por parte del estado.

Una vez localizado el yacimiento la empresa se pone en marcha para buscar inversores que le permitan costear tal emprendimiento. Las fuentes de financiamiento pueden ser provenientes del mercado nacional o del exterior.

Una concesión puede ser financiada por varios inversores, necesitándose almacenar el monto aportado por cada inversor y el porcentaje de rentabilidad esperado.

Los yacimientos que se descubren pueden poseer distintos tipos de minerales, hallándose estos agrupados en familias:

- ✓ Cales.
- ✓ Carbonatos.
- ✓ Calcitas.

La rentabilidad de un mineral esta dada por los costos incurridos para su extracción, pudiendo variar para cada estudio.

BAUXITA & CO. cuenta con un numero de clientes reducido pero de importante volumen. De estos se necesita guardar: Nombre, País, Provincia, Localidad, dirección y teléfonos.

### **Consideraciones adicionales**

- ✓ Los minerales tienen un precio de lista único. Es importante para BAUXITA & CO. poder conservar un histórico de precios.
- ✓ Un cliente puede comprar varios minerales en una misma factura.
- ✓ Una familia de minerales agrupa a varios minerales.
- ✓ Un yacimiento puede contener varios minerales en él.
- ✓ Un yacimiento no es un establecimiento.
- ✓ Sobre una venta de mineral se pueden aplicar descuentos comerciales, siendo estos únicos por mineral.
- ✓ Sobre una venta se pueden aplicar un descuento por cliente, siendo este único por toda la operación de venta.



**Se pide**

1. Realizar un modelo de datos en tercera forma normal que cubra los requisitos planteados en el enunciado. Adicionalmente realizar el diagrama lógico correspondiente al modelo normalizado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos".



### I.14. Caso SUSODA

Hernán Ibarra desea implementar un sistema que le permita producir la cantidad de soda necesaria para no tener stock en grandes cantidades, y de esta forma no inmovilizar el capital de la empresa, y ahorrarle importantes costos a la empresa.

Para llevar a cabo esta tarea Hernán lo ha contratado a usted, quién al ser un estudiante avanzado de la carrera de sistemas de información está en condiciones de desarrollar un modelo de datos en 3FN para cubrir las necesidades del nuevo sistema.

En una entrevista mantenida con Luis Azcurra nos cuenta que la manera en la que trabaja la empresa es la siguiente:

Al ingresar un nuevo pedido el departamento de ventas registra el número de cliente, el o los productos solicitados, la cantidad pedida, el lugar de entrega, el plazo de entrega y la forma de pago (efectivo o cuenta corriente).

Luego de cargado el pedido ventas envía el mismo a almacenes quien prepara el pedido, de no haber suficientes unidades del producto en stock se solicita a producción la confección de las unidades faltantes. Producción con el parte recibido de almacenes procede a la fabricación de lo solicitado, previo control de las cantidades existentes en stock de los insumos necesarios para la fabricación del producto, por lo que resulta necesario poder conocer que insumos componen cada producto, la unidad de medida del mismo y la cantidad.

En caso de que producción detecte que la cantidad de insumos disponibles menos la cantidad necesaria para la elaboración del producto sea menor al punto de reposición (stock mínimo) del insumo, emite una solicitud a compras para la adquisición de los mismos, informando en dicha solicitud también la cantidad que es necesaria adquirir y el plazo de entrega máximo que debe cumplir el proveedor.

Con la llegada de las solicitudes de compra de insumos, el departamento de compras contacta a los distintos proveedores para ver si estos pueden satisfacer la demanda, solicitando que envíen una cotización en la cuál incluyan el tiempo de entrega, las cantidades y el precio del insumo. Dado que existen casos en donde el proveedor no es capaz de proveer el total de las unidades solicitadas es que resulta necesario también guardar las cantidades cotizadas.

Una vez que se tienen las diferentes cotizaciones las mismas son cargadas en el sistema y es realizada una comparación de las mismas.

Una vez seleccionada la mejor opción se emite una orden de compra, la cuál posee los siguientes datos:

Cabecera: Nombre del proveedor y Fecha emisión.

Renglón: Insumo/s a adquirirse, cantidad, precio acordado y cotización asociada.

Al momento de la recepción de los insumos comprados los mismos son ingresados en el stock y como constancia se emite un comprobante de recepción de insumos, el cuál debe ser registrado en el sistema. Los datos informados en dicho comprobante son: Orden/es de compra

relacionada/s, Fecha de recepción y cantidad recibida. Es importante poder guardar la cantidad que se recibió dado que el proveedor puede llegar a entregar de más o menos.

A medida que producción termina los productos, los mismos son enviados al sector almacenes acompañados de un formulario de nombre transferencia de productos terminados, en este se incluye: Fecha traslado, producto y cantidad. Con este formulario en almacén se procede al alta de los productos en el stock.

### **Consideraciones adicionales**

- ✓ Una misma solicitud de compra de insumos puede incluir más de un insumo.
- ✓ No todos los proveedores necesariamente deben proveer todos los insumos.
- ✓ La empresa sólo opera con clientes residentes en el país, pero los mismos pueden encontrarse en cualquier provincia.
- ✓ De los clientes es importante conocer su email y los teléfonos de contacto.
- ✓ La forma de pago de un pedido de venta es única para todo el pedido.
- ✓ El lugar y plazo de entrega de un pedido puede no ser el mismo para todos los productos pedidos.
- ✓ ? Una cotización recibida por parte del proveedor puede hacer referencia a más de una solicitud de cotización emitida.
- ✓ Toda cotización tiene una fecha de vencimiento.
- ✓ Una orden de compra puede agrupar varias cotizaciones distintas.
- ✓ Las cantidades incluidas en una orden de compra siempre son por el total de unidades informadas por el proveedor en la cotización.



#### **Se pide**

1. Realizar un modelo de datos en tercera forma normal que cubra los requisitos planteados en el enunciado. Adicionalmente realizar el diagrama lógico correspondiente al modelo normalizado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos"

**I.15. Caso CUKI S.R.L.**

CUKI S.R.L. es reconocida en el mercado por su incomparable sabor. Su fuerte es producir galletitas que gustan a todos. Además, distribuye y vende sus productos a mayoristas y minoristas.

La empresa cuenta con una administración central y la fábrica propiamente dicha. Además cuenta con varios puntos de venta (sucursales) en la Capital Federal y provincia de Bs. As. que llevan a cabo funciones de distribución y facturación en las zonas a las cuales se vende la mercadería.

Cada sucursal tiene una zona de trabajo asignada por lo que es la responsable de atender a los clientes que operen dentro de su zona.

De los clientes interesa guardar: Nombre, tipo de cliente, sucursal asociada y teléfonos de contacto.

La fábrica envía regularmente mercadería hacia estas sucursales, donde se reciben los pedidos de los clientes de la zona. Allí se preparan los remitos con la mercadería a ser enviada al cliente y se facturan dichos remitos.

En caso de no existir stock suficiente en el punto de venta, se realiza un pedido a la fábrica, para esto se utiliza un formulario de pedido el cuál debe poder ser registrado en el sistema. En dicho formulario se indica la fecha del pedido, la sucursal, los productos solicitados y las cantidades.

Los remitos correspondientes a la mercadería enviada al cliente se asocia a un pedido de venta, pudiéndose realizar envíos en forma parcial es decir que un mismo pedido se haya entregado con varios remitos, por lo que resulta importante conocer la cantidad entregada con cada remito.

Hay diversas formas de pago, estas pueden ser:

- ✓ Efectivo.
- ✓ Cheque.

De las sucursales y fábricas es importante conocer: La provincia, localidad, calle y número donde se ubican.

Compras a partir de una solicitud de materiales enviada por producción se encarga de realizar los pedidos de materias primas a los proveedores, especificándose los productos y cantidades. La forma de contactos con los proveedores es a través de alguno de los siguientes medios:

- ✓ Teléfonos.
- ✓ Mail.

En la solicitud de materiales se informa la fábrica que realiza la solicitud, los insumos pedidos y la cantidad pedida.

Las órdenes de compra emitidas se encuentran asociadas a una solicitud de materiales, dado que quizás una solicitud o renglón de solicitud se puede asignar en diferentes órdenes de compras, dado que quizás se realice el pedido a más de un proveedor, es que resulta necesario conocer la cantidad que fue comprada de cada insumo en una orden de compra.

Toda orden de compra posee una fecha de emisión y está asociada a un determinado proveedor.

Cada tipo de galleta tiene distintas composiciones de materias primas.

Los tipos de galletitas que comercializa la empresa son variados, encontrándose agrupados por Grupos y dentro de este por tipos.

Grupos:

- ✓ Crackers.
- ✓ Dulces.

Tipos:

- ✓ Sándwich con y sin sal.
- ✓ Salvado con y sin sal.
- ✓ Surtido.
- ✓ Azucaradas.
- ✓ Pepas.
- ✓ Anillos.

### **Consideraciones adicionales**

- ✓ Los pedidos recibidos de los clientes deben poder ser registrados en el sistema como así también los remitos emitidos.
- ✓ Una factura puede incluir muchos remitos.
- ✓ La forma de pago es pactada con el cliente al momento de emitir la factura.
- ✓ En toda orden de compra por pedido del área de auditoría es que resulta necesario guardar el legajo y el nombre del comprador.
- ✓ Los compradores son empleados de la compañía y a su vez cada uno de estos posee un jefe, siendo el jefe también un empleado de la empresa. La estructura jerárquica de CUKI es:
  - Gerente – Jefe – Empleado.
- ✓ Existen muchos proveedores que nos proveen la misma materia prima y un proveedor no necesariamente provee todas las materias primas.
- ✓ En la fábrica se lleva de un stock de materias primas com así también de productos.
- ✓ En un mismo remito se pueden incluir muchos pedidos de un cliente.

- ✓ En cada renglón de la orden de compra además de la cantidad solicitada también figura el precio unitario del insumo.
- ✓ La empresa maneja una lista de precios para todos los productos comercializados. Siendo importante conservar un histórico de la misma.



**Se pide**

1. Realizar un modelo de datos en tercera forma normal que cubra los requisitos planteados en el enunciado. Adicionalmente realizar el diagrama lógico correspondiente al modelo normalizado.



**Importante**

Aquellos datos que considere necesario conocer y no figuren en el relevamiento los debe suponer, colocándolos en un capítulo aparte llamado "Supuestos"



J

## **Resolución de Casos Prácticos Diseño Lógico**

---

### Objetivos

- ✓ Mostrar una solución a los casos considerados relevantes.
- ✓ Mostrar la metodología de resolución, como así también la visualización y orden de la misma.



**Importante**

Recuerde que un mismo ejercicio puede admitir en algún punto distintas soluciones a un mismo requerimiento, por lo que a continuación se presenta una solución propuesta para cada ejercicio, pudiendo alguna de ellas diferir en algún grado de la solución desarrollada por usted, ante esto resulta de vital importancia asegurarse de que lo diseñado por usted sea compatible con lo expresado en el relevamiento, no se olvide que el último paso de un buen diseño es testear que el modelo diseñado sea compatible con la realidad relevada.



### J.1. Resolución Caso COMERCOM

PROVEEDOR (#ID\_PROVEEDOR, NOMBRE, CALLE, NUMERO)

PROVEEDOR\_TELEFONO (#ID\_PROVEEDOR, #TELEFONO)

UNIDAD\_MEDIDA (#ID\_UNIDAD\_MEDIDA, DESCRIPCION)

MATERIAL (#ID\_MATERIAL, NOMBRE, ID\_UNIDAD\_MEDIDA)

MATERIAL\_PROVEEDOR (#ID\_MATERIAL, #ID\_PROVEEDOR, PRECIO)

DEPARTAMENTO (#ID\_DEPARTAMENTO, NOMBRE)

CABECERA\_PEDIDO (#NRO\_PEDIDO, FECHA, ID\_PROVEEDOR)

RENGLON\_PEDIDO (#NRO\_PEDIDO, #ID\_MATERIAL, CANTIDAD)



#### Aclaración

Los campos con # y **negrita** son la clave primaria de la tabla.



## J.2. Resolución Caso MAGIC FUN

JUEGO (#ID\_JUEGO, NOMBRE, ID\_CATEGORIA)

CATEGORIA (#ID\_CATEGORIA, NOMBRE)

SOPORTE (#ID\_SOporte, NOMBRE)

PROVEEDOR (#ID\_PROVEEDOR, NOMBRE, ID\_PAIS, ID\_PCIA, ID\_LOCAL)

PROVEEDOR\_TELEFONO (#ID\_PROVEEDOR, #TELEFONO)

PAIS (#ID\_PAIS, NOMBRE)

PROVINCIA (#ID\_PAIS, #ID\_PCIA, NOMBRE)

LOCALIDAD (#ID\_PAIS, #ID\_PCIA, #ID\_LOCALIDAD, NOMBRE)

PLATAFORMA (#ID\_PLATAFORMA, NOMBRE)

JUEGO\_PLATAFORMA (#ID\_JUEGO, #ID\_PLATAFORMA)

JUEGO\_SOporte (#ID\_JUEGO, #ID\_SOporte)

JUEGO\_PROVEEDOR (#ID\_JUEGO, #ID\_PROVEEDOR)

JUEGO\_PROV\_SOPOR\_PLAT (#ID\_JUEGO, #ID\_PROVEEDOR, #ID\_SOporte, #ID\_PLATAFORMA)

JUEGO\_PRECIO (#ID\_JUEGO, #ID\_PROVEEDOR, #ID\_SOporte, #ID\_PLATAFORMA, #FECHA, PRECIO)

CLIENTE (#ID\_CLIENTE, NOMBRE)

CABECERA\_FACTURA (#NRO\_FACTURA, FECHA, ID\_CLIENTE)

RENGLON\_FACTURA (#NRO\_FACTURA, #RENGLON, ID\_JUEGO, ID\_PROVEEDOR, ID\_SOporte, ID\_PLATAFORMA, CANTIDAD)



### Aclaración

Los campos con # y **negrita** son la clave primaria de la tabla.



## J.3. Caso SOLAR COLLEGE

ALUMNO (#ID\_ALUMNO, APELLIDO, NOMBRE, FNACIMIENTO, EMAIL)

ALUMNO\_TELEFONO (#ID\_ALUMNO, #TELEFONO)

MATERIA (#ID\_MATERIA, NOMBRE)

CURSO (#ID\_MATERIA, #ID\_CURSO, #ID\_DOCENTE)

DOCENTE (#ID\_DOCENTE, NOMBRE, APELLIDO, DIRECCION, FNACIMIENTO, ID\_CARGO, FINGRESO, ID\_JEFE)

CARGO (#ID\_CARGO, DESCRIPCION)

DIA (#ID\_DIA, DESCRIPCION)

DIA\_HORARIO (#ID\_DIA, #ID\_HORARIO, DESCRIPCION)

CURSO\_DIA (#ID\_MATERIA, #ID\_CURSO, #ID\_DIA)

CURSO\_DIA\_HORARIO (#ID\_MATERIA, #ID\_CURSO, #ID\_DIA, #ID\_HORARIO)

CURSO\_ALUMNO (#ID\_MATERIA, #ID\_CURSO, #AÑO, #CUATRIMESTRE, #ID\_ALUMNO, NOTA)

**Aclaración**

Los campos con # y **negrita** son la clave primaria de la tabla.



