



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

“CONSTRUCCIÓN DE UN SOFTWARE
EDUCATIVO DE BASE DE DATOS”

T E S I S
PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACIÓN
QUE PRESENTA
OLGA LIDIA CISNEROS MONROY

DIRECTOR DE TESIS
ING. GABRIELA BETZABÉ LIZARRÁGA RAMÍREZ

MÉXICO, D.F.

SEPTIEMBRE DE 2009



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México por abrirme sus puertas y darme la oportunidad de formarme como profesionalista. Por la educación integral que recibí.

A mis padres que siempre me han brindado todo su amor, su apoyo, su confianza, su comprensión y por creer en mí, por que gracias a ellos crecí en una familia maravillosa que me ha impulsado y ayudado a alcanzar todos mis objetivos. Mil gracias por todo su esfuerzo y hacer de mí la persona que soy. Por ser tan persistentes y lograr que culminara este proyecto.

A mi hermana a quien adoro y ha sido una pieza crucial en mi vida, espero te sirva de ejemplo al igual que a mis sobrinos Donovan e Ian.

A Brandon por ser lo más maravilloso que la vida pudo regalarme. Eres mi inspiración y el motor de mi vida.

A mi esposo, a mis tíos, a mi cuñado Oscar y a mis primos por sus consejos y apoyo.

A mis amigos que me acompañaron en este camino, por todas las vivencias que compartimos, por su amistad incondicional, su apoyo, su paciencia, sus consejos, muy en especial a Nancy A., Andrei S., Edgar R., Jesús D., Reyna V., Pablo C., Cristina A., Sergio G.

A mi asesora por todo su tiempo, su paciencia, su apoyo, su dedicación, su constancia y su amistad. Al fin lo logramos!!!! Gracias por no abandonarme.

A cada uno de mis profesores por compartir sus conocimientos, su experiencia y ser parte de mi formación; muy en especial a la profesora Ma. Teresa Peñuñuri por su impulso y confianza.

A mis asesores por el tiempo dedicado a la revisión de este trabajo.

I. Introducción	5
1.1 Nuevas Tecnologías e Innovación Educativa	5
1.2 Enseñanza a través del uso de tutoriales	6
1.3 Etapas de la Metodología CASE (Computer Aided Systems Engineering).....	7
1.4 Tendencias en la Educación Virtual.....	8
II. Análisis del Software Educativo Interactivo usando Metodología CASE	9
2.1 Diagrama Entidad-Relación.....	9
2.2 Diagrama Jerárquico Funcional	10
2.3 Diagrama De Flujo De Datos.....	11
III. Diseño del Software Educativo Interactivo	13
3.1 Diseño de la base de datos	13
3.2 Diseño de la aplicación	16
3.3 Especificaciones De Los Programas	17
IV. Construcción del Software Educativo Interactivo	24
a) BIENVENIDA.....	26
b) HERRAMIENTAS CASE.....	27
• Metodología Case	27
• Fase de estrategia	29
• Fase de análisis.....	30
• Fase de diseño	31
• Construcción	32
• Documentación	32
• Transición.....	33
• Producción	33
c) TEORÍA DE BASE DE DATOS.....	34
• Cronología de las bases de datos.....	34
• Definición de una base de datos.....	38
• Objetivos de una base de datos	38
• Definición de un sistema de base de datos.....	38
• Ventajas de un sistema de base de datos.....	38
• Desventajas de un sistema de base de datos	38
• Sistema Manejador de Base de Datos (DBMS).....	39
• Administrador de la base de datos	40
• Diccionario de datos.....	40
• Estructura del diccionario de datos	40
• Introducción al análisis, diseño y construcción	42
• Teoría del modelo de datos Entidad-Relación	44
d) ANÁLISIS	47
• Identificar entidades	47
• Descripción de entidades	48
• Establecer las relaciones entre entidades	48
• Diferencias entre atributos y entidades	49
• Identificadores únicos	50
• Identificadores únicos secundarios	51
• Resolución de relaciones M:M	52
• Normalización en Análisis.....	54
• Subtipos o subentidades y supertipos o superentidades.....	56