#### J.4. Caso RENT A KAB

PROVINCIA (#ID\_PCIA, NOMBRE)

LOCALIDAD (#ID\_PCIA, #ID\_LOCAL, NOMBRE)

CLIENTE (#ID\_CLIENTE, NOMBRE, FNACIMIENTO, ID\_PCIA, ID\_LOCAL, CALLE, NÚMERO, PISO, DEPTO)

CLIENTE\_TELEFONO (#ID\_CLIENTE, #TELEFONO)

CHOFER (#ID\_CHOFER, NOMBRE, APELLIDO, DIRECCION, DNI, NRO\_REGISTO)

MARCA (#ID\_MARCA, DESCRIPCION)

MODELO (#ID\_MODELO, DESCRIPCION)

TITULAR (#ID\_TITULAR, NOMBRE)

AUTO (#ID\_AUTO, ID\_MARCA, COLOR, AÑO, ID\_TITULAR, ID\_MODELO)

TURNO (#ID\_TURNO, DESCRIPCION)

AUTO\_CHOFER (#ID\_AUTO, #ID\_TURNO, #FECHA\_DESDE, ID\_CHOFER)

FORMA\_PAGO (#ID\_FORMA\_PAGO, DESCRIPCION)

VIAJE (#ID\_VIAJE, FECHA, ID\_CLIENTE, CANTIDAD\_PASAJEROS, DESTINO, ID\_FORMA\_PAGO, DEMORA\_APROXIMADA, ID\_CHOFER\_ASIGNADO, PRECIO\_APROXIMADO)

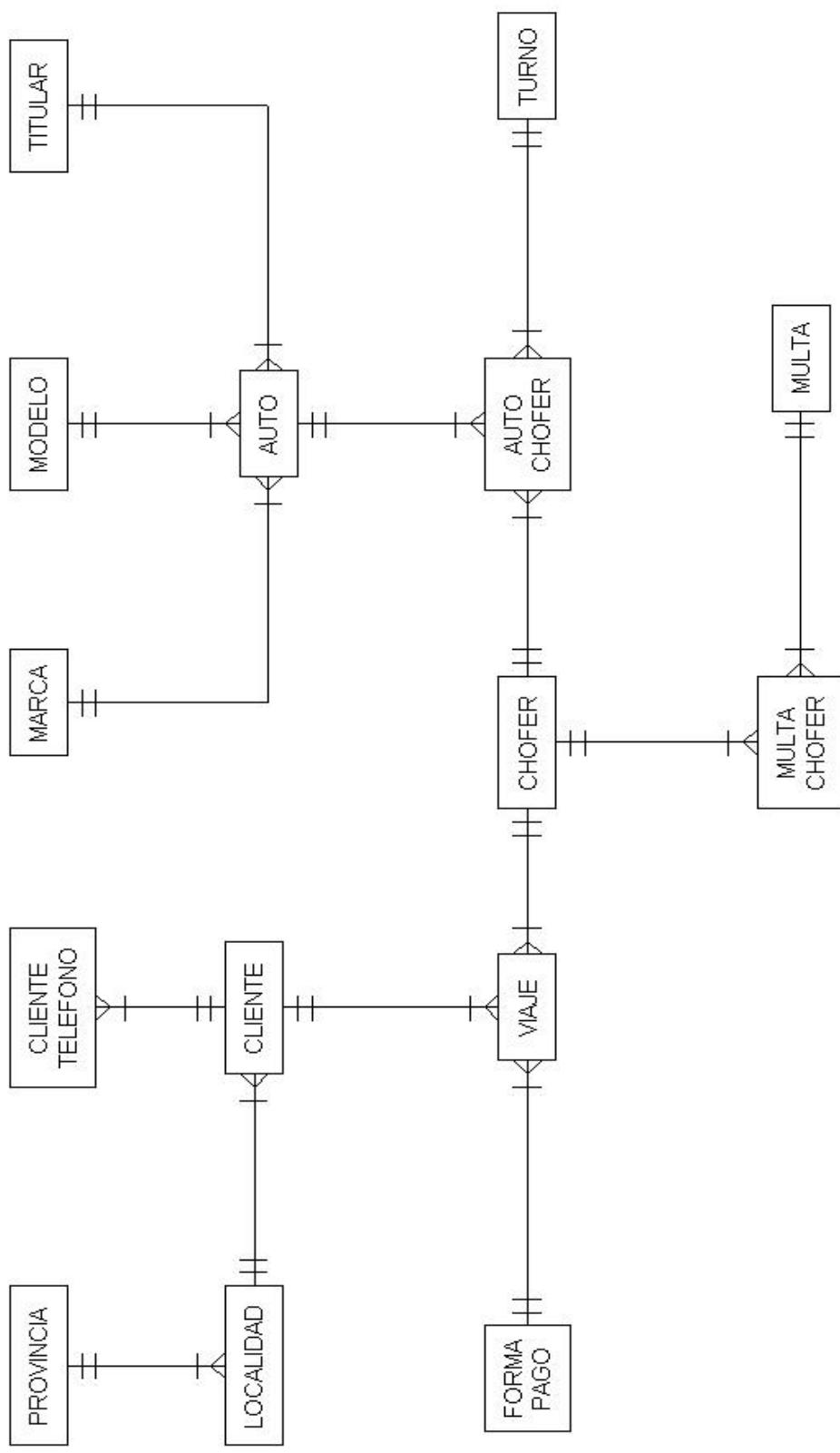
MULTA (#ID\_MULTA, DESCRIPCION, CANTIDAD\_DIAS\_SUSPENSION)

MULTA\_CHOFER (#ID\_CHOFER, #FECHA\_DESDE\_SUSPENSION, ID\_MULTA)



#### Aclaración

Los campos con # y **negrita** son la clave primaria de la tabla.





## J.5. Caso TERRA CONTROL

### Errores encontrados

- ✓ La tabla CLIENTE y CLIENTE\_DIRECCION poseen la misma clave primaria por lo que se las puede fusionar dado que sus campos depende en forma completa de la misma clave.
- ✓ El teléfono es un grupo repetitivo por lo que no se está cumpliendo con la primera forma normal.
- ✓ La tabla CLIENTE\_CLAVE no corresponde ya que la clave que utiliza un cliente para ingresar al sistema es una sola, o sea no es un grupo repetitivo por lo que debe ser un campo de la tabla CLIENTE.
- ✓ El campo nombre\_provincia no depende de la clave primaria de la tabla LOCALIDAD por lo que dicha tabla no se encuentra en 3FN.
- ✓ En la tabla SERVICIO el campo descripción no corresponde que forme parte de la clave primaria.
- ✓ En la tabla SERVICIO el campo precio no depende en forma completa de la clave primaria de la tabla.
- ✓ En la tabla MARCA\_MODELO el campo nombre\_marca no depende de la clave primaria de la tabla.
- ✓ Falta una tabla en la cuál se encuentran codificados los distintos modelos para un auto.
- ✓ En la tabla RENGLON\_PRESUPUESTO el campo clave no depende de la clave primaria de la tabla, sino que depende del cliente por lo que corresponde que sea un campo de la tabla CLIENTE.
- ✓ En la tabla RENGLON\_PRESUPUESTO los campos fecha y dni poseen siempre el mismo valor para todos los renglones del presupuesto por lo que no son parte del grupo repetitivo por lo que lo correcto es que formen parte de la tabla CABECERA\_PRESUPUESTO.

### Modelo corregido

CLIENTE (#**DNI**, MAIL, ID\_PCIA, ID\_LOCAL, CALLE, NUMERO, PISO, DEPARTAMENTO, CLAVE)

CLIENTE\_TELEFONO (#**DNI**, #**TELEFONO**)

PROVINCIA (#**ID\_PCIA**, NOMBRE)

LOCALIDAD (#**ID\_PCIA**, #**ID\_LOCAL**, NOMBRE)

SERVICIO (#ID\_SERVICIO, DESCRIPCION, CANTIDAD\_DIAS\_TRABAJO)

PRECIO\_SERVICIO (#ID\_SERVICIO, #FECHA, PRECIO)

MARCA (#ID\_MARCA, NOMBRE)

MODELO (#ID\_MODELO, #ID\_MODELO, NOMBRE)

AÑO (#ID\_AÑO)

MODELO\_AÑO (#ID\_MODELO, #ID\_MARCA, #ID\_AÑO)

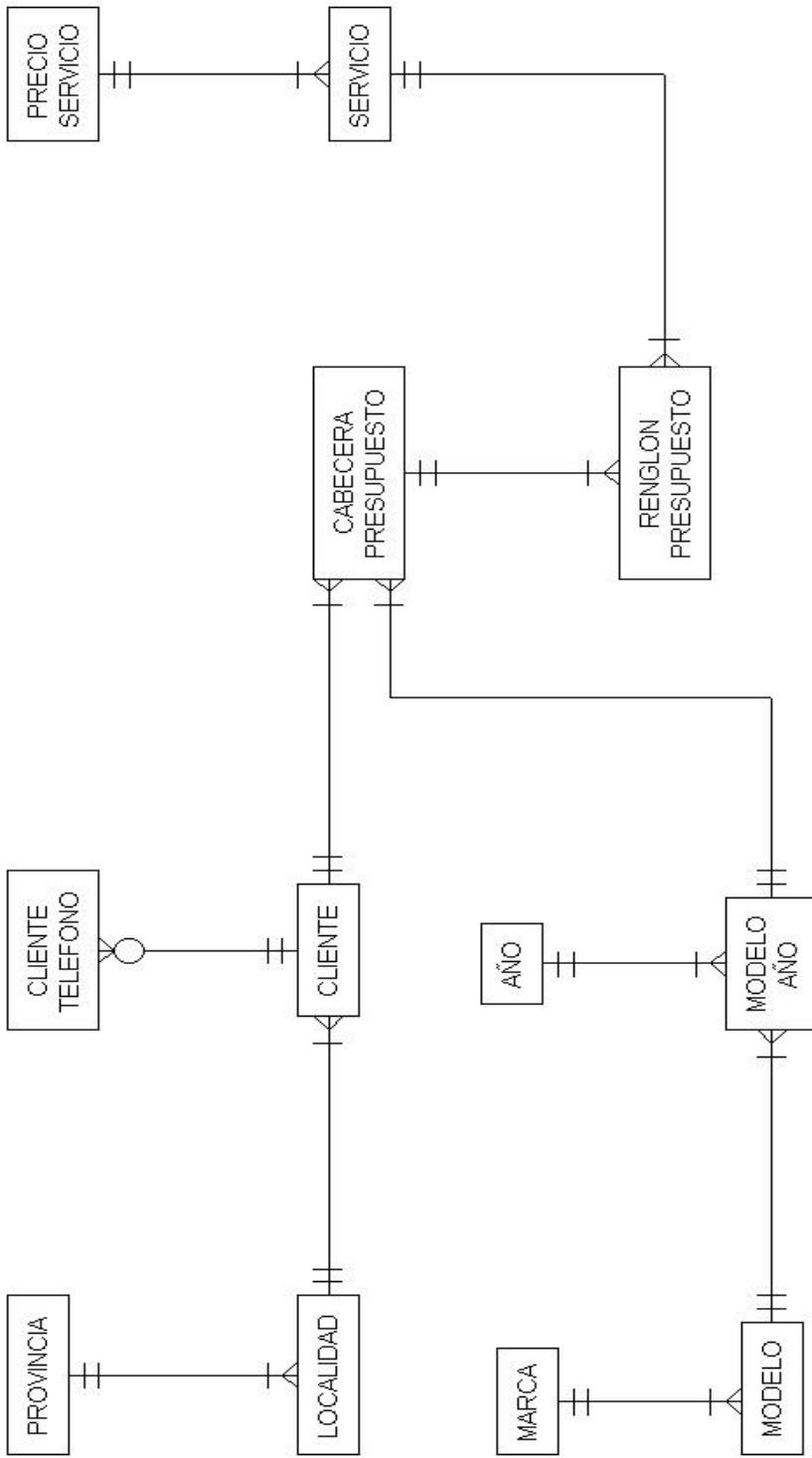
CABECERA\_PRESUPUESTO (#ID\_PRESUPUESTO, FECHA, DNI, ID\_MARCA, ID\_MODELO, AÑO)

RENGLON\_PRESUPUESTO (#ID\_PRESUPUESTO, #ID\_SERVICIO)



**Aclaración**

Los campos con # y **negrita** son la clave primaria de la tabla.





## J.6. Caso BAUXITA &amp; CO

FAMILIA\_MINERAL (#ID\_FAMILIA, NOMBRE)

MINERAL (#ID\_MINERAL, NOMBRE, ID\_FAMILIA)

ESTUDIO (#ID\_ESTUDIO, ID\_PAIS, ID\_Pcia, ID\_LOCAL, FECHA\_INICIO, FECHA\_FIN, MINERAL\_BUSCADO, COSTO\_ESTUDIO, RESULTADO\_ESTUDIO)

PAIS (#ID\_PAIS, NOMBRE)

PROVINCIA (#ID\_PAIS, #ID\_Pcia, NOMBRE)

LOCALIDADL (#ID\_PAIS, #ID\_Pcia, #ID\_LOCAL, NOMBRE)

PROFESIONAL (#ID\_PROFESIONAL, NOMBRE, ID\_PROFESION)

PROFESION (#ID\_PROFESION, NOMBRE)

ESTUDIO\_PROFESIONAL (#ID\_ESTUDIO, #ID\_PROFESIONAL)

MINERAL\_ENCONTRADO\_ESTUDIO (#ID\_ESTUDIO, #ID\_MINERAL, COSTO\_EXTRACCION)

CONCESION (#ID\_CONCESION, ID\_ESTUDIO, NRO\_RESOLUCION, FECHA\_DESDE, FECHA\_HASTA, NRO\_CATASTRO, CANON\_A\_PAGAR)

INVERSOR (#ID\_INVERSOR, NOMBRE, ID\_PAIS, ID\_Pcia, ID\_LOCAL)

INVERSOR\_CONCESION (#ID\_INVERSOR, #ID\_CONCESION, MONTO, PORCENTAJE\_RENTABILIDAD)

MINERAL\_PRECIO (#ID\_MINERAL, #FECHA, PRECIO)

MINERAL\_DESCUENTO (#ID\_MINERAL, #FECHA, PORCENTAJE\_DESCUENTO)

CLIENTE (#ID\_CLIENTE, NOMBRE, ID\_PAIS, ID\_Pcia, ID\_LOCAL, CALLE, NUMERO, PISO, OFICINA)

CLIENTE\_TELEFONO (#ID\_CLIENTE, #TELEFONO)

CLIENTE\_DESCUENTO (#ID\_CLIENTE, #FECHA, PORCENTAJE\_DESCUENTO)

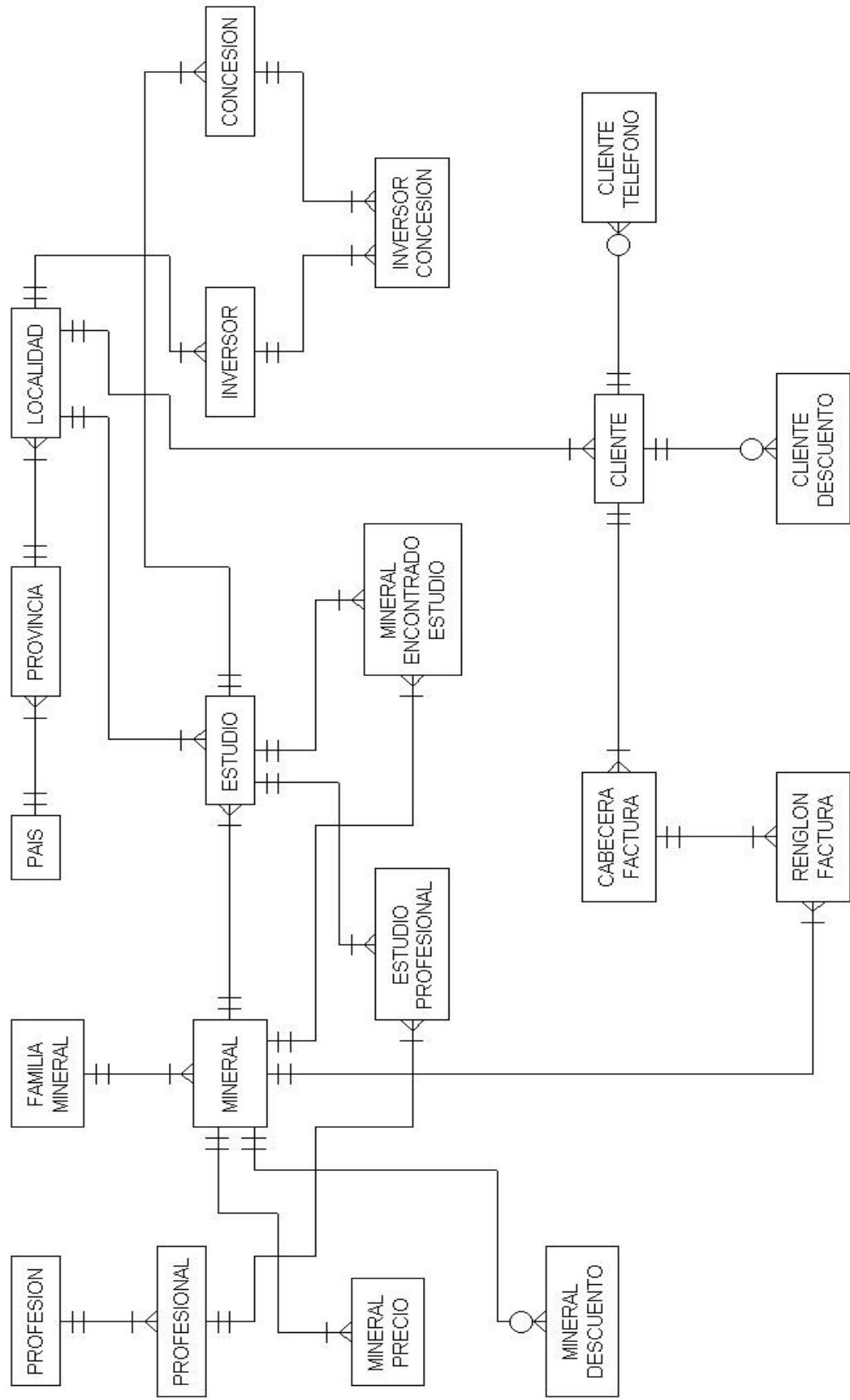
CABECERA\_FACTURA (#NRO\_FACTURA, FECHA, ID\_CLIENTE)

RENGLON\_FACTURA (#NRO\_FACTURA, #RENGLON, ID\_MINERAL, CANTIDAD)



**Aclaración**

Los campos con # y **negrita** son la clave primaria de la tabla.