• Relaciones excluyentes (arco).....	57
e) ÁLGEBRA RELACIONAL	58
• Teoría	58
• Operadores fundamentales y adicionales.....	60
• Ejemplos.....	61
• Estándares	65
f) DISEÑO	68
• Teoría de diseño	68
• Llaves primarias.....	68
• Llaves foráneas	69
• Integridad de datos.....	69
• Diseño inicial de la base de datos	70
• Normalización en diseño.....	80
• Conversión a la primera forma normal en diseño	81
• Conversión a la segunda forma normal en diseño	81
• Conversión a la tercera forma normal en diseño.....	82
• Definir la integridad referencial.....	84
• Diseño de índices	86
• Establecer vistas.....	88
• Desnormalización de la base de datos.....	89
• Planeación de uso del almacenamiento físico.....	92
g) CONSTRUCCIÓN	93
• Integridad referencial	93
• DDL (Data Definition Language / Lenguaje de Definición de Datos)	93
• DCL (Data Control Language / Lenguaje de Control de Datos)	94
• DML (Data Manipulation Language / Lenguaje de Manipulación de Datos)	94
V. Administración del Tutorial	95
5.1 Administración de Capítulos.....	95
5.2 Administración de Temas	97
5.3 Administración de Exámenes.....	99
5.4 Administración de Alumnos	101
5.5 Administración de Tipo de Preguntas	103
5.6 Administración de Preguntas, Subpreguntas y Respuestas.....	105
5.7 Administración de Asignación de Respuestas	107
5.8 Consulta de Historial Académico	109
5.9 Gráfica de Resultados de Exámenes	110
VI. Conclusiones	111
Apéndice I. Ejemplo código de página ASP	112
Apéndice II. Ejemplo de código store procedure	113
Bibliografía	114

I. Introducción

1.1 Nuevas tecnologías e innovación educativa

La importancia de las nuevas tecnologías ha sido y es un hecho incuestionable. Su influencia se ha dejado sentir en la totalidad de los ámbitos de nuestra sociedad (transportes, comunicaciones, producción, medicina, seguridad, etc.). En consecuencia, no es de extrañar que haya alcanzado asimismo, un aspecto de tanta importancia para nuestras vidas como es el de la educación.

Mediante la expresión "Nuevas tecnologías" suele hacerse referencia al conjunto de dispositivos e instrumentos de aparición relativamente reciente. Por su parte, la expresión "Innovación educativa" suele emplearse para designar aquellos aspectos docentes que suponen un cambio cualitativo con respecto a las formas y maneras del proceder tradicional.

Aunque pueda parecer sorprendente, muchos alumnos universitarios no han usado todavía nunca una computadora. La novedad que ésta implica para ellos favorece la curiosidad y atención, máxime en aquellos que consideran la informática como un aspecto de gran importancia en la sociedad actual.

Por otra parte, desde una perspectiva meramente perceptual, el aprendizaje mediante computadora resulta con frecuencia más atractivo, pues suele incluir gráficos y dibujos, profusión de colores, movimiento, sonido, etc. La enorme interacción que demanda (entrada de respuestas, selección de opciones, confirmación de operaciones, etc.) crea la ilusión de que el tiempo transcurre "más rápidamente" y si el programa está suficientemente bien diseñado provoca, en ocasiones, que la situación de aprendizaje se transforma en una situación lúdica.

Finalmente, la actuación exitosa con los programas genera una agradable sensación de comprensión e incluso dominio del aspecto abordado que hace desear un reiterado uso de tales materiales conllevando dicho uso un mejor aprendizaje.

Una de las principales contribuciones de las Tecnologías de la Información y la Comunicación (TIC), sobre todo de las redes telemáticas, al campo educativo es que abren un abanico de posibilidades en modalidades formativas que pueden situarse tanto en el ámbito de la educación a distancia, como en el de modalidades de enseñanza presencial.

Las perspectivas que las TIC presentan para su uso educativo, exigen nuevos planteamientos que a su vez requerirán un proceso de reflexión sobre el papel de la educación a distancia en un nuevo mundo comunicativo, pero también provocarán un cuestionamiento de las instituciones educativas. En efecto, el entramado de redes de comunicación y las posibilidades crecientes de los sistemas multimedia cuestionan, tanto para la educación a distancia como para la presencial, la utilización de los sistemas educativos convencionales. En este sentido, un posible punto de encuentro podemos encontrarlo en los planteamientos del aprendizaje abierto (Lewis y Spencer, 1986; Lewis, 1988; Salinas y Sureda, 1992). En el aprendizaje abierto, independientemente de la distancia o de si la enseñanza es presencial, la toma de decisiones sobre el aprendizaje la toma el estudiante o los estudiantes mismos. Estas decisiones afectan a todos los aspectos del aprendizaje: si se realizará o no; **qué aprendizaje** (selección de contenido o destreza); **cómo** (métodos, media, itinerario); **dónde aprender** (lugar del aprendizaje); **cuándo aprender** (comienzo y fin, ritmo); **a quién recurrir para solicitar ayuda** (tutor, amigos, colegas, profesores, etc.); cómo será la valoración del aprendizaje, aprendizajes posteriores, etc.

Otro aspecto más concreto que consideramos debe constituir motivo de reflexión es el grado de interactividad y de control de la comunicación que ofrece el sistema. Ambos, interactividad y control, están determinados por las capacidades y recursos tecnológicos de que dispone el emisor y, sobre todo, el receptor, pero dependerá sobre todo del modelo didáctico que inspire el proyecto. Se trata, por tanto, de lograr el equilibrio entre la potencialidad tecnológica aportada por las redes y las posibilidades educativas que el sistema es capaz de poner en juego.

1.2 Enseñanza a través del uso de tutoriales

Las computadoras comenzaron a ser utilizadas como un medio de impartir la enseñanza, de manera que los elementos de materia (reactivos) y su secuencia de exposición, estuviesen adaptados a las diferencias individuales de los alumnos en particular en forma de tutorial. Las primeras experiencias de enseñanza impartida mediante computadora comenzaron en Estados Unidos a principio de 1960 y en Europa en 1965 específicamente en España.

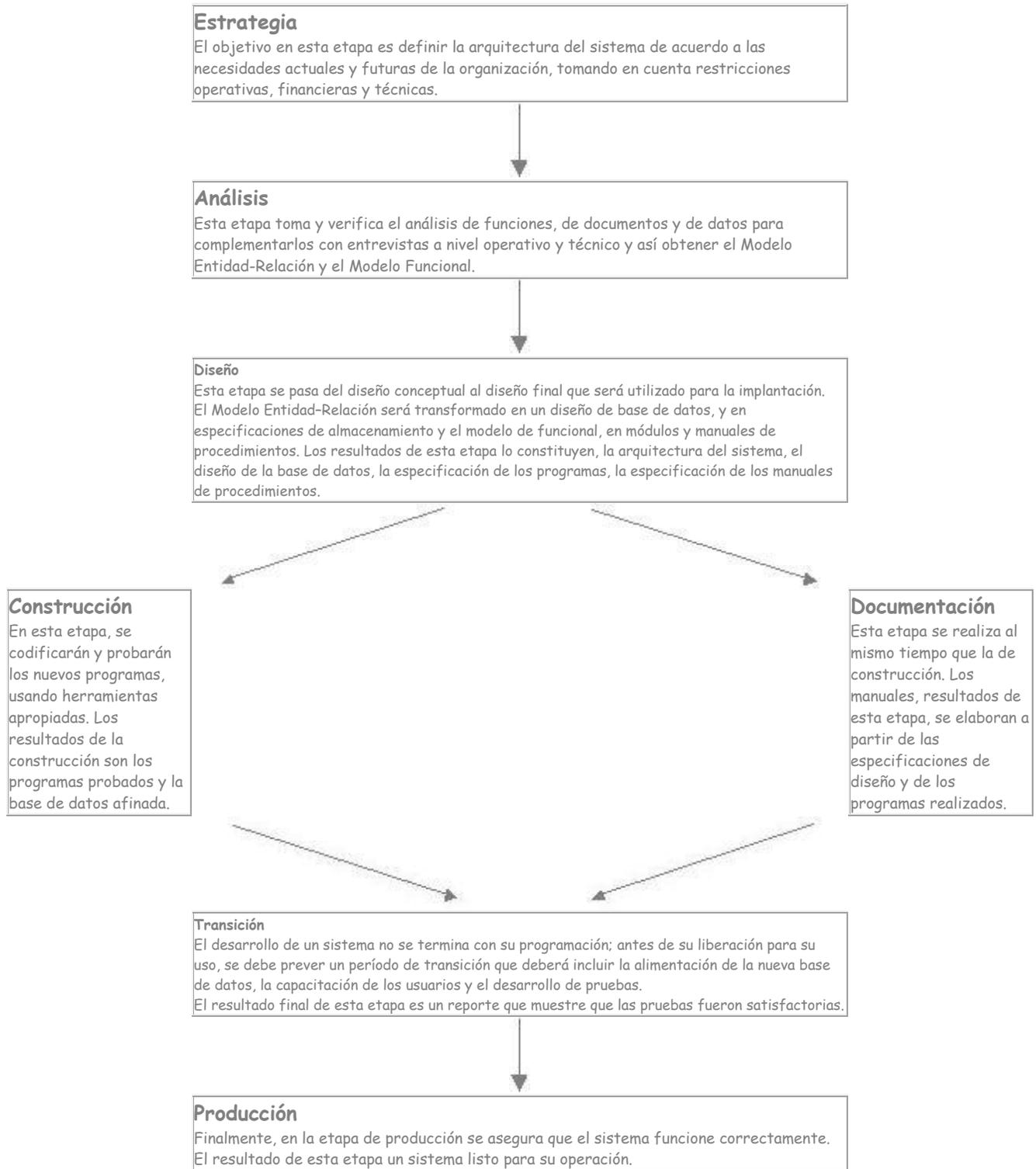
Puede decirse que inicialmente la enseñanza asistida por computadora (EAC) estuvo orientada a incorporar en una máquina los métodos de la enseñanza programada. Esta primera etapa tuvo resultados muy limitados, debido quizá al alto costo de las computadoras y a la poca flexibilidad que permitía la enseñanza programada.