**J.7. Caso CUKI S.R.L.**

TIPO\_CLIENTE (#ID\_TIPO\_CLIENTE, DESCRIPCION)

PROVINCIA (#ID\_PCIA, NOMBRE)

LOCALIDAD (#ID\_PCIA, #ID\_LOCAL, NOMBRE)

ZONA (#ID\_ZONA, NOMBRE)

SUCURSAL (#ID\_SUCURSAL, NOMBRE, ID\_PCIA, ID\_LOCAL, CALLE, NRO, ID\_ZONA)

CLIENTE (#ID\_CLIENTE, NOMBRE, ID\_TIPO\_CLIENTE, ID\_SUCURSAL)

CLIENTE\_TELEFONO (#ID\_CLIENTE, #TELEFONO)

GRUPO (#ID\_GRUPO, NOMBRE)

TIPO (#ID\_GRUPO, #ID TIPO, NOMBRE)

PRODUCTO (#ID\_PRODUCTO, NOMBRE, ID\_GRUPO, ID\_TIPO)

LISTA\_PRECIO\_PRODUCTO (#ID\_PRODUCTO, #FECHA, PRECIO)

FABRICA (#ID\_FABRICA, NOMBRE, ID\_PCIA, ID\_LOCAL, CALLE, NRO)

STOCK\_PRODUCTO\_FABRICA (#ID\_PRODUCTO, #ID\_FABRICA, #FECHA, CANTIDAD)

STOCK\_PRODUCTO\_SUCURSAL (#ID\_PRODUCTO, #ID\_SUCURSAL, #FECHA, CANTIDAD)

CABECERA\_PEDIDO\_CLIENTE (#NRO\_PEDIDO\_CLI, FECHA, ID\_CLIENTE)

RENGLON\_PEDIDO\_CLIENTE (#NRO\_PEDIDO\_CLI, #ID\_PRODUCTO, CANTIDAD)

CABECERA\_PEDIDO\_SUCURSAL (#NRO\_PEDIDO\_SUC, FECHA, ID\_SUCURSAL)

RENGLON\_PEDIDO\_SUCURSAL (#NRO\_PEDIDO\_SUC, #ID\_PRODUCTO, CANTIDAD)

CABECERA\_REMITO (#NRO\_REMITO, FECHA, ID\_CLIENTE)

RENGLON\_REMITO (#NRO\_REMITO, #NRO\_PEDIDO\_CLI, #ID\_PRODUCTO, CANTIDAD)

FORMA\_PAGO (#ID\_FORMA\_PAGO, DESCRIPCION)

CABECERA\_FACTURA (#NRO\_FACTURA, FECHA, ID\_CLIENTE, ID\_FORMA\_PAGO)

RENGLON\_FACTURA (#NRO\_FACTURA, #NRO\_REMITO, #ID\_PRODUCTO)

INSUMO (#ID\_INSUMO, NOMBRE)

STOCK\_INSUMO (#ID\_INSUMO, #FECHA, CANTIDAD)

CABECERA\_SOLICITUD\_MATERIAL (#NRO\_SOLICITUD, FECHA, ID\_FABRICA)

RENGLON\_SOLICITUD\_MATERIAL (#NRO\_SOLICITUD, #ID\_INSUMO, CANTIDAD)

PROVEEDOR (#ID\_PROVEEDOR, NOMBRE, MAIL)

PROVEEDOR\_TELEFONO (#ID\_PROVEEDOR, #ID\_INSUMO)

PROVEEDOR\_INSUMO (#ID\_PROVEEDOR, #ID\_INSUMO)

COMPOSICION\_INSUMO\_PRODUCTO (#ID\_PRODUCTO, #ID\_INSUMO, CANTIDAD)

EMPLEADO (#ID\_LEGATO, NOMBRE, ID\_LEGATO\_JEFE)

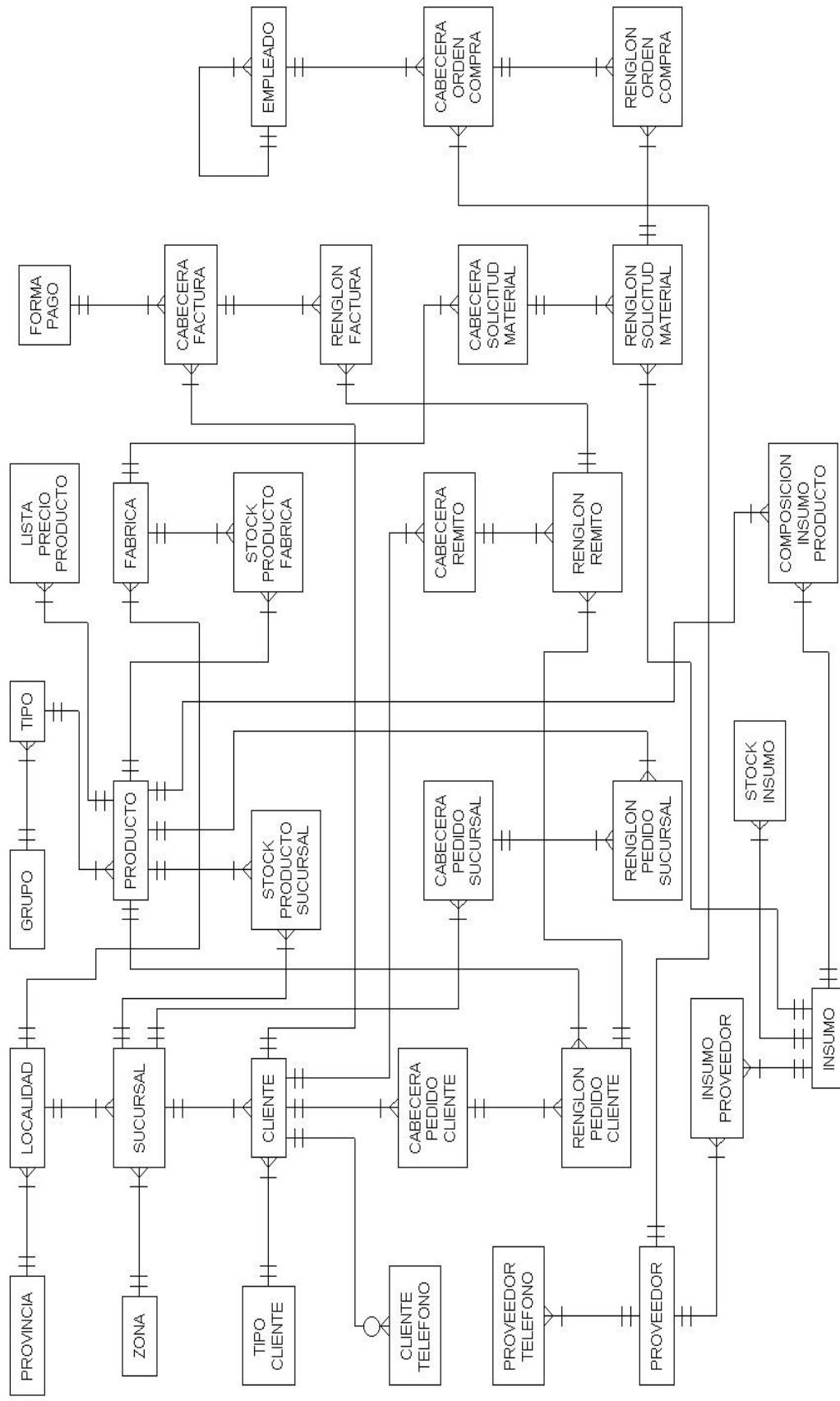
CABECERA\_ORDEN\_COMPRA (#NRO\_OC, FECHA, ID\_PROVEEDOR, ID\_LEGATO\_COMPRADOR)

RENGLON\_ORDEN\_COMPRA (#NRO\_OC, #NRO\_SOLICITUD, #ID\_INSUMO, CANTIDAD,  
PRECIO\_UNITARIO)



#### Aclaración

Los campos con # y **negrita** son la clave primaria de la tabla.



# K

## Lenguaje de Definición de Datos

---

### Objetivos

- ✓ Entender como plasmar un modelo lógico de datos.
- ✓ Manejar formas de persistir una base de datos.
- ✓ Adquirir habilidades en la sintaxis de comandos DDL.
- ✓ Lograr el entendimiento de comandos básicos que persisten estructura

## K. Lenguaje de Definición de Datos

En este capítulo abordaremos la última de las fases del diseño y modelado de datos, abandonando el precepto que hemos seguido a lo largo del proceso de diseño: "Abstraernos o aislarlos del Motor de Base de datos a utilizar, considerando sólo requisitos funcionales para lograr en primera instancia un DER (Diseño Conceptual) y a partir de este un modelo de datos Normalizado en tercera forma normal (Diseño Lógico)"

A partir de este momento cada decisión que tomemos deberá ser factible dentro del Sistema de Gestión de Base datos que dará soporte a nuestro modelo.

Existen varios puntos de vista respecto del alcance de esta fase. Hay quienes por ejemplo consideran que debería abarcar hasta el monitoreo y Tuning de base de datos pasando por el diseño de mecanismos de seguridad y reglas de acceso.

Nuestro enfoque es mucho más sencillo y se limita la construcción a través del Lenguaje de Definición de Datos (DDL) las estructuras que representarán dentro de nuestra base las tablas, campos y relaciones existentes en nuestro modelo lógico, dado que consideramos que el resto de las tareas mencionadas son responsabilidad de otras áreas y que exceden a un proyecto en particular (ej. Políticas de seguridad corporativas, Normas de auditoría, disponibilidad y características de Hardware, características y volumen real de transacciones entre otras).

Por último resulta útil mencionar que a partir de 1986 y hasta la actualidad el ANSI (American National Standard Institute) supervisa el desarrollo de los estándares del SQL (y del DDL como subconjunto de este), a pesar de tener que todos los motores de base de datos soportan ANSI hay casos en donde encontraremos cláusulas propietarias que no responden al estándar.

### Lenguaje de Definición de datos

El lenguaje de Definición de datos, en inglés Data Definition Language (DDL), se encarga de la creación y modificación de la estructura de los objetos de la base de datos.

Los siguientes operadores nos permitirán crear, definir y modificar las características de los mismos:

- ✓ CREATE
- ✓ ALTER
- ✓ DROP

### **CREATE**

El comando CREATE se utiliza para crear objetos dentro de la base de datos.

La cantidad y naturaleza de los objetos soportados depende exclusivamente del motor de base de datos en el cual implementaremos nuestro modelo, siendo los más básicos y los que interesan a los fines de este libro los siguientes:

Objeto	Descripción
Tabla	Unidad basica de almacenamiento compuesta por registros y columnas. Es la representacion física de las entidades.
Vista	Representacion lógica de un subconjunto de datos de una o mas tablas.
Indices	Estructura de almacenamiento relacionada a las tablas para proveer mejor rendimiento de las consultas sobre la tabla.
Restricciones	Fuerzan el cumplimiento de condiciones (Ej. reglas de negocio) en registros datos de una tabla. Son la representacion física de las relaciones.

Cuadro 1

Los motores de bases de datos también soportan objetos de características y complejidad más avanzada, como por ejemplo: Sinónimos, Cluster de tablas, tablas temporales, clases Java, Store procedures (Funciones,procedimientos y paquetes) entre otras.

### Sintaxis

```
CREATE TABLE [NOMBRE_TABLA]
(
    Nombre_campo      tipodatos (extensión) Null/Not Null  Restricción
    .....
    .....
);
```

#### Donde

**[NOMBRE\_TABLA]:** Es el nombre que llevará la tabla.

**Nombre\_campo:** Es el nombre del campo que formará parte de la tabla.

**Tipodatos (extensión):** Permite indicar el tipo de información que se podrá ingresar dentro del campo, a su vez y solo para algunos casos también se deberá indicar la cantidad de caracteres que se podrán ingresar (extensión). Este punto se explicará con mayor detalle más adelante.

**Null/Not Null:** Permite indicar si el campo podrá o no contener información nula. La nulidad de una columna se refiere a si se requiere o no una entrada para esa columna. Si desea permitir que una columna indique que el valor es desconocido, especifique NULL. Si desea insistir que cada fila tenga una entrada en esa columna, especifique NOT NULL. Si no especifica NULL o NOT NULL, se utilizará el valor predeterminado, el cuál es que las columnas se creen con la opción NULL.

### Restricción

Permite indicar si el uso del campo será limitado. Este punto se explicará con mayor detalle más adelante.

### Tipos de datos

Como dijimos anteriormente especifica el tipo de información a ingresar dentro del campo creado.

Los tipos de datos de SQL pueden dividirse en los siguientes grupos: cadenas, binarios, enteros, numéricos con aproximación o exactos y de fecha / hora (entre otros).

La siguiente tabla muestra los tipos de datos básicos soportados por MS SQL Server:

Categoría	Tipo de datos	Comentarios
Cadena	Char(n)	Guarda cadenas de caracteres de longitud fija
	Varchar(n)	Guarda cadenas de caracteres de longitud variable.
Binario	Binary (n)	Guarda información binaria en pares de 2 bytes
Entero	Int, smallint	Guarda valores enteros.
Numérico	Float, Real	Guarda información numérica aproximada.
	Decimal, Numeric	Guarda información numérica con precisión y escala fija.
Fecha y Hora	Datetime, Smalldatetime	Guarda fechas y horas.

**Cuadro 2**

### Restricciones (Constraints)

Las restricciones forman parte de la definición de las tablas y permiten:

- ✓ Forzar la unicidad del conjunto de columnas que definen la clave primaria (Primary Key).
- ✓ Forzar la unicidad de una columna (Unique).
- ✓ Forzar valores predeterminados (Default).
- ✓ Forzar la existencia de valores (Not Null).
- ✓ Restringir el dominio de un campo (Check).
- ✓ Forzar integridad referencial (Foreign Key). Es decir restringir el conjunto de los valores que puede tomar la columna, los cuales estarán determinados por un dominio específico. Este tipo de restricciones establecen una jerarquía del tipo padre-Hijo entre la tabla referenciada (tabla padre) y la referente (tabla hijo).

Ahora analizaremos cada una en particular:

### **PRIMARY KEY**

La primary key o clave primaria es un campo o una combinación de campos que identifican únicamente cada registro de una tabla. La definición de este tipo de restricción puede hacerse a nivel de columna o de tabla.

#### **Definición a nivel de columna**

Este tipo de definición sólo es admitida en los casos donde la misma no sea compuesta.

#### **Ej.**

Si deseáramos crear la tabla Curso y su primary key es el campo codigo\_curso, la sintaxis será:

```
CREATE TABLE Curso (
  codigo_curso int      PRIMARY KEY,
  nombre char(40) Not Null,
  .....
  .....
  .....
)
```

#### **Definición a nivel de tabla**

Esta forma es valida tanto para el caso anterior como para el caso de clave primaria compuesta (formada por dos o mas campos).

Siguiendo con el ejemplo utilizado en el punto anterior veamos como crear una primary key a nivel de tabla para el campo codigo\_curso:

```
CREATE TABLE Curso (
  codigo_curso int,
  nombre char(40) Not Null,
  .....
  .....
  .....
CONSTRAINT pk_curso PRIMARY KEY (codigo_curso),
  .....
)
```

Si la tabla Curso en cambio tuviera como clave primaria los campos codigo\_curso y codigo\_categoria, la forma de definición de la misma sería:

```
CREATE TABLE Curso (
  codigo_curso int,
  codigo_categoria int,
  nombre char(40) Not Null,
  .....
  .....
  .....
```

```
CONSTRAINT pk_curso PRIMARY KEY (codigo_curso, codigo_categoria),
```

```
.....  
)
```



#### Importante

La constraint Primary Key tiene implícito una restricción de tipo Not Null, ya que al generarla estamos declarando la “no aceptación” de valores nulos. Si al momento de la creación de la Primay Key de una tabla indicamos las dos condiciones el motor de la Base de Datos nos arrojara un error, es por ello que el siguiente ejemplo NO es válido.

```
CREATE TABLE Curso (
  codigo_curso int      PRIMARY KEY NOT NULL,
  nombre char(40) Not Null,
  .....
  .....
  .....
)
```

#### FOREIGN KEY

La restricción Foreign Key es una columna o combinación de columnas sobre la/s que se establece un vínculo de integridad referencial entre datos de dos tablas.

#### Representan una jerarquía Padre-Hijo donde

- ✓ Los valores de los campos de la tabla referenciada (Tabla Padre) mas el valor Null forman el conjunto de valores posibles que pueden tomar los campos de la tabla que realizan la referencia (Tabla Hijo).
- ✓ Los campos referenciados de la Tabla Padre deben ser la clave primaria de la misma, de lo contrario no es posible crear la restricción.

La definición de este tipo de restricción puede realizarse a nivel de columna o tabla.

#### Definición a nivel de columna

Sólo se puede realizar a este nivel para claves simples.

```
CREATE TABLE NOMBRE_TABLA(
  Campo1 Tipo de Dato CONSTRAINT Nombre FOREIGN KEY REFERENCES Tabla_Padre(Campo1) [ON
  DELETE {CASCADE|SET NULL}],
  ...
)
```

Ej. Si deseamos generar una Foreign key para el campo Código\_categoria de la tabla Curso referenciando al campo Código\_categoria de la tabla categoría, la sintaxis a utilizar es:

```
CREATE TABLE Curso (
    código_curso int,
    nombre char(40) Not Null,
    código_categoria number CONSTRAINT fk_curso_categoria FOREIGN KEY REFERENCES categoría
    (código_categoria)
    ...
)
```

#### Definición a nivel de tabla

Se puede realizar a este nivel para claves simples o compuestas.

```
CREATE TABLE NOMBRE_TABLA(
    Campo1 Tipo de Dato,
    ...
    CONSTRAINT Nombre Constraint FOREIGN KEY (Campo1,Campo2) REFERENCES Tabla
    Padre(Campo1,Campo2) [ON DELETE {CASCADE|SET NULL}]
)
```

El modificador ON DELETE no es obligatorio.