En la actualidad con la expansión del uso de la computadora (propiciada por su abaratamiento) y con el apoyo de otras disciplinas (la comunicación, la psicología educativa, la ingeniería de sistemas entre otras) se han creado sistemas computacionales más poderosos y eficientes que unido a nuevos enfoques en la instrucción por computadora, permiten la elaboración de programas computacionales mucho más versátiles y accesibles a los profesores, investigadores y a los propios estudiantes, dando un giro favorable a su elaboración y por consiguiente a la utilización de la computadora como medio de enseñanza.

En la enseñanza con computadora las diferentes posibilidades son: ejercitación, adiestramiento, simulación, evaluación y tutorías. Los programas cuya función es la tutoría, presentan el material de enseñanza y el estudiante interviene directamente interactuando a través de una computadora. La denominación común de programas tutoriales procede del hecho de que para la enseñanza de un tema específico, dichos programas se convierten en sustitutos del profesor.

Existen estudios que han demostrado la efectividad de la utilización de la computadora como medio de enseñanza. Otros estudios como los realizados por Sanfeliz y Behar (1990) llegaron a la conclusión de que no es la computadora como medio de instrucción la que conduce a mejores resultados en el aprendizaje, sino que es el cuidadoso diseño de los contenidos utilizados para este propósito la principal causa. Esto nos indica que debemos trabajar más en el diseño de los programas y que para ello es necesario un equipo interdisciplinario. También es cierto que estudiantes que requieren de una motivación externa les es más difícil aprender por medio de programas computacionales, no así los estudiantes que tienen una motivación intrínseca y eso es conveniente tenerlo en cuenta cuando se quiera implementar un programa de alguna asignatura con el apoyo de la EAC.

1.3 Etapas de la Metodología CASE (Computer Aided Systems Engineering)



1.4 Tendencias en la educación virtual

La educación a distancia nos brinda la posibilidad de contar con nuevos procesos de aprendizaje y transmisión del conocimiento a través de las redes modernas de comunicaciones; sin embargo requiere crear un nuevo sistema de centros educativos a distancia y en red, capacitar a las personas para que puedan actuar competentemente, así como nuevos escenarios, instrumentos y métodos para los procesos educativos.

La educación virtual, cuyo mejor exponente actual es la red Internet, no es presencial, sino distal y no se basa en recintos espaciales con interior, frontera y exterior, sino que depende de redes electrónicas cuyos nodos de interacción pueden estar diseminados por diversos países. Este entorno multimedia no sólo es un nuevo medio de información y comunicación, sino también un espacio para la interacción y la memorización. Por ello, cada vez es preciso diseñar nuevos escenarios y acciones educativas, es decir, proponer una política educativa específica para el entorno cibernético.

Al apoyar una política educativa específica para las aulas virtuales no se pretende sustituir a las Universidades y escuelas; lo que se pretende es que a los centros académicos se les superpongan redes educativas digitales a través de las cuales se desarrollarían procesos educativos del entorno virtual, complementarios a los entornos reales.

El acceso universal a esos escenarios y la capacitación para utilizar competentemente las nuevas tecnologías se convierten en dos nuevas exigencias emanadas del derecho a que cualquier ser humano reciba una educación adecuada al mundo en el que vive.

El derecho a la educación universal tiene que ampliarse, no basta con enseñar a leer, escribir y hacer cálculos matemáticos; además de introducir conocimientos básicos de historia, literatura y ciencias, todo ello es necesario y lo seguirá siendo en los espacios naturales y urbanos en los que tradicionalmente se ha desarrollado la vida social, pero de manera progresiva, gran parte de la vida social se desplegará en el espacio electrónico y virtual, y por eso es preciso implementar la escuela tradicional con una escuela digital y virtual.

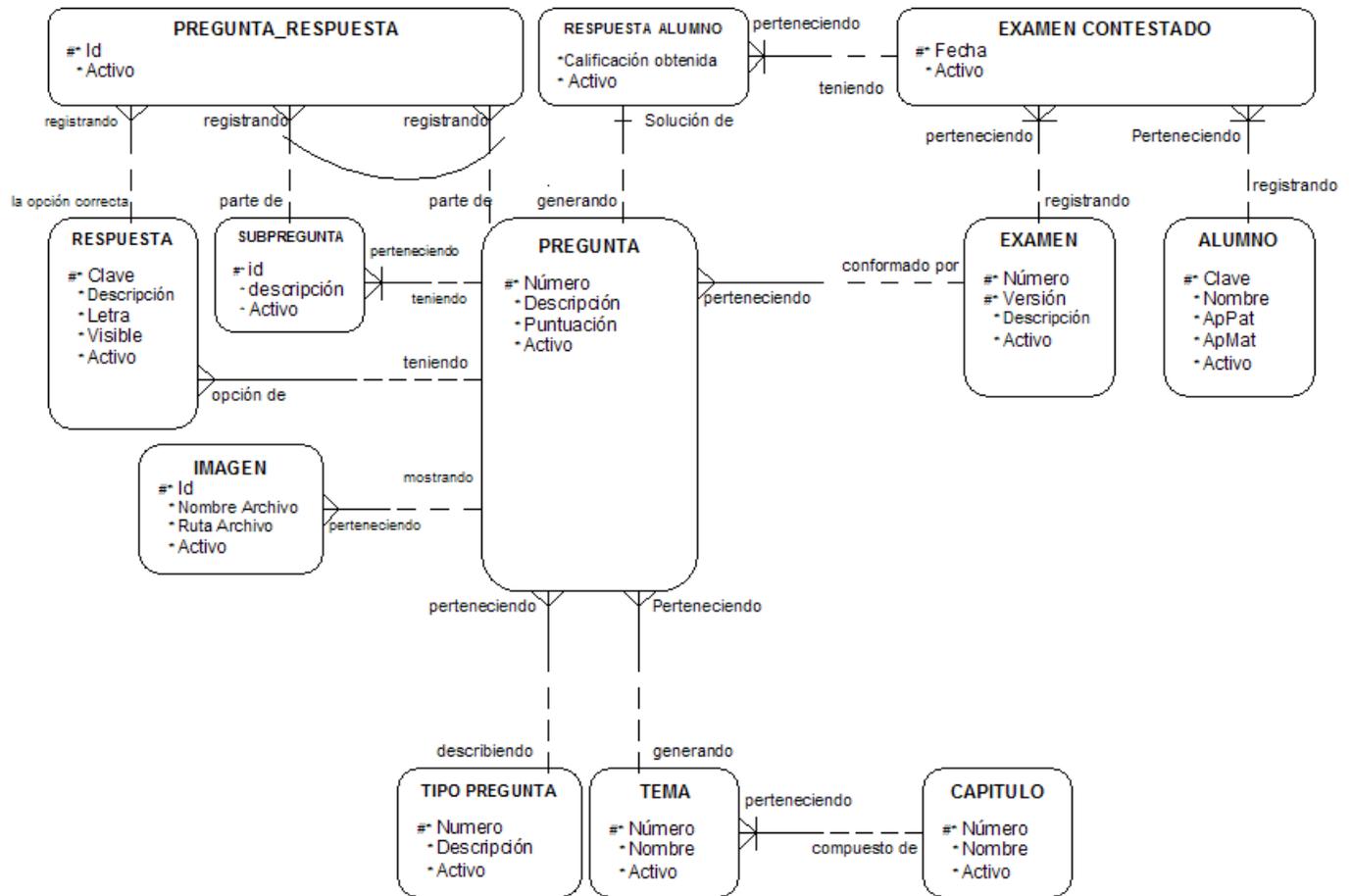
En el nuevo milenio, las redes telemáticas son la expresión más desarrollada del entorno virtual debido a su carácter multimedia, muy importante a efectos educativos, y al grado de interactividad. La realidad virtual abre nuevas posibilidades para el desarrollo de procesos perceptivos y sensoriales.