Cuando se intenta eliminar un registro de una tabla Padre que participa en una clave foránea, el motor de base datos validará que no exista ningún registro dependiente (que tenga ese valor) en la tabla hija.

Este modificador acepta dos opciones:

Opción	Acción
Cascade	Eliminará automáticamente los registros dependientes en la tabla hija
Set Null	Convertirá los campos de los registros dependientes de la tabla hija en NULL.

Cuadro 3

Siguiendo con el ejemplo utilizado en el punto anterior veamos como crear una Foreign key a nivel de tabla:

```
CREATE TABLE Curso (
    código_curso int,
    nombre char(40) Not Null,
    CONSTRAINT fk_curso_categoria FOREIGN KEY (código_categoria) REFERENCES categoría
    (código_categoria)
```

....

)

### CHECK

Son restricciones de dominio que define una condición que deben satisfacer todos los registros de la tabla. Permiten restringir el conjunto de valores que puede contener un campo.

Solo puede referenciar una columna y debe ser de la misma tabla.

La definición de este tipo de restricción puede realizarse a nivel de columna o tabla.

Definición a nivel de columna:

```
CREATE TABLE NOMBRE_TABLA (
    Campo1      int primary key,
    ...
    ...
    ...
    Campo3      int check (campo3 CONDICION)
)
```

Donde **CONDICION** debe ser cualquier predicado.

Cuando se intente dar de alta o modificar un registro, el motor de base de datos evaluará la condición y resolverá la sentencia según la siguiente tabla:

Resultado Evaluacion	Accion
FALSO	Viola la restriccion. La sentencia retorna error.
VERDADERO	No viola la restriccion. La sentencia NO retorna error.
DESCONOCIDO (campo con valor nulo)	NO viola la restriccion. La sentencia NO retorna error.

**Cuadro 4**

Ej. Generación de una restricción Check para el campo duración de la tabla curso, indicando que el mismo deberá contener valores mayores a cero.

```
CREATE TABLE Curso (
    codigo_curso      int primary key,
    nombre            char(40) Not Null,
    descripcion       char(40) Not Null,
    duracion          int check (duración > 0),
    codigo_categoria int Not Null
)
```

### Definición a nivel de tabla

```
CREATE TABLE NOMBRE_TABLA (
    Campo1      int primary key,
    ...          char(40) Not Null,
    ...          char(40) Not Null,
    Campo3      int,
    CONSTRAINT "Nombre Restricción" CHECK(camp1 CONDICION)
)
```

Siguiendo con el ejemplo utilizado en el punto anterior veamos como crear una Check a nivel de tabla:

```
CREATE TABLE Curso (
    codigo_curso int primary key,
    nombre        char(40) Not Null,
    descripcion   char(40) Not Null,
    duracion      int Not Null,
    codigo_categoria int Not Null,
    CONSTRAINT ck_curso_duracion CHECK (duracion > 0)
)
```

Con lo visto hasta el momento veamos como sería la creación de la tabla curso integrando todas las restricciones indicadas. Procederemos a crear la tabla Curso correspondiente al caso Aprendiendo, donde:

- ✓ Código\_curso será Primary Key.
- ✓ Código\_categoria sera Foreign Key referenciando al campo código\_categoria de la tabla categoría.
- ✓ Código\_curso\_requisito sera Foreign Key referenciando al campo código\_curso de la tabla curso.
- ✓ Duración solo admitirá valores superiores a 0.

### La sintaxis para la creación de la tabla Curso del caso aprendiendo será

```
CREATE TABLE Curso (
    codigo_curso int,
    nombre        char(40) Not Null,
    descripcion   char(40) Not Null,
    duracion      int Not Null,
    codigo_categoria int Not Null,
    codigo_curso_requisito int Not Null,
    CONSTRAINT pk_curso PRIMARY KEY (codigo_curso),
    CONSTRAINT fk_curso_categoria FOREIGN KEY (codigo_categoria) REFERENCES categoria (codigo_categoria),
    CONSTRAINT fk_curso_requisito FOREIGN KEY (codigo_curso_requisito) REFERENCES curso (codigo_curso),
    CONSTRAINT ck_curso_duracion CHECK (duracion > 0))
```

**ALTER**

El comando ALTER se utiliza para modificar la estructura de un objeto.

Se pueden agregar/quitar/modificar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, Agregar o eliminar restricciones, etc.

Sintaxis:

Acción a realizar	Objeto	Sintaxis completa
Añadir	Columna	Alter table NombreTabla <b>Add</b> Column NombreColumna Tipo de dato
	Primary Key	Alter table NombreTabla <b>Add</b> Constraint NombreRestriccion Primary Key(column1,...,column_n)
	Foreign Key	Alter Table NombreTabla <b>Add</b> Constraint NombreRestriccion Foreign Key (campo) References TablaPadre(campo)
	Check	Alter Table NombreTabla <b>Add</b> Constraint NombreRestriccion Check (campo CONDICION)
Modificar	Columna	Alter Table NombreTabla <b>Modify</b> (campo1 tipo, Campo2 tipo, campo_n tipo)
Eliminar	Columna	Alter table NombreTabla <b>Drop</b> NombreColumna [Cascada Constraint] (*)
	Primary Key	Alter table NombreTabla <b>Drop</b> Constraint Nombre
	Foreign Key	
	Check	

Cuadro 5

(\*) La cláusula *Cascade Constraint* es opcional. Si la columna a eliminar es referenciada por una clave foránea, esta restricción es automáticamente eliminada.

Si se intenta eliminar una columna referenciada y no se utiliza este modificador, la ejecución de la sentencia terminará en error.

**Ejemplos**

Para añadir la columna comentario a la tabla Curso, debemos ejecutar la siguiente sentencia:

```
ALTER TABLE Curso add comentario char (40) Not Null
```

Para eliminar el campo duración de la tabla Curso, debemos ejecutar la siguiente sentencia:

```
ALTER TABLE Curso drop duración
```

**Para agregar una restricción de tipo check al campo código\_empresa de la tabla Curso para que solo admita valores superiores a 1:**

```
ALTER TABLE Curso add CONSTRAINT ck_cod_empresa CHECK (codigo_empresa > 1)
```

**Para eliminar la foreign key fk\_curso\_requisito de la tabla Curso:**

```
ALTER TABLE Curso DROP CONSTRAINT fk_curso_requisito
```

**Para agregar nuevamente la foreign key fk\_curso\_requisito a la tabla Curso**

```
ALTER TABLE Curso add CONSTRAINT fk_curso_requisito FOREIGN KEY (codigo_curso_requisito)
REFERENCES curso (codigo_curso),
```

## DROP

El comando DROP se utiliza para eliminar objetos dentro de la base de datos.

### Sintaxis

```
DROP TABLE [NOMBRE_TABLA]
```

**Para eliminar la tabla Curso**

```
DROP TABLE Curso
```



### Importante

Con lo visto en este capítulo estamos en condiciones de resolver el punto 3 y 4 del caso Aprendiendo

```
CREATE TABLE ASISTENTE (
ID_ASISTENTE      INT PRIMARY KEY,
NOMBRE             CHAR(40),
APELLIDO          CHAR(40),
NRO_DOC            INT,
FECHA_NACIMIENTO DATETIME,
ID_NACIONALIDAD   INT,
ID_SEXO            CHAR(1),
CALLE              CHAR(40),
NUMERO             INT,
PISO               INT,
DEPTO              CHAR(3),
ID_EMPRESA         INT,
CONSTRAINT FK_ASISTENTE_EMP FOREIGN KEY (ID_EMPRESA)
REFERENCES EMPRESA (ID_EMPRESA),
CONSTRAINT FK_ASISTENTE_NACIONALIDAD FOREIGN KEY (ID_NACIONALIDAD) REFERENCES
NACIONALIDAD (ID_NACIONALIDAD),
CONSTRAINT FK_ASISTENTE_SEXO FOREIGN KEY (ID_SEXO) REFERENCES SEXO (ID_SEXO)
)
```

```
CREATE TABLE CURSO (
ID_CURSO INT,
NOMBRE CHAR(40) NOT NULL,
DESCRIPCION CHAR(40) NOT NULL,
DURACION INT NOT NULL,
ID_CATEGORIA NOT NULL,
ID_CURSO_REQUISITO NOT NULL,
CONSTRAINT PK_CURSO PRIMARY KEY (ID_CURSO),
CONSTRAINT FK_CURSO_CATEGORIA FOREIGN KEY (ID_CATEGORIA)
REFERENCES CATEGORIA (ID_CATEGORIA),
CONSTRAINT FK_CURSO_CURSO_REQUISITO FOREIGN KEY (ID_CURSO_REQUISITO) REFERENCES
CURSO (ID_CURSO),
CONSTRAINT CK_CURSO_DURACION CHECK (DURACION > 0)
)
```



Puede profundizar más estos conceptos leyendo el capítulo 3 de “**Un Enfoque Práctico de SQL**” 2da Edición de Morteo – Bocalandro. Edit. Ediciones Cooperativas.

L

## Casos Prácticos de DDL

---



### L.1. Caso CUKI S.R.L



#### Se pide

1. Realizar la creación de la tabla en la cuál se guardan los datos de cada sucursal y el stock que hay de cada producto en la fábrica, para cada tabla deberá definirse la clave primaria, las claves foráneas, las restricciones de obligatoriedad (Not Null) y todas las restricciones de dominio (check) que considere necesarias.
2. Adicionalmente deberá realizarse la creación de las tablas a las cuales se hace referencia en alguna clave foránea. Para estas tablas debe definirse la clave primaria, las claves foráneas, las restricciones de obligatoriedad y las restricciones de dominio que considere necesarias. Y en estas también debe realizarse la creación de las tablas a las que se hace referencia en alguna clave foránea.



#### Importante

Algunas de las soluciones a los ejercicios planteados en éste capítulo se las pueden encontrar en el capítulo **M. Resoluciones de DDL**.



## L.2. Caso MAGIC FUN



### Se pide

1. Realizar la creación de las tablas correspondientes, para cada tabla deberá definirse la clave primaria y las claves foráneas que considere necesarias.



### L.3. Caso ABC MOVIL



#### Se pide

1. Realizar la creación de las tablas correspondientes, para cada tabla deberá definirse la clave primaria y las claves foráneas que considere necesarias.



#### L.4. Caso TERRA CONTROL



##### Se pide

1. Realizar la creación de las tablas definiendo para cada tabla la clave primaria, las claves foráneas y las restricciones de obligatoriedad que considere necesarias. En la tabla en la cuál se registran los clientes validar que el dominio de los campos número y piso correspondientes a la dirección solo acepten números mayores a cero y además validar que la longitud del campo password sea siempre de seis caracteres.



### L.5. Caso SUSODA



#### Se pide

1. Realizar la creación de las tablas correspondientes, para cada tabla deberá definirse la clave primaria y las claves foráneas que considere necesarias.



#### L.6. Caso TODO LIBRO



##### Se pide

1. Realizar la creación de las tablas correspondientes, para cada tabla deberá definirse la clave primaria y las claves foráneas que considere necesarias.



### L.7. Caso CHICHO PETS



#### Se pide

1. Realizar la creación de las tablas correspondientes, para cada tabla deberá definirse la clave primaria y las claves foráneas que considere necesarias.



#### L.8. Caso E-BOLSA



##### Se pide

1. Realizar la creación de las tablas definiendo para cada tabla la clave primaria, las claves foráneas, las restricciones de obligatoriedad y de dominio que considere necesarias.



### L.9. Caso SOLARIUM TOSTADITOS



#### Se pide

1. Realizar la creación de las tablas definiendo para cada tabla la clave primaria, las claves foráneas, las restricciones de obligatoriedad y de dominio que considere necesarias.

**M**

**Resolución de Casos Prácticos  
de DDL**

---



**Importante**

Recuerde que un mismo ejercicio puede admitir en algún punto distintas soluciones a un mismo requerimiento, por lo que a continuación se presenta una solución propuesta para cada ejercicio, pudiendo alguna de ellas diferir en algún grado de la solución desarrollada por usted, ante esto resulta de vital importancia asegurarse de que lo diseñado por usted sea compatible con lo expresado en el relevamiento, no se olvide que el último paso de un buen diseño es testear que el modelo diseñado sea compatible con la realidad relevada.

**M.1. Caso CUKI S.R.L**

```
CREATE TABLE PROVINCIA
(
ID_PCIA INT PRIMARY KEY,
NOMBRE CHAR(40) NOT NULL
)
```

```
CREATE TABLE LOCALIDAD
(
ID_PCIA INT,
ID_LOCAL INT,
NOMBRE CHAR(40) NOT NULL,
CONSTRAINT PK_LOCALIDAD PRIMARY KEY (ID_PCIA, ID_LOCAL),
CONSTRAINT FK_LOCALIDAD_PCIA FOREIGN KEY (ID_PCIA)
REFERENCES PROVINCIA (ID_PCIA)
)
```

```
CREATE TABLE SUCURSAL
(
ID_SUCURSAL INT PRIMARY KEY,
NOMBRE CHAR(40) NOT NULL,
ID_PCIA INT NOT NULL,
ID_LOCAL INT NOT NULL,
CALLE CHAR(50) NOT NULL,
NRO INT NOT NULL,
ID_ZONA INT NOT NULL,
CONSTRAINT FK_SUCURSAL_LOCAL FOREIGN KEY (ID_PCIA, ID_LOCAL)
REFERENCES LOCALIDAD (ID_PCIA, ID_LOCAL),
CONSTRAINT FK_SUCURSAL_ZONA FOREIGN KEY (ID_ZONA)
REFERENCES ZONA (ID_ZONA)
)
```

```
CREATE TABLE FABRICA
(
ID_FABRICA INT PRIMARY KEY,
NOMBRE CHAR(40) NOT NULL,
ID_PCIA INT NOT NULL,
ID_LOCAL INT NOT NULL,
CALLE CHAR(40) NOT NULL,
NRO INT NOT NULL,
```

```
CONSTRAINT FK_FABRICA_LOCAL FOREIGN KEY (ID_PCIA, ID_LOCA)
REFERENCES LOCALIDAD (ID_PCIA, ID_LOCAL)
)

CREATE TABLE GRUPO
(
ID_GRUPO INT PRIMARY KEY,
NOMBRE CHAR(40) NOT NULL
)

CREATE TABLE TIPO
(
ID_GRUPO INT,
ID_TIPO INT,
NOMBRE CHAR(40),

CONSTRAINT PK_TIPO PRIMARY KEY (ID_GRUPO, ID_TIPO),

CONSTRAINT FK_TIPO_GRUPO FOREIGN KEY (ID_GRUPO)
REFERENCES GRUPO (ID_GRUPO)
)

CREATE TABLE PRODUCTO
(
ID_PRODUCTO INT PRIMARY KEY,
NOMBRE CHAR(40) NOT NULL,
ID_GRUPO INT NOT NULL ,
ID_TIPO INT NOT NULL ,

CONSTRAINT FK_PRODUCTO TIPO FOREIGN KEY (ID_GRUPO, ID_TIPO)
REFERENCES TIPO (ID_GRUPO, ID_TIPO)
)

CREATE TABLE STOCK_PRODUCTO_FABRICA
(
ID_PRODUCTO INT,
ID_FABRICA INT,
FECHA DATETIME,
CANTIDAD NUMERIC(15,2),

CONSTRAINT PK_STK_PROD_FAB PRIMARY KEY (ID_PRODUCTO, ID_FABRICA, FECHA),

CONSTRAINT FK_STK_PROD_FAB_PROD FOREIGN KEY (ID_PRODUCTO)
REFERENCES PRODUCTO (ID_PRODUCTO),

CONSTRAINT FK_STK_PROD_FAB_FABRICA FOREIGN KEY (ID_FABRICA)
REFERENCES FABRICA (ID_FABRICA),

CONSTRAINT CK_STK_PROD_FAB_FECHA CHECK (FECHA >= GETDATE()),

CONSTRAINT CK_STK_PROD_FAB_CANT CHECK (CANTIDAD > 0)
)
```



## M.2. Caso TERRA CONTROL

```
CREATE TABLE PROVINCIA
(
ID_PCIA INT PRIMARY KEY,
NOMBRE CHAR(40) NOT NULL
)
```

```
CREATE TABLE LOCALIDAD
(
ID_PCIA INT ,
ID_LOCAL INT,
NOMBRE CHAR(40) NOT NULL,
CONSTRAINT PK_LOCALIDAD PRIMARY KEY (ID_PCIA, ID_LOCAL),
CONSTRAINT FK_LOCAL_PCIA FOREIGN KEY (ID_PCIA)
REFERENCES PROVINCIA (ID_PCIA)
)
```

```
CREATE TABLE CLIENTE
(
DNI      INT PRIMARY KEY,
MAIL     CHAR(30),
ID_PCIA  INT NOT NULL,
ID_LOCAL  INT NOT NULL,
CALLE    CHAR(40) NOT NULL,
NUMERO   INT NOT NULL,
PISO     INT ,
DEPARTAMENTO CHAR(3),
PASSWORD  CHAR(6) NOT NULL ,
CONSTRAINT FK_CLIENTE_LOCAL FOREIGN KEY (ID_PCIA, ID_LOCAL)
REFERENCES LOCALIDAD (ID_PCIA, ID_LOCAL),
CONSTRAINT CK_CLIENTE_NRO CHECK (NUMERO > 0),
CONSTRAINT CK_CLIENTE_PISO CHECK (PISO > 0),
CONSTRAINT CK_CLIENTE_PASSWORD CHECK (LEN(PASSWORD) = 6)
)
```

```
CREATE TABLE CLIENTE_TELEFONO (
DNI      INT,
TELEFONO INT,
```

```
CONSTRAINT PK_CLIENTE_TELEFONO PRIMARY KEY (DNI, TELEFONO),
```

```
CONSTRAINT FK_CLIENTE_TEL_DNI FOREIGN KEY (DNI)
REFERENCES CLIENTE (DNI)
)
```

```
CREATE TABLE SERVICIO
(
ID_SERVICIO          INT PRIMARY KEY,
DESCRIPCION          CHAR(40) NOT NULL,
CANTIDAD_DIAS_TRABAJO INT NOT NULL
)
```

```
CREATE TABLE PRECIO_SERVICIO
(
ID_SERVICIO          INT,
FECHA                DATETIME,
PRECIO NUMERIC(15,2) NOT NULL ,
```

```
CONSTRAINT PK_PRECIO_SERVICIO PRIMARY KEY (ID_SERVICIO, FECHA),
CONSTRAINT FK_PRECIO_SERV_SERVICIO FOREIGN KEY (ID_SERVICIO)
REFERENCES SERVICIO (ID_SERVICIO)
)
```

```
CREATE TABLE MARCA
(
ID_MARCA INT PRIMARY KEY,
NOMBRE   CHAR(40) NOT NULL
)
```

```
CREATE TABLE MODELO
(
ID_MARCA  INT ,
ID_MODELO INT ,
NOMBRE    CHAR(40) NOT NULL ,
```

```
CONSTRAINT PK_MODELO PRIMARY KEY (ID_MARCA, ID_MODELO),
CONSTRAINT FK_MODELO_MARCA FOREIGN KEY (ID_MARCA)
REFERENCES MARCA (ID_MARCA)
)
```

```
CREATE TABLE AÑO
(
ID_AÑO INT PRIMARY KEY
)
```

```
CREATE TABLE MODELO_AÑO
(
ID_MARCA INT,
ID_MODELO INT,
ID_AÑO INT,
CONSTRAINT PK_MODELO_AÑO PRIMARY KEY (ID_MARCA, ID_MODELO, ID_AÑO),
CONSTRAINT FK_MODELO_AÑO_MOD FOREIGN KEY (ID_MARCA, ID_MODELO) REFERENCES MODELO
(ID_MARCA, ID_MODELO),
CONSTRAINT FK_MODELO_AÑO_AÑO FOREIGN KEY (ID_AÑO)
REFERENCES AÑO (ID_AÑO)
)

CREATE TABLE CABECERA_PRESUPUESTO
(
ID_PRESUPUESTO INT PRIMARY KEY,
FECHA DATETIME NOT NULL,
DNI INT NOT NULL,
ID_MARCA INT NOT NULL,
ID_MODELO INT NOT NULL,
AÑO INT NOT NULL,
CONSTRAINT FK_CAB_PRESUP_DNI FOREIGN KEY (DNI)
REFERENCES CLIENTE (DNI),
CONSTRAINT FK_CAB_PRESUP_MARCA_MODELO FOREIGN KEY (ID_MARCA, ID_MODELO)
REFERENCES MARCA_MODELO (ID_MARCA, ID_MODELO)
)

CREATE TABLE RENGLON_PRESUPUESTO
(
ID_PRESUPUESTO INT,
ID_SERVICIO INT,
CONSTRAINT PK_RENG_PRESUP PRIMARY KEY (ID_PRESUPUESTO, ID_SERVICIO),
CONSTRAINT FK_RENG_PRESUP_SERV FOREIGN KEY (ID_SERVICIO)
REFERENCES SERVICIO (ID_SERVICIO)
)
```



# N

## Uso de Herramientas

---

### Objetivos

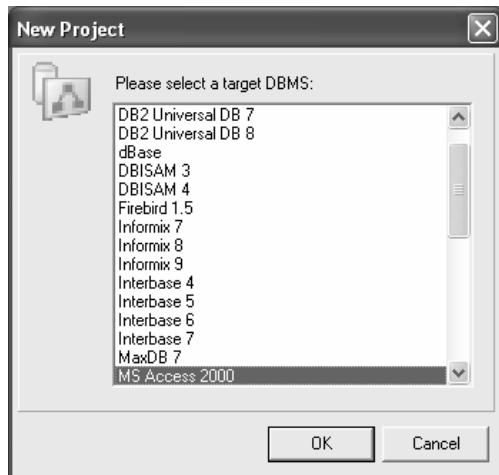
- ✓ Conocer herramientas que tienen presencia en el mercado.
- ✓ Armar sus propios diagramas.
- ✓ Lograr armar modelos tanto lógicos como físicos.
- ✓ Familiarizarse con algunas de las funcionalidades que ofrecen las herramientas de modelado de datos.
- ✓ Mostrar dos herramientas: DeZign y DBDesigner.

## DeZign

### Definir un nuevo proyecto

Lo primero que se debe hacer es crear un nuevo proyecto.

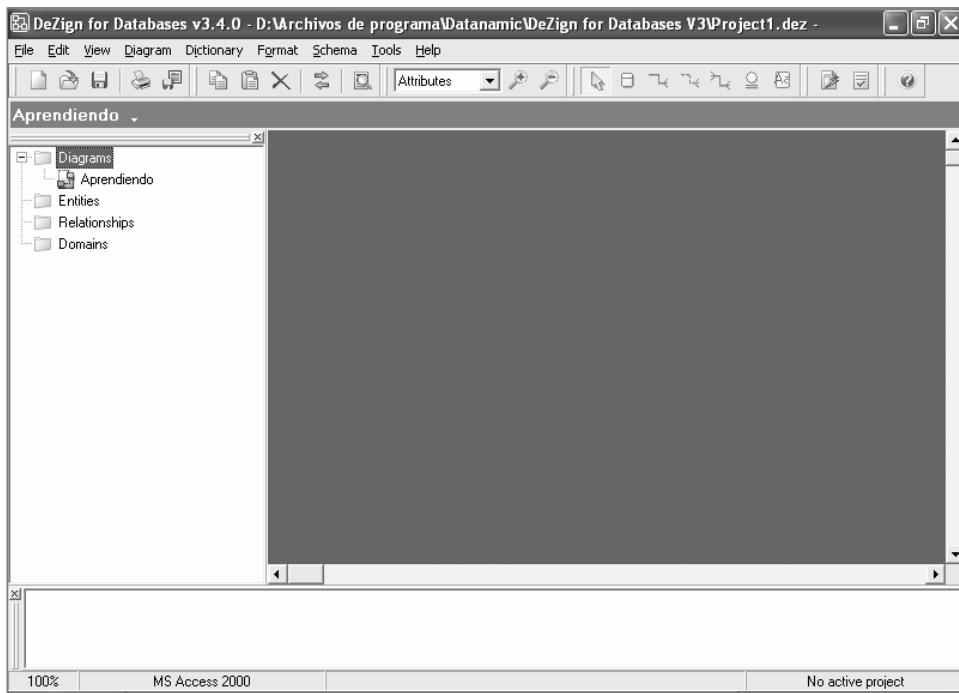
Dado que la mayoría de las herramientas de modelado de datos trabaja contra una base de datos determinada se deberá especificar cual es la misma., para lo cual previamente crearemos una base de datos vacía a la altura del c:\Aprendiendo.mdb, llamada Aprendiendo. Esto lo que permite es impactar directamente en la base de datos cualquier cambio que se realice en la herramienta, como así también a través de ingeniería inversa recuperar todos los metadatos de la base de datos, es decir entidades, relaciones, tipos de datos, etc.



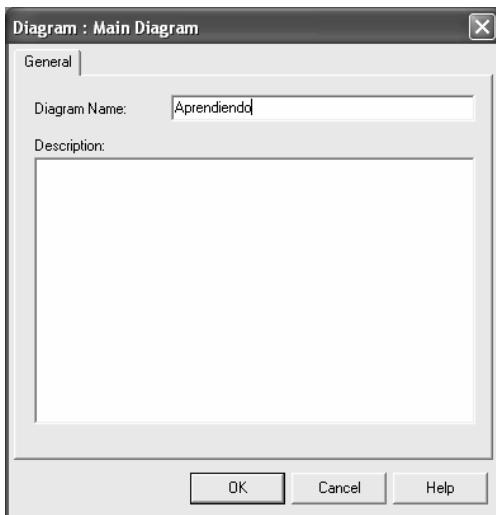
El siguiente paso consiste en definir el nombre del diagrama. Esto lo haremos en la parte del menú que dice Diagramas.

### Definiendo el Diagrama

La pantalla principal de fondo, a la derecha es el área de trabajo. A la izquierda se puede observar un navegador que permite recorrer los diagramas, entidades, relaciones y dominios.



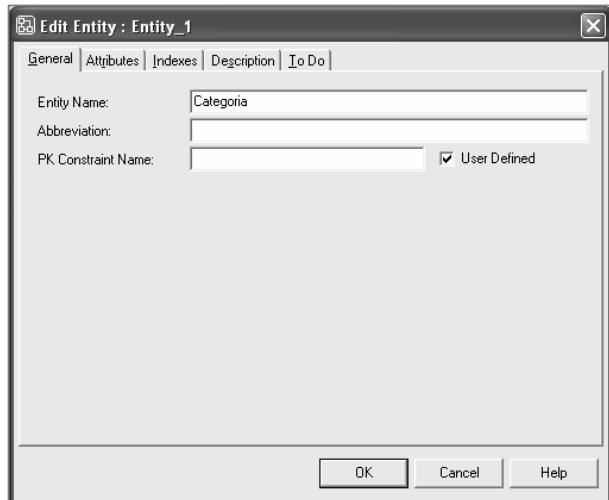
Lo primero que debemos hacer es poner un nombre al diagrama sobre el cual debemos trabajar.



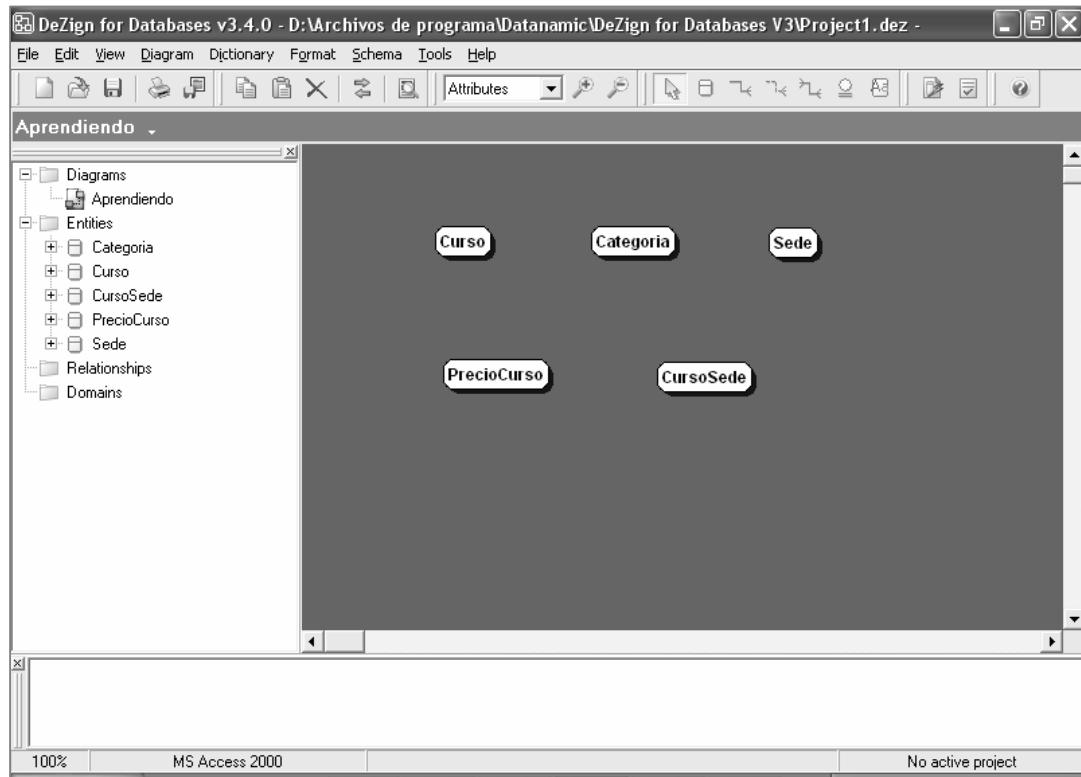
Sería bueno agregar un detalle acerca de que es lo que resuelve el Diagrama. Esto puede hacerse en el recuadro blanco llamado Description.

### Creando las entidades

Para que el modelo cobre vida se deben agregar entidades, las tablas del mismo. De esta forma debajo de la carpeta Entities, con el botón derecho del mouse procederemos a crear una nueva entidad, y especificamos su nombre en donde dice Entity Name:



De la misma forma crearemos el resto de las entidades, a saber Curso, Sede, PrecioCurso, CursoSede, quedándonos unas cinco entidades que pueden ser vistas desde el panel izquierdo de navegación, bajo el nombre de Entities:



### Definiendo las relaciones

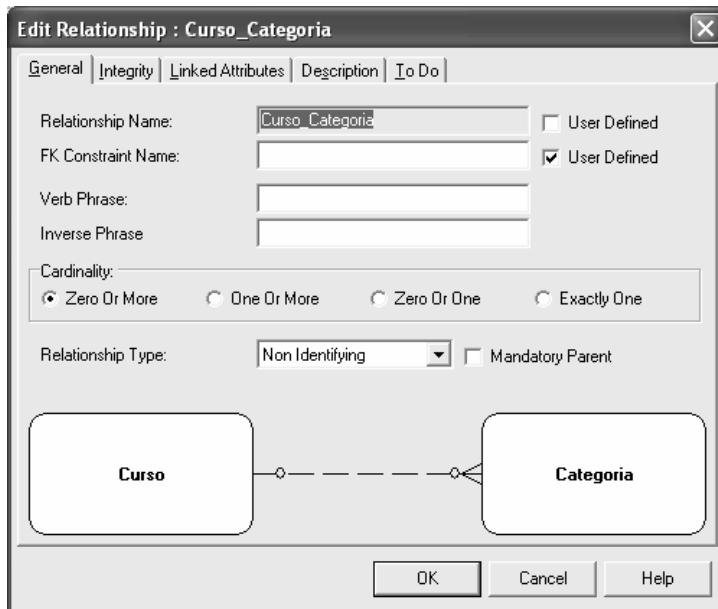
Se pueden agregar las relaciones entre cada una de ellas. En la parte superior derecha de la herramienta podemos ver los tipos de relaciones que se pueden utilizar.



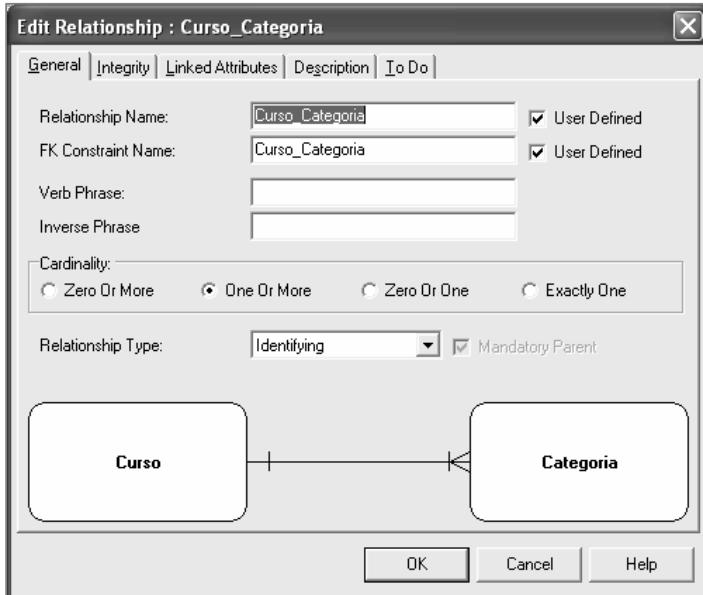
Otra forma de establecer una relación entre dos entidades es desde Dictionary – New Relationship. Debemos especificar entre qué entidades se va a definir esta nueva relación, supongamos que lo hacemos entre Curso y Categoría.