Una de las tareas que debe hacerse es crear los escenarios telemáticos educativos (aulas virtuales, escuelas y universidades electrónicas, Intranets para la educación, programas televisivos específicos, videojuegos adecuados a las edades.), hay una enorme labor de formación de agentes educativos por llevar a cabo. Al respecto, la aparición de un nuevo lenguaje en Internet, el VRML (Virtual Reality Modeling Language "Lenguaje para Modelado de Realidad Virtual"), tiene gran importancia, dicho lenguaje permite generar escenarios virtuales en donde interactúan modelos digitales manejados por personas, pudiendo moverse, gesticular, intercambiar objetos digitales como ficheros, mensajes, etc

II. Análisis del software educativo interactivo usando metodología CASE

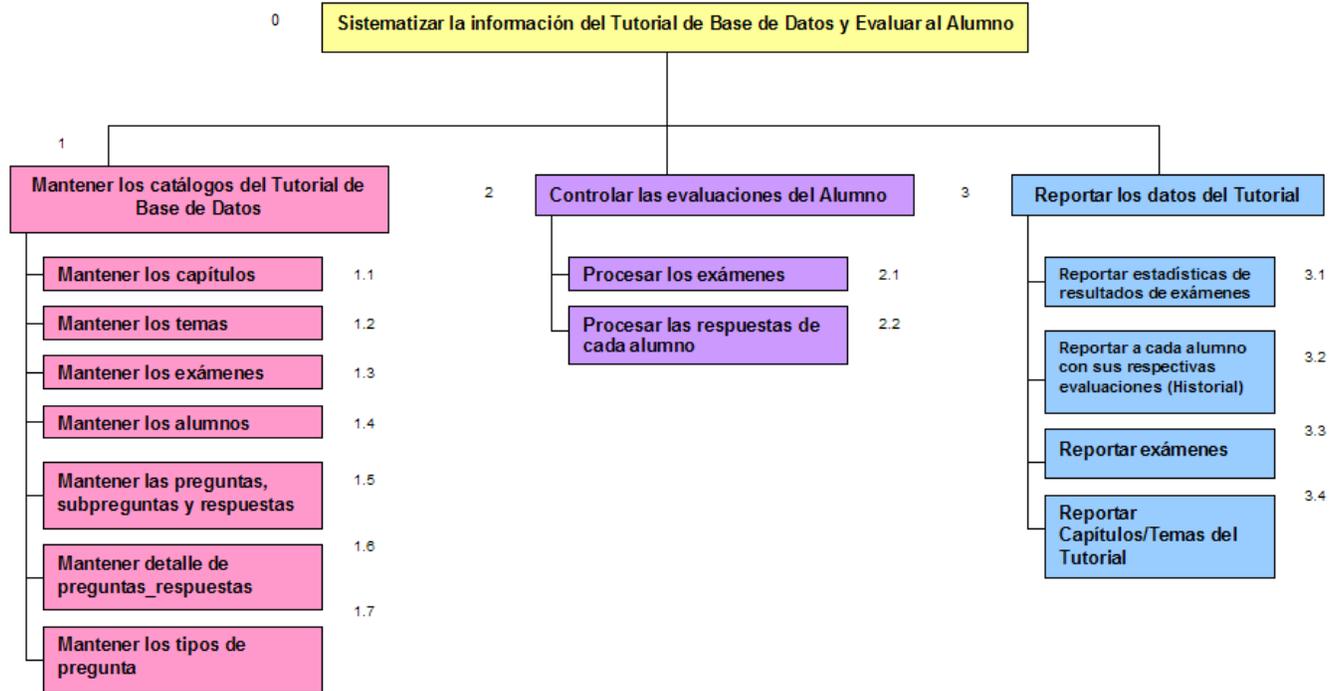
2.1 Diagrama Entidad-Relación

El Diagrama Entidad – Relación (DER) modela mediante relaciones lógicas todos los datos involucrados en el sistema, de tal manera que cualquier tipo de explotación (consulta, inserción o modificación) sean posibles.