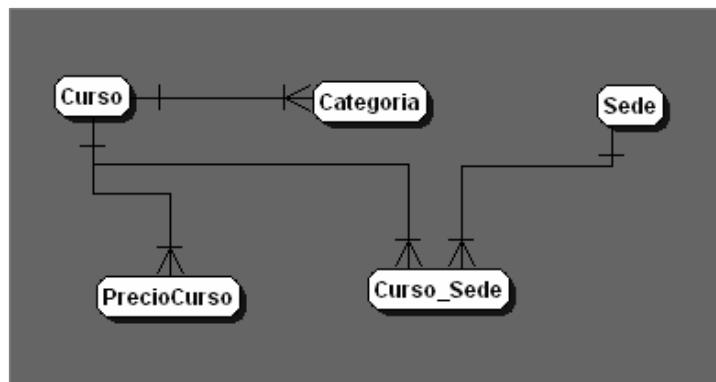
Haciendo Click en Create aparece un cuadro de diálogo con varias opciones a configurar:



Debemos especificar el tipo de relación que se pretende modelar, eligiendo entre otras cosas la cardinalidad de la misma (en el ejemplo anterior se gráfica cero a muchas):

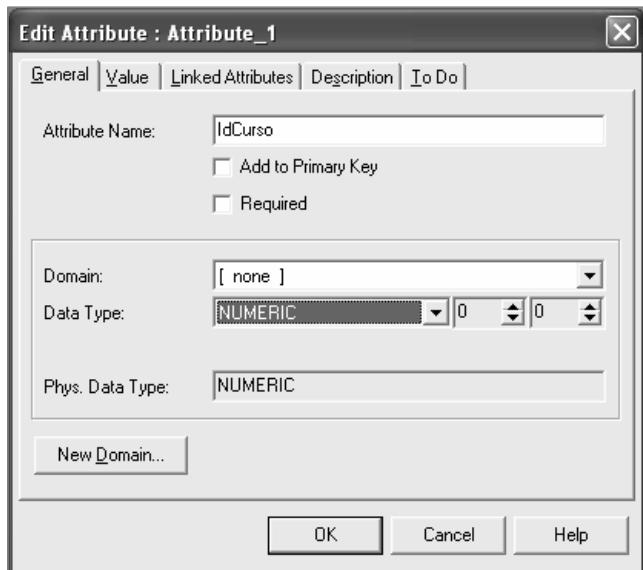


En este ejemplo elegimos el tipo de cardinalidad uno a muchas. De esta forma se pueden ir agregando todas las relaciones, y cambiándolas desde el menú mencionado anteriormente. Este sería un modelo de cómo iría quedando parte del DER del caso Aprendiendo.



### Agregando Atributos

Podemos bajar más aun de nivel de detalle y especificar no sólo los nombres de las entidades sino sus atributos y tipos de datos. Si queremos hacerlo se puede hacer doble click sobre la entidad y en la pestaña llamada Atributtes se pueden ir definiendo uno a uno cada uno de los atributos:



### Definiendo las propiedades de los atributos

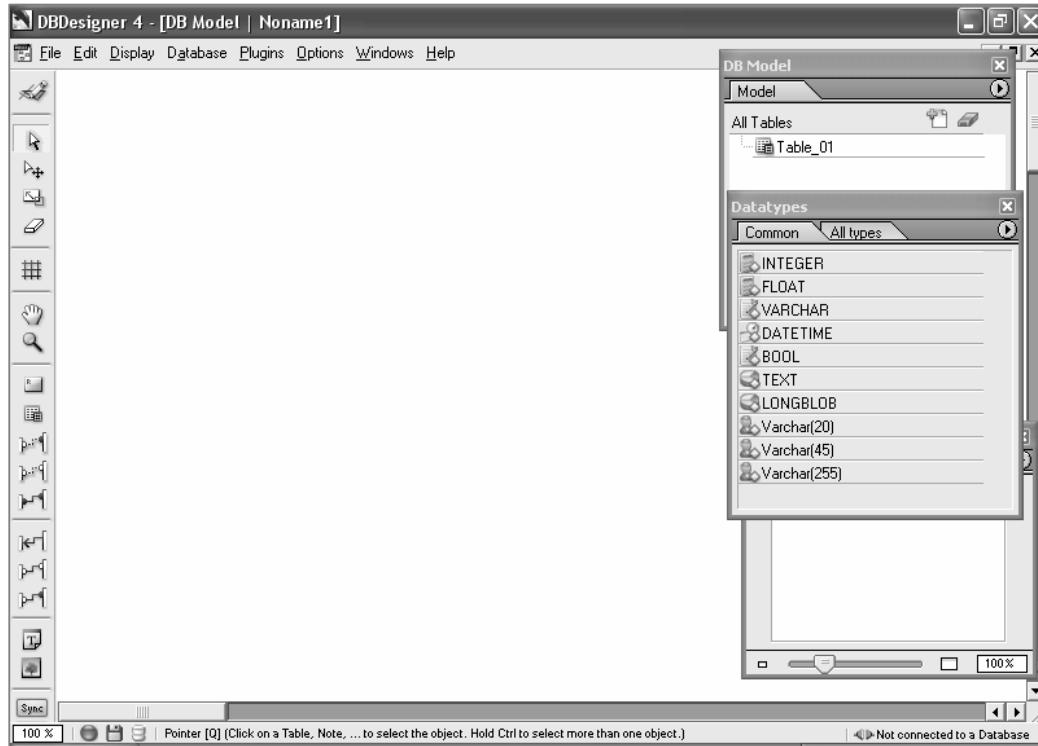
En la ventana de definición de atributos mostrada arriba se pueden observar que cosas podemos definir de los atributos: su nombre, si es parte o no de la clave primaria, si es un atributo requerido o no (admite o no ceros), el tipo de datos, longitud, precisión, etc. De esta forma si especificamos todos los atributos nos quedaría un menú como el siguiente, en donde pueden cambiarse el orden mostrado de los mismos con las flechitas laterales:

Key	Attribute Name	Datatype	Length	Precision	Domain
	IdCurso	NUMERIC	0	0	
	Nombre	TEXT	40	0	
	Duracion	TEXT	30	0	
	Categoría	TEXT	30	0	
	Precio	MONEY	0	0	

## DBDesigner

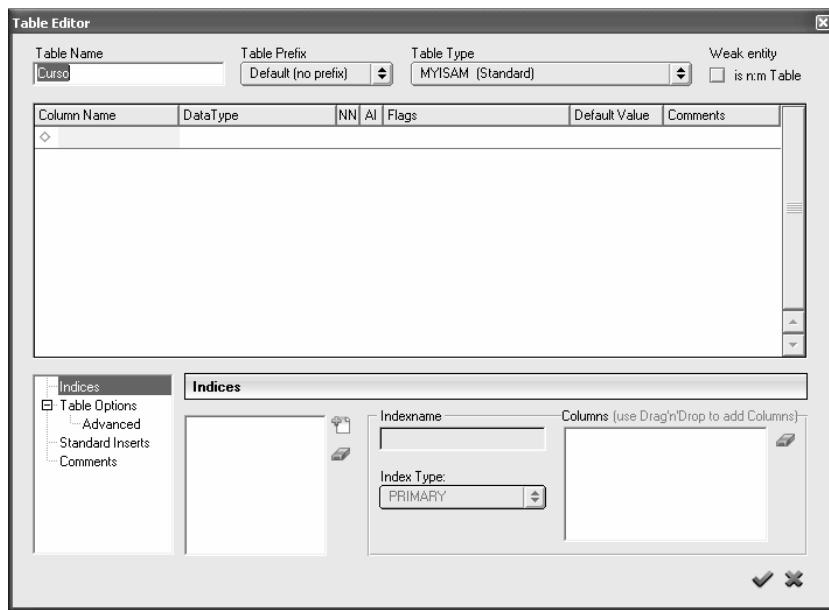
### Creando un Diagrama

Para crear un nuevo diagrama con esta herramienta iremos a File - New

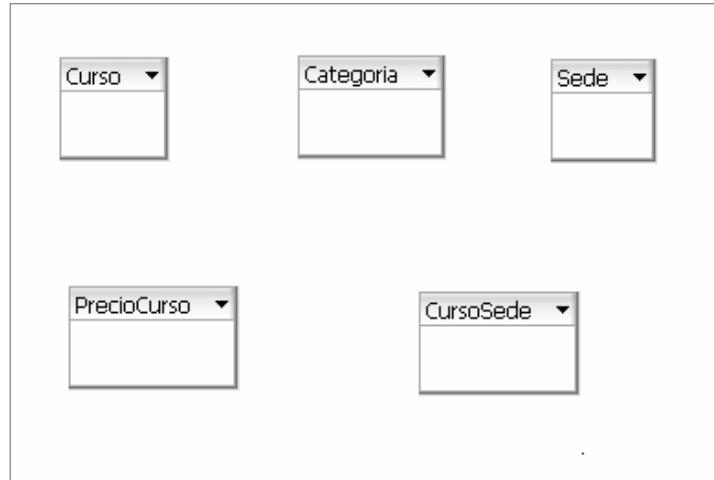


### Definiendo una nueva entidad

Sobre el panel izquierdo hacemos click a New Table arrastrando y soltando sobre la parte blanca de fondeo de la herramienta aparece por default una tabla cuyo nombre es **Table\_01**. Haciendo doble click sobre la misma se despliega una ventana que nos permite cambiar su nombre. Pondremos a nuestra primer tabla del modelo el nombre de **Curso**.



Luego vamos agregando una a una las tablas del modelo, de manera tal de lograr un esquema similar al que se muestra:

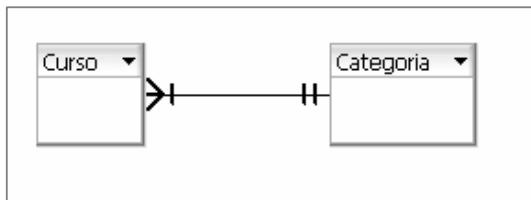


#### Definiendo una relación entre dos entidades

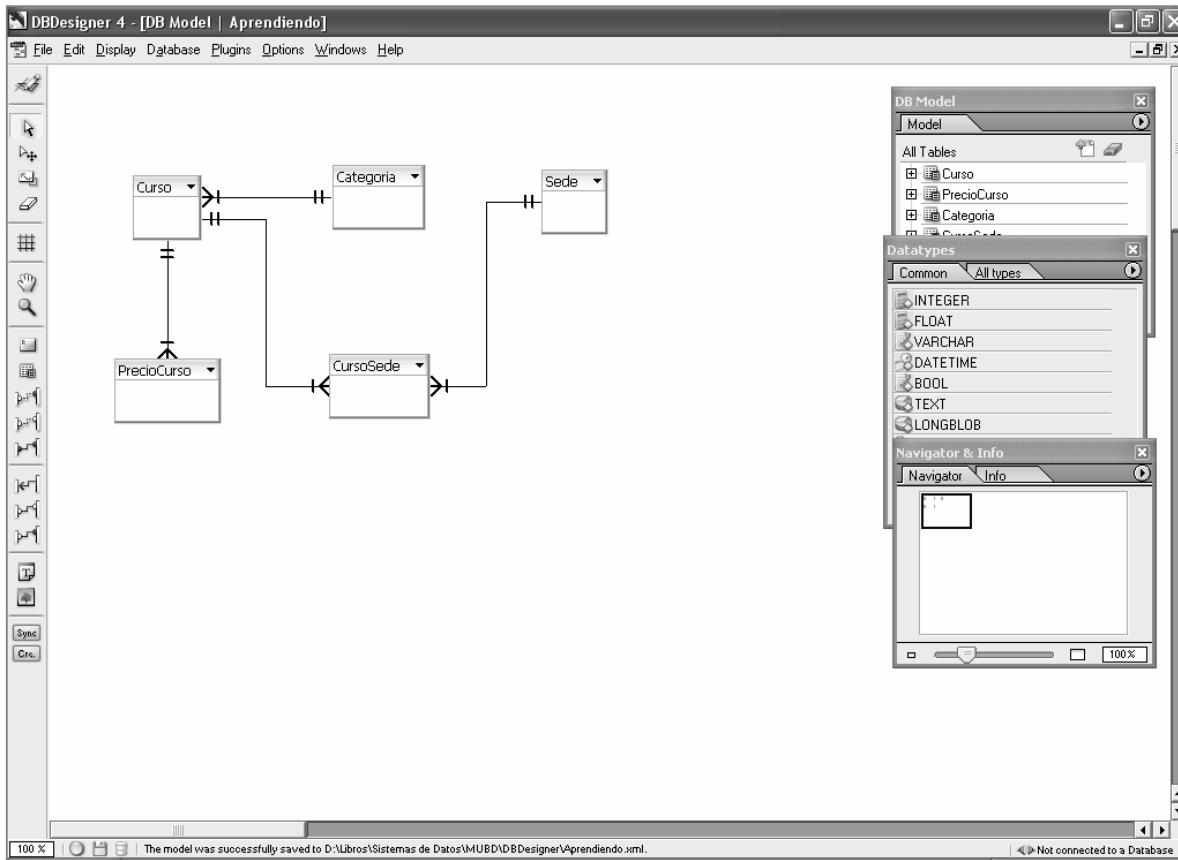
Sobre el panel izquierdo tenemos varios menús de herramientas, que nos muestran los tipos de relaciones que se pueden utilizar, para especificar cual es la relación que se quiere modelar entre cada una de las entidades. Por ejemplo para especificar el tipo de relación muchos a uno entre Curso y Categoria elegimos la primer flecha del siguiente menú:



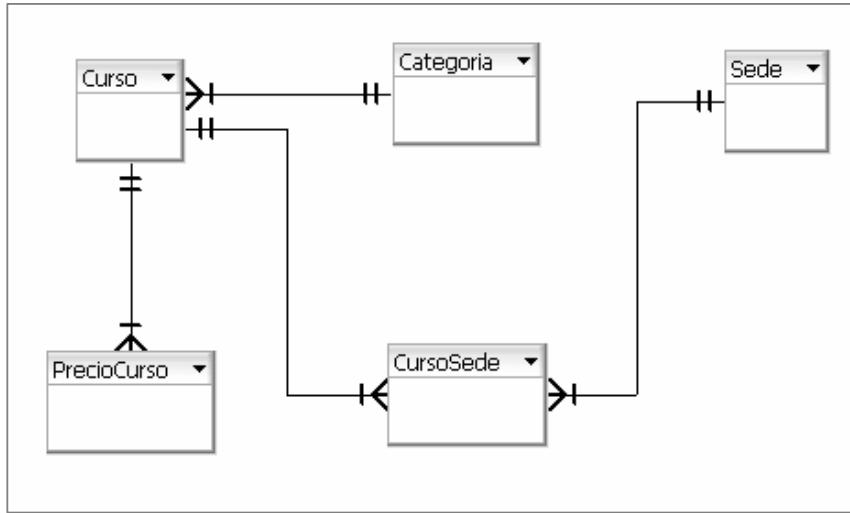
Para lograrla haremos click primero en el tipo de relación (primer flecha), del menú mostrado anteriormente, y luego hacemos click en Categoría y luego en Curso. De esta forma queda una relación graficada como la que se muestra a continuación.



De igual forma procederemos con el resto de las entidades, para especificar los distintos tipos de relaciones que mantienen entre ellas.



De esta forma nos quedará un diagrama un poco mejor definido:



### Agregando atributos a nuestras entidades, y definiendo sus propiedades

Si queremos refinar en grado de detalle y definir el modelo físico debemos comenzar a agregar los atributos a las entidades, lo haremos de la siguiente forma:

Column Name	DataType	NN	AI	Flags	Default Value	Comments
idCurso	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
Nombre	CHAR			<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
Descripcion	DOUBLE PRECISION			<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
Duracion						

Indices

Table Options

Advanced

Standard Inserts

Comments

Indices

PRIMARY

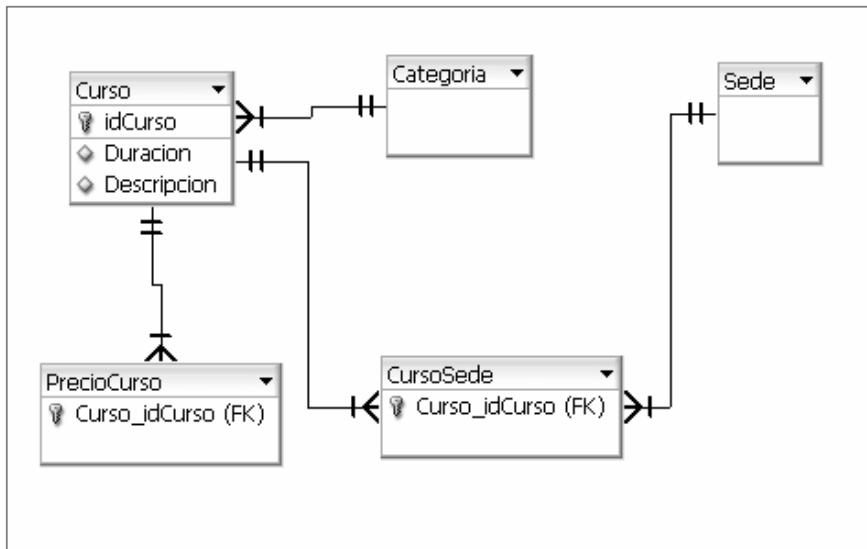
Indexname: PRIMARY

Columns (use Drag'n'Drop to add Columns): idCurso

Index Type: PRIMARY

✓ ✕

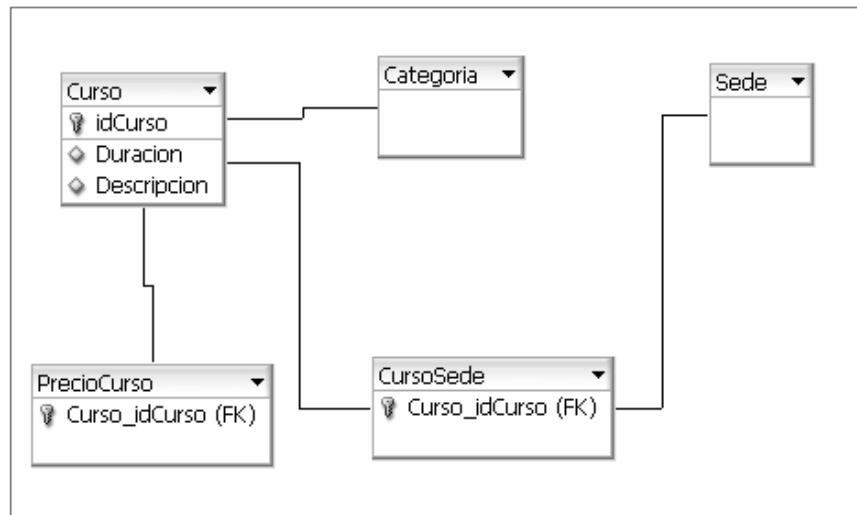
Hay varias propiedades para setear tales como el nombre del atributo, si es o no clave, el tipo de datos, si acepta valores nulos o no, etc.