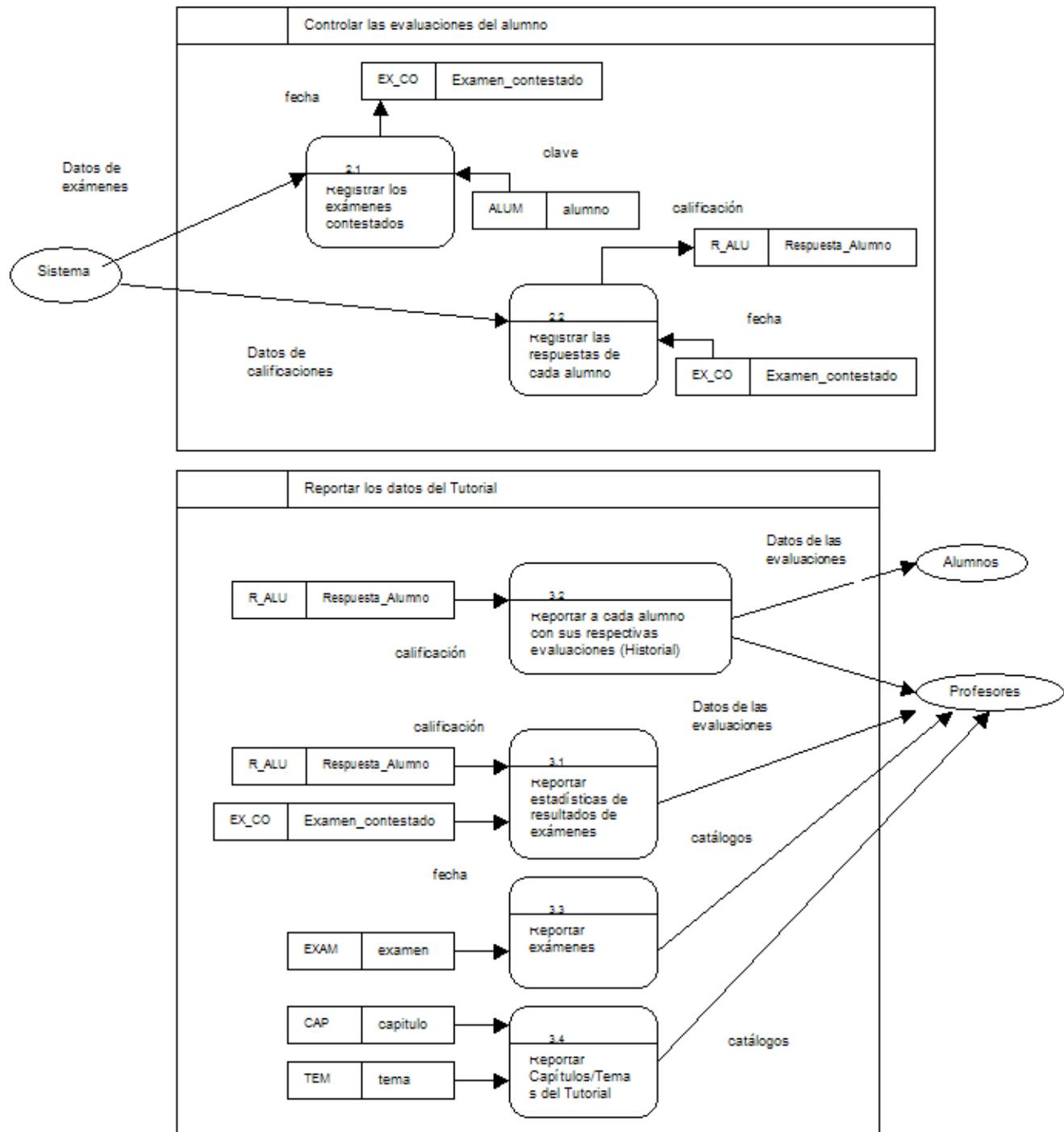
2.2 Diagrama Jerárquico Funcional

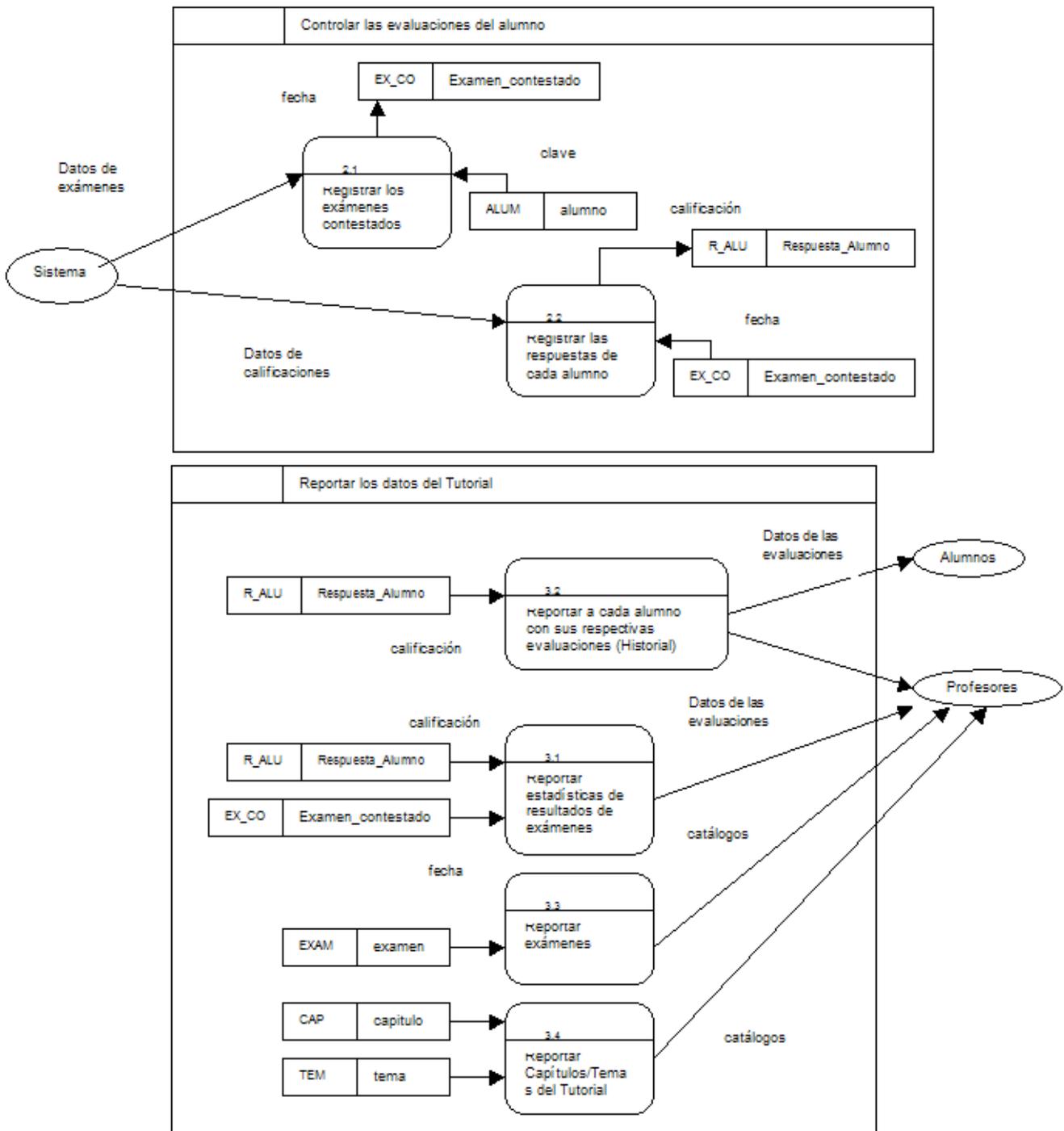
El Diagrama Jerárquico Funcional modela los diferentes servicios que ofrecerá el sistema mediante una organización y clasificación de las diversas funciones y subfunciones que fueron identificadas en el análisis.



2.3 Diagrama De Flujo De Datos

El Diagrama de Flujo de Datos modela el flujo de la información de cada uno de los procesos, así como sus datos de entrada y datos de salida





III. Diseño del software educativo interactivo

El diseño es la transformación del DER; las entidades se convierten en tablas, los atributos en columnas, los identificadores únicos en llaves primarias y las relaciones en llaves foráneas.

3.1 Diseño de la base de datos

CAPITULOS

Columna	Número	Nombre
Tipo De Llave	PK	
Nulos/Únicos	NN-U	NN
Ejemplos	1	Bienvenida
	2	Teoría
	3	Análisis

TEMAS

Columna	NoCapitulo	Número	Nombre
Tipo De Llave	PK, FK	PK	