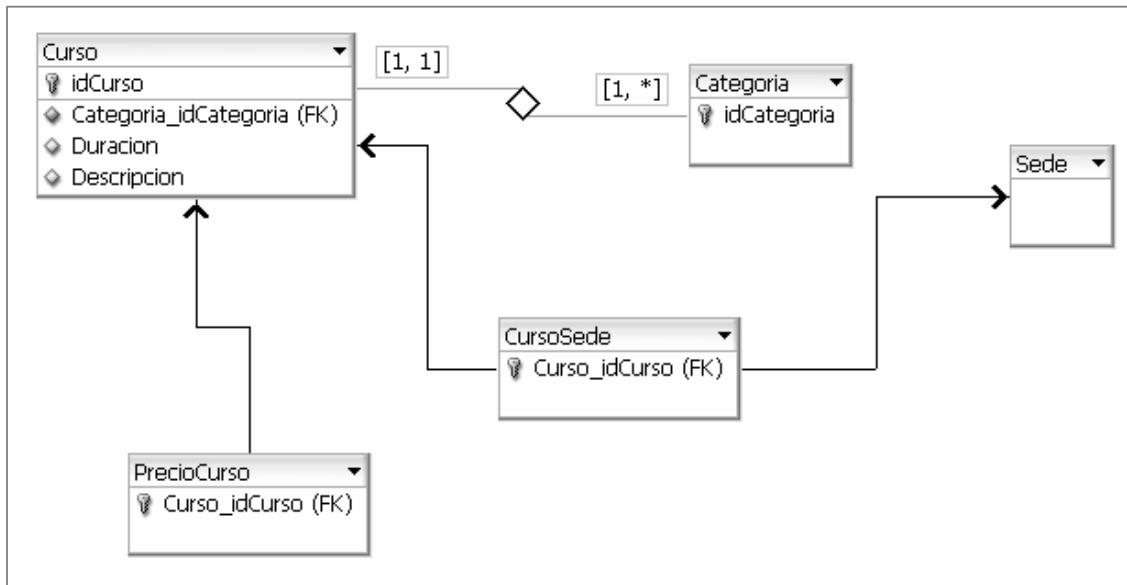
#### Cambiando la notación de las relaciones

En el ejemplo anterior agregamos tres atributos al modelo. Otra funcionalidad notable de estas herramientas es que se pueden cambiar los tipos de notaciones utilizadas para representar un modelo de datos. Así por ejemplo si se desea utilizar otro tipo de notación y no pata de Gallo, podríamos elegirla desde Display – Notation – Traditional, o bien EER, o bien EER (1,n)

#### Notación Tradicional



Por ejemplo si decidiéramos ver la cardinalidad en las relaciones se podría usar la última del modelo que es EER (1,n) y quedaría algo así:

**Notación EER (1,n)**

Esta última se parece más a un diseño de clases, tal como lucen las aplicaciones hoy en día.





## Casos Adicionales

---

### Objetivos

- ✓ Profundizar temas vistos a lo largo de la obra.
- ✓ Ampliar la cantidad de casos presentados.
- ✓ Incentivar la resolución de casos por cuenta propia.
- ✓ Ampliar el espectro de casos y situaciones tratados.



### O.1. Caso SALIENDO DEL PASO

Una empresa de tecnología comercializa productos electrónicos. Lamentablemente José Cuervo, no ha cursado ninguna materia de sistemas y elaboró con un sistema de gestión de base de datos del tipo Opensource la siguiente tabla. La misma pretende registrar las ventas de los productos por la empresa. Leyendo un artículo en una revista técnica José leyó algo de formas normales.

	<b>Productos</b>
#	FechaFactura
#	CodCliente
	RazónSocialCliente
	TeléfonoCliente
	Teléfono2Cliente
	Teléfono3Cliente
#	NroFactura
	FechaNacimientoHijoCliente
	FechaNacimientoHijo2Cliente
	AniosHijoCliente
	AniosHijo2Cliente
#	CodProducto
	DescripcionProducto
	Cantidad
	PrecioUnitario
	CodigoMoneda
	Cotizacion
	StockActualDelProducto



Preste atención que los # forman la clave de la tabla.



**Se pide**

1. Que ayude a José Cuervo y normalice la siguiente tabla hasta la tercera forma normal, eliminando la redundancia posible.



## O.2. Caso WEBSHOPING

La empresa WEBSHOPING posee un sistema el cuál es accedido desde la Web. Dado que fue desarrollándose de a poco, el mismo no se sabe si se encuentra normalizado.

El siguiente modelo de datos pretende almacenar la información respecto a:

- ✓ Datos personales de los Usuarios de un sitio Internet.
- ✓ Historia de accesos discriminando a qué lugares del sitio accedió.
- ✓ Historia de las compras realizada por cada usuario.

<b>Usuarios</b>	
Id_usu	
Nombre	
Apellido	
Telefono Particular	
Telefono Movil	
Fax	
Nombre Hijo 1º	
Nombre Hijo 2º	
Nombre Esposa	

<b>Compras</b>	
Id_usu	
Fecha y Hora de Compra	
Nombre de Producto	
Cantidad	
Precio Unitario del Producto	

<b>Accesos</b>	
Id_usu	
Fecha y Hora de Acceso	
Nombre de la página accedida	



### Se pide

1. Decidir si el modelo se halla en 3º forma normal. En caso de no estarlo, corregirlo para que se encuentre en 3FN sin pérdida de información.



### O.3. Caso ELECTRO VENTAS

La empresa ELECTRO VENTAS comercializa productos electrónicos. La misma quiere registrar los pedidos que realizan los clientes y poder decirles si puede satisfacerlos o no.

Esta última decisión la tomará en base a los pedidos existentes sobre un determinado producto y al stock actual que posee, siguiendo la siguiente fórmula:

$$\text{Stock_Disponible} = \text{Stock_Actual} - \text{Cant_Pedida}$$



La Cant\_Pedida aún no fue entregada.



#### Se pide

1. Realizar un modelo de datos expresado en tercera forma normal que permita registrar los pedidos realizados por un cliente y si existe o no stock de un determinado producto para satisfacer el pedido.



#### O.4. Caso LA TOALLITA LIMPIA

Una compañía de limpieza, para los efectos de realizar la facturación a sus clientes, necesita registrar la cantidad de horas que estuvo trabajando la gente y los productos que fueron utilizados por dicho personal.

Los principales datos son los siguientes:

- ✓ Nombre y apellido del empleado.
- ✓ Costo por hora del empleado.
- ✓ El conjunto de productos que se utilizan para realizar las tareas de limpieza junto con el costo asociado al mismo.
- ✓ Registro de la cantidad de horas que trabajó el personal en cada uno de los clientes y los productos que utilizó para realizar las tareas.



##### Se pide

1. Basado en la información anterior se pide diseñar un modelo de datos relacional que se encuentre en Tercera forma normal: Especificar nombre de las tablas, clave primaria y campos principales.



## 0.5. Caso DON PEINE

En la peluquería DON PEINE utilizan un sistema informático realizado con tecnología de objetos, cuya pantalla muestra lo siguiente:

**Peluquería Don Peine**

<b>Datos Orden Servicio:</b>	<b>Forma de Pago:</b>
Número de Orden: 0012789	Local: Centro
Fecha: 04/12/2005	Hora: 12:10 a.m.
Peluquero: Matías	Nombre Cliente: Jacinto
	Paga con: <input checked="" type="radio"/> Tarjeta <input type="radio"/> Efectivo
	Tarjeta Banco: <input type="radio"/> Banco Río <input checked="" type="radio"/> Banco Francés
	Importe: 35.00
<b>Servicio/s prestados:</b>	
Cantidad: 1	Servicio: Corte de Pelo
	Precio: 35

**Peluquería Don Peine**  
Agradece su visita.



Preste atención a que se pueden agregar n ítems en la orden de servicio mostrada por pantalla.



### Se pide

- Como Ud. de objetos no entiende mucho, y hay 20 personas antes que le toque su turno, es que se le pide realizar el diseño lógico de la aplicación expresado en 3FN.
- Dado que la demora es más larga de lo que parecía Ud. sigue imaginándose como sería el diagrama lógico y logra esbozar el mismo.

**O.6. Caso VIAJE VELOZ**

La empresa Viaje Veloz ha puesto a su disposición un nuevo sistema para mejorar el control de sus viajeros. Para ello utiliza una tarjeta magnética, la cual debe ser pasada por un lector para poder ingresar a la estación. Existen distintos tipos de tarjetas, cada una tiene un costo diferente, con una cierta cantidad de viajes. El sistema tiene que poder determinar la cantidad de viajes realizados en la línea de forma diaria.

**Se pide**

1. Realizar el DER correspondiente.
2. Realizar el pasaje del DER a un modelo de datos expresado en tercera forma normal.



## O.7. Caso SU PEDIDO

La empresa de Desarrollos a su Gusto cuenta con un Sistema Informático para administrar los requerimientos (Pedidos) que realizan sus Clientes. Este sistema permite almacenar la siguiente información:

- ✓ Nro. de Requerimiento.
- ✓ Fecha en que se dio de alta.
- ✓ Prioridad.
- ✓ Cliente que realizó el pedido .
- ✓ Tareas asignadas a cada uno de los requerimientos y el empleado que lo realizó.
- ✓ Cantidad de horas que se le dedicó a los anteriores trabajos.
- ✓ Fecha en la que se finalizó el requerimiento.
- ✓ Estado del Requerimiento.

### Aclaraciones

- ✓ Un requerimiento puede ser satisfecho por varias tareas. Dichas tareas pueden haber sido realizadas en distintos días.
- ✓ Un requerimiento se inicia en el estado "Pendiente". Cuando un empleado realiza alguna tarea asociada al mismo, se pasa al estado "En curso" y cuando el empleado considera que se ha completado el requerimiento, el mismo pasa a estado "Finalizado".



#### Se pide

1. Realizar el DER correspondiente a lo expresado en el relevamiento.
2. Realizar el pasaje del DER a un modelo de datos normalizado expresado en tercera forma normal.



## O.8. Caso VAMOS A LA FERIA

El Sistema se inicia con la registración de una persona para el ingreso a la feria. Luego de esta registración se le entrega a cada participante una tarjeta con un código de barra que lo identificará durante su estadía dentro de la feria y por el tiempo en que esta se extienda.

Dentro de la feria, cada stand, mediante un lápiz óptico, toma registro de la persona que está interesada en algún producto/servicio que el stand promociona. Una vez que el interesado deja el local, la persona que lo atendió marca con una cruz los productos/servicios en los cuales se encontraba interesado.

Al finalizar la feria se le envía a cada empresa participante, el detalle de los datos recabados para su empresa. Por otra parte, el resumen general sobre la actividad de los participantes se publica en un boletín de la feria y en su sitio de Internet.



### Se pide

1. Realizar el Diagrama de Entidad – Relación.
2. Realizar un modelo de datos en tercera forma normal que cumpla con todos los puntos que figuran en el relevamiento.
3. Realizar el Diagrama lógico correspondiente al modelo normalizado.
4. Realizar la creación de todas las tablas del modelo normalizado como así también de todas las restricciones que considere necesarias para garantizar la consistencia de los datos.



### O.9. Caso EL PUBLICISTA

La empresa El Publicista se dedica a la colocación de avisos de selección de personal en distintos medios gráficos. La misma tiene como clientes a las consultoras de RRHH, que le solicitan la publicación de los mismos. Cuando uno de sus clientes está interesado en publicar un aviso para seleccionar determinado perfil envía lo siguientes datos a la empresa: Fecha del aviso, perfil buscado, Código de Referencia, Remuneración ofrecida, Mail de Contacto, Tareas a Realizar, y Plazo de Contratación.

Por cada aviso publicado la empresa cobra una tarifa, dependiendo del medio.

Las empresas pueden publicar el aviso en distintos medios, tales como TV, diarios, revistas. Los datos de los medios son, Razón Social, Dirección, Teléfonos y Mails de contacto.

El Publicista desea saber qué cantidad de avisos publica por mes, en qué medio, y además de cuánto le cobra cada uno de ellos.

Por parte de sus clientes debe registrar la cantidad de avisos publicados entre dos fechas a determinar. Los datos relevantes de los mismos son Nombre, Apellido, o Razón Social, Dirección, Teléfonos y Mails de contacto.



#### Se pide

1. Realizar el Diagrama de Entidad – Relación.
2. Realizar un modelo de datos en tercera forma normal que cumpla con todos los puntos que figuran en el relevamiento.
3. Realizar el Diagrama lógico correspondiente al modelo normalizado.
4. Realizar la creación de todas las tablas del modelo normalizado como así también de todas las restricciones que considere necesarias incluir para garantizar la consistencia de los datos.



# P

## Preguntas Frecuentes

---

### Objetivos

- ✓ Aclarar dudas respecto de los temas tratados.
- ✓ Responder a las preguntas más usualmente formuladas.
- ✓ Advertir sobre posibles equivocaciones.
- ✓ Profundizar en temas de cierta complejidad.
- ✓ Dar opiniones o puntos de vistas propios respecto de problemas particulares.

## P. Preguntas frecuentes

### 1. ¿Cuántos tipos de claves primarias existen?

Dos, a saber:

- a. Atómica: Cuando está formada por un único atributo.
- b. Compuesta: Cuando está formada por dos o más atributos.

### 2. ¿Cuál es la cantidad máximas de atributos que pueden formar parte de una clave primaria o foránea?

No hay restricción sintáctica, sino que dependerá de la situación que se está modelando.

### 3. ¿Las claves primarias puede ser cualquier atributo o conjunto de atributos?

La clave primaria de una tabla es el campo o conjunto de campos que hacen único un registro dentro de una tabla, es la situación que se está modelando la que define que campo o conjunto de campos forman parte de la clave primaria.

Un error muy común es el de definir claves primarias que no satisfacen lo solicitado por la realidad que se intenta representar mediante la tabla creada, por lo que siempre es necesario validar que la clave cumple con la condición de unicidad planteada por la situación modelada.

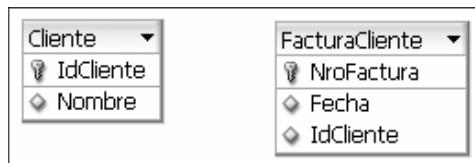
### 4. ¿Es coherente que existan dos tablas que tengan como clave primaria los mismos atributos?

En un modelo normalizado puro esto sería generar redundancia innecesaria, ya que podría juntar ambas en una única tabla. Aunque pueden existir casos particulares en donde el diseñador del sistema decide por una cuestión de necesidad romper con esta regla y manejar dos tablas con clave primaria equivalente, produciéndole así una ruptura del modelo normalizado en su estado puro. Este tipo de situaciones son las menos y suelen ser casos prácticamente aislados.

Un ejemplo típico de esto es donde se tiene una tabla en la cual se registran los datos de los clientes y otra tabla de nombre clientes a depurar en la cual se encuentran todos los códigos de cliente que se desean depurar(eliminar) del sistema, en este caso ambas tablas tendrán como clave primaria la misma.

### 5. ¿Cuál es el sentido de generar una clave foránea?

Permiten garantizar la consistencia de los datos. Dado que evitan la supresión de datos que refieran a otra entidad de mayor nivel, evitándose así que los datos contenidos en la base queden inconsistentes. Un ejemplo de esto es:



En este caso existe una clave foránea definida entre el campo id cliente de la tabla factura cliente que referencia al id cliente de la tabla cliente. Esto garantiza que no pueda eliminar ningún registro de la tabla cliente si existe alguna factura registrada en la tabla factura cliente para el id cliente que se quiere borrar. La validación que se realiza es la siguiente:

Al intentar borrar de la tabla cliente el cliente con id cliente = 16 se verifica si existe algún registro en la tabla factura cliente en donde en el campo id cliente tenga como valor 16, si no existe ningún registro entonces se puede eliminar el registro corresponde al id cliente 16 de la tabla cliente, pero en caso de que exista alguna factura para el id cliente 16 entonces la base de datos emitirá un mensaje de error informando que no se puede eliminar el registro con id cliente 16 de la tabla clientes ya que existen facturas para dicho id cliente en la tabla facturas.

#### **6. ¿Puede existir más de una clave foránea para una misma tabla?**

Sí. Habrá tantas claves foráneas como atributos o conjunto de atributos pueden ser validados contra lo ingresado en otra tabla.

#### **7. ¿Es necesario crear todas las restricciones de integridad (claves foráneas) que sean posibles sobre un atributo o conjunto de atributos?**

Esto dependerá habitualmente de la situación que se está modelando, pero como regla general hay que evitar el crear restricciones redundantes, es decir dos o más restricciones para validar el mismo campo dentro de una tabla.

#### **8. ¿En qué circunstancia es necesario crear claves foráneas compuestas?**

Esta situación se presenta en aquellos casos en donde se tienen dos tablas que se relacionan mediante dos o más campos. Y la validación debe realizarse en forma conjunta, dado que la validación individual de cada campo puede producir un resultado distinto al de la validación realizada en conjunto. Un ejemplo de esto es el siguiente:

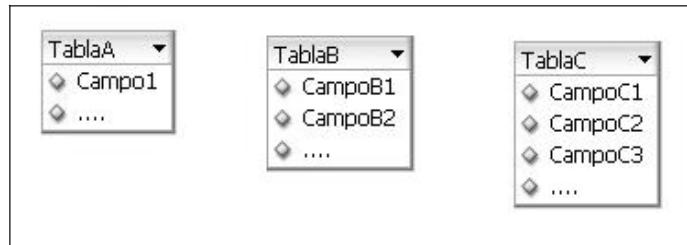


Las claves foráneas a crear son las siguientes:

- ✓ En la tabla provincia se crea una clave foránea sobre el campo id pais que referencia a la tabla pais campo id pais.
- ✓ En la tabla localidad se crea una clave foránea compuesta sobre los campos id pais - id pcia, la cuál referencia a la tabla provincia campos id pais – id pcia.

**9. ¿Las relaciones derivables son dibujadas en el Diagrama Lógico?**

No. Dado que las relaciones representadas en el diagrama lógico coinciden con las restricciones de clave foránea a ser implementadas. Un ejemplo de esto es:



Los campos A1, B1 y C1 son el mismo atributo, es decir guardan el mismo valor.

Los campos B2 y C2 son el mismo atributo, es decir guardan el mismo valor.

La tabla A y B se relacionan mediante los campos A1 y B1.

La tabla B y C se relacionan mediante los campos (B1,B2) y (C1,C2)

Si analizamos lo anterior vemos que se podría decir que también existe una relación entre la tabla A y C mediante los campos A1 y C1 pero no es correcto representarla dado que ya se encuentra incluida en la relación existente entre la tabla B y C.

También del análisis de este ejemplo podemos ver como se cumple lo expuesto en el punto 8, ya que no resultaría correcto validar el campo C1 contra A1 y separadamente el campo C1 contra B2, dado que ambos valores pueden existir en forma independiente pero no necesariamente en forma combinada.

**10. ¿Respecto a las restricciones de dominio (Check), puede una misma restricción afectar a más de un campo?**

No. Se debe crear una restricción de dominio distinta por cada campo al que se le desea restringir el dominio.

**11. ¿Los campos que forman parte de la clave primaria se los debe declarar como Not Null?**

No. Ya que cuando se crea la clave primaria implícitamente ya queda definido que dichos campos no admiten valores nulos. Es importante recordar que la clave primaria es el resultado de la combinación de una restricción única más una restricción not null.

**12. ¿Es necesario declarar todos los campos que no forman parte de la clave primaria como Not Null?**

No. Los campos que no forman parte de la clave primaria no necesariamente deben ser definidos como Not Null, sino que dependerá de la realidad que se está modelando.

**13. ¿Se deben incluir en una tabla los campos calculables?**

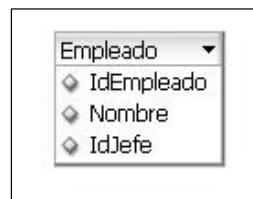
No se incluyen, ya que al ser calculables los podemos obtener a partir de otro campo o campos.

Es importante aclarar que existen veces en donde el diseñador decide en forma consciente incluir dichos campos en la tabla, esto lo que hace es romper con el modelo normalizado, ya que incluye en el mismo redundancia innecesaria. Lo que sucede es que hay veces en donde por una cuestión de ganar en rendimiento se decide romper con esta regla.

Aclaración: Hay que tener en cuenta que el proceso de de-normalización es algo que se relaciona con el ajuste de rendimiento y se suele realizar en una etapa posterior a la normalización.

**14. ¿Para qué se utilizan las entidades recursivas?**

Las entidades recursivas son utilizadas principalmente para reflejar la existencia de una relación jerárquica entre dos atributos de una misma entidad. Un ejemplo común de este tipo de entidades es cuando se quiere reflejar la relación empleado – jefe.



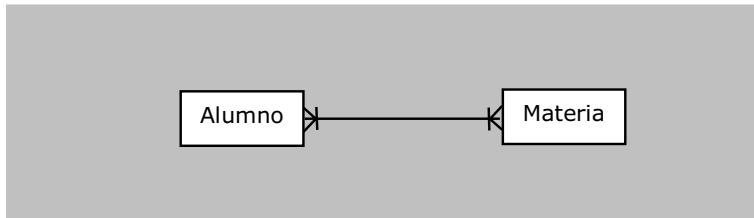
En la tabla empleado para cada empleado que forma parte de la compañía se tendrá en el campo id jefe el id empleado correspondiente al empleado que es su jefe. Lo que se busca en este ejemplo es poder representar en una entidad el organigrama de la empresa.

**15. ¿Puede existir el caso de que una entidad se relacione con otra a través de dos atributos pero que uno sea parte de la clave primaria y el otro no?**

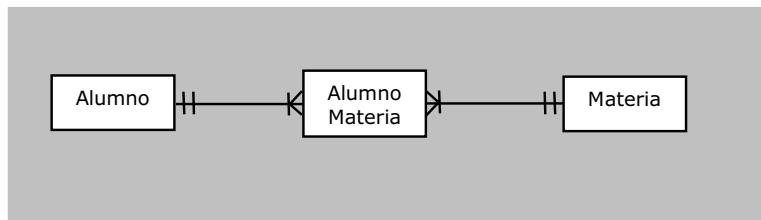
Si el modelo está correctamente definido esto no debe suceder.

**16. ¿En un diagrama de entidad relación pueden existir relaciones de muchos a muchos?**

En el DER realizado en el modelo conceptual pueden existir relaciones del tipo muchos a muchos, pero cuando a dicho diseño se le aplica la técnica de normalización, las relaciones de este tipo se transforman en una nueva tabla de tipo asociativa (relacional) que vincula las dos tablas que estaban unidas por la relación.

**Modelo conceptual**

Muchos alumnos pueden cursar la misma materia y muchas materias pueden ser cursadas por muchos alumnos.

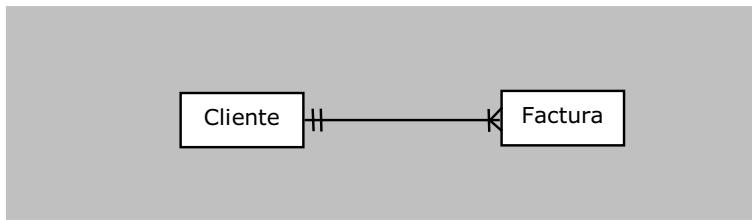
**Modelo normalizado**

### 17. ¿Cómo me doy cuenta en el diagrama lógico si la relación es de uno a muchos o de muchos a muchos?

Como regla general se puede adoptar la siguiente:

- ✓ Si la clave primaria de la tabla esta formada por un único atributo entonces la cardinalidad para cualquier relación de esa entidad siempre será uno. Ej. Tabla clientes donde la clave primaria es el id de cliente, ya que no pueden existir dos clientes que comparten el mismo número, por lo que nunca se tendrán dos registros con mismo id.
- ✓ Si el campo por el cuál se une la relación para una entidad no son parte de la clave primaria entonces la cardinalidad para esa entidad será de muchos a menos que exista una restricción de tipo unique sobre la/s columna/s. Ej. Tabla clientes donde la clave primaria es el id de cliente y la tabla facturas donde la clave primaria es el número de factura y uno de sus atributos no clave es el id de cliente, a consecuencia de que el id de cliente no es clave y tampoco tiene una restricción unique que lo afecte, entonces resulta que en la tabla facturas podemos encontrar muchas facturas distintas para el mismo id de cliente.

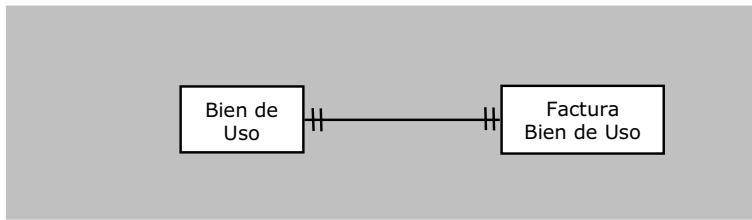
Como quedaría dibujada la relación entre estas dos entidades, en la entidad factura el cliente puede tener más de una factura:



Un ejemplo de relaciones de uno a uno sería el siguiente:

Tabla de Bienes de Uso en donde se registran todos los bienes de uso que posee la compañía y tabla de facturas de bienes de uso en donde se registran las facturas cada vez que se vende un bien de uso, como un bien de uso solo se lo puede vender una única vez es que nunca podrían existir dos o más facturas para el mismo bien por lo que para garantizar esta condición es que se crea sobre el campo id bien de uso de la tabla facturas una restricción de unicidad(unique), a consecuencia de esto es que la relación entre ambas tablas es de uno a uno.

Como quedaría dibujada la relación entre estas dos entidades, en la entidad factura bien de uso un bien únicamente puede figurar una sola vez:



**18. ¿En un DER la cardinalidad de la relación utilizada para representar la recursividad es de uno a muchos o de uno a uno?**

Puede ser tanto de uno a muchos como de uno a uno, siempre dependerá de la realidad que se está modelando.

**19. Tabla de históricos. ¿Cómo se debe definir las claves para una tabla en la cuál se desea llevar un histórico?**

Cuando se habla de histórico lo que se busca es poder guardar la historia de las distintas instancias por las que pasó un atributo, por ejemplo un caso común es el de la lista de precios, habitualmente se desea conservar un detalle de cómo fue variando el precio de un bien a lo largo del tiempo. Para este caso se necesitaría una tabla con la siguiente definición:



Mediante la combinación de los campos producto y fecha para formar la clave de la tabla es que se logra guardar los distintos precios que se le asignaron a un producto a lo largo del tiempo, además se cumple al mismo tiempo con la condición de que un artículo no tenga más de un precio para el día. Como objeción a esto uno podría decir pero que pasa si se vive un contexto inflacionario donde la marcación de precios podría ser cada hora, la respuesta a esta pregunta es que el tipo de dato correspondiente a la fecha podría ser tel, tipo fecha, hora, minuto y segundo con lo cuál en el caso mas extremo se podría hasta llevar el precio segundo a segundo, pero siempre se estaría garantizando la unicidad dado que nunca se encontraran dos registros con el mismo producto y fecha.

## 20. Una vez diseñado el modelo. ¿Como se hace para validarlo?

Una buena práctica es imaginarse distintas situaciones que podrían darse y ver como responde el modelo diseñado a ellas.

Una manera eficiente es testear cada tabla en particular y luego realizar una prueba de integración en la cuál se prueba el modelo como un todo, evaluando así como funciona de manera independiente y en conjunto. Es importante recordar que al momento de diseñar un modelo como así también durante el testeo del mismo no se debe perder de vista que el modelo tiene que cumplir con las condiciones que surgen de la situación relevada. Esto es el algo que nunca debe perderse de foco, para hacer entonces un modelo válido que realmente represente la realidad que se intenta modelar.

## 21. ¿Cómo se procede a borrar registros en tablas relacionadas?



En la tabla PROVINCIA existe una foreign key sobre el campo id\_pais que referencia al campo id\_pais a la tabla PAIS.

En la tabla LOCALIDAD existe una foreign key sobre los campos id\_pais, id\_pcia que referencian a los campos id\_pais, id\_pcia de la tabla PROVINCIA.

Si analizamos lo anterior observamos que existe una relación de padre - hijo entre la tabla PAIS y PROVINCIA, como así también entre la tabla PROVINCIA y LOCALIDAD. Por transitividad se puede decir que también existe una relación de este tipo entre la tabla PAIS y LOCALIDAD.

La existencia de estas relaciones determina que no puedo borrar un país de la tabla PAIS si existe alguna provincia para ese país en la tabla PROVINCIA, y de esta misma forma tampoco puede borrar la combinación país/provincia de la tabla PROVINCIA si existe alguna localidad asociada a los mismos en la tabla LOCALIDAD.

Entonces las preguntas que puede uno hacerse serían:

¿Como hago para borrar un país de la tabla PAIS?

Lo primero que debo borrar son los registros correspondientes a la tabla de menor nivel dentro de la jerarquía de restricciones generada a partir del anidamiento de las foreign key, en este ejemplo primero se debe borrar todas las localidades que existan en la tabla LOCALIDAD para el país-provincia, luego borrar el registro correspondiente al país-provincia de la tabla PROVINCIA y recién entonces borrar el registro correspondiente al país en la tabla de PAIS.

¿Puede borrar directamente el registro correspondiente a una determinada provincia de la tabla PROVINCIA?

Como existe una otra tabla que tiene una foreign key que referencia a la tabla PROVINCIA, es que al momento de borrar se valida que no exista dicha provincia en la/s tabla/s que hacen referencia a esta, en nuestro ejemplo que no exista ninguna localidad para la provincia a borrar, en caso de existir no nos permitirá borrar la provincia, por lo que primero debemos borrar las localidades y luego las provincias. Este control de CONSISTENCIA es realizado por el gestor de la base de datos, quién tiene en su diccionario registrado todas las constraint de tipo foreign key existentes como así también la relación de jerarquía que existen entre ellas.

¿Cuál es el sentido que tiene que no me permita borrar una provincia de la tabla PROVINCIA si es que existe alguna localidad asociada a la misma?

La idea de las restricciones foreign key es garantizar la CONSISTENCIA de los datos contenidos en las tablas, por lo que si pudiera borrar el registro correspondiente a una determinada provincia de la tabla PROVINCIA, dejando en la tabla LOCALIDAD las localidades asociadas a la misma entonces estaría teniendo una inconsistencia a nivel de datos, ya que tendría localidades que están asociadas a un id\_pcía que no existe en la tabla de PROVINCIA. Es por esto como dijimos antes que existen las foreign key como restricción, ya que me permiten garantizar desde el propio modelo la consistencia de los datos contenidos en este.

Veamos ahora lo anterior utilizando un caso práctico con datos como ejemplo:

PAIS	
ID_PAIS	NOMBRE
1	ARGENTINA
2	BOLIVIA
3	CHILE

PROVINCIA		
ID_PAIS	ID_PCIA	NOMBRE
1	1	BUENOS AIRES
1	2	FORMOSA
2	1	LA PAZ

LOCALIDAD			
ID_PAIS	ID_PCIA	ID_LOCAL	NOMBRE
1	1	BUENOS AIRES	AVELLANEDA
1	1	FORMOSA	HAEDO

Supuesto 1: Se quiere borrar el país 1(Argentina) de la tabla PAIS.

¿Que pasa si quiero borrar el país 1(Argentina) de la tabla PAIS?

Al intentar realizar esta acción recibiría un mensaje de error de parte de la base de datos, ya que como existe al menos una provincia para dicho país no lo puedo borrar.

¿Que pasaría entonces si quiero borrar las provincias asociadas al país 1(Argentina) de la tabla PROVINCIA?

Tampoco podría realizar esta acción, ya que existe al menos una localidad asociada a dicha provincia en la tabla LOCALIDAD.

¿Como tendría que hacer entonces para borrar el país 1(Argentina) de la tabla PAIS?

Los pasos a realizar serían:

- Borrar todas las localidades que existen para las provincias asociadas al país 1(Argentina) en la tabla LOCALIDAD.
- Borrar todas las provincias que existen para el país 1(Argentina) en la tabla PROVINCIA.
- Borrar el país 1(Argentina) de la tabla PAIS.

Supuesto 2: Se quiere borrar el país 2(Bolivia) de la tabla PAIS.

¿Que pasa si quiero borrar el país 2(Bolivia) de la tabla PAIS?

Al intentar realizar esta acción recibiría un mensaje de error de parte de la base de datos, ya que como existe al menos una provincia para dicho país no lo puedo borrar.

¿Que pasaría entonces si quiero borrar las provincias asociadas al país 2(Bolivia) de la tabla PROVINCIA?

Como no existe ninguna localidad asociada a ninguna provincia correspondiente a dicho país, entonces puedo borrar las provincias asociadas a dicho país de la tabla PROVINCIA.

¿Como tendría que hacer entonces para borrar el país 2(Bolivia) de la tabla PAIS?

Los pasos a realizar serían:

- a) Borrar todas las provincias que existen para el país 2(Bolivia) en la tabla PROVINCIA.
- b) Borrar el país 2(Bolivia) de la tabla PAIS.

Supuesto 3: Se quiere borrar el país 3(Chile) de la tabla PAIS.

¿Que pasa si quiero borrar el país 3(Chile) de la tabla PAIS?

Lo puedo borrar perfectamente ya que no existe ninguna provincia asociada a dicho país en la tabla PROVINCIA y por consecuencia ninguna localidad asociada a alguna provincia correspondiente a dicho país.