Nulos/Únicos	NN	NN-U	NN
Ejemplos	2	1	Cronología de las bases de datos
	2	2	Definición de una base de datos
	3	1	Identificar Entidades

ALUMNOS

Columna	Clave	Nombre
Tipo De Llave	PK	
Nulos/Únicos	NN-U	NN
Ejemplos	1	Carlos Ojeda Jiménez
	2	Raúl Ortega Hernández
	3	Donovan Estrella Alarcón

EXÁMENES

Columna	Número	Versión	Descripción
Tipo De Llave	PK	PK	
Nulos/Únicos	NN	NN-U	NN
Ejemplos	1	1	Examen de teoría y análisis Sem 2003-2
	1	2	Examen de teoría y análisis Sem 2004-1
	2	1	Examen de Álgebra Relacional Sem 2003-2

PREGUNTAS

Columna	NoExamen	VersionExam	NoCapitulo	NoTema	Número	Descripción	Puntuación
Tipo De Llave	FK1	FK1	FK2	FK2	PK		
Nulos/Únicos	NN	NN	NN	NN	NN-U	NN	NN
Ejemplos	1	1	2	1	1	Relaciona cada pregunta con su respuesta correcta.	4
	1	1	2	1	2	Menciona las etapas para tener una base de datos operacional e indique que se obtiene de cada una de ellas	2
	1	1	2	1	3	¿Cuándo se utiliza UID?	2

SUBPREGUNTAS

Columna	Num_Preg	Id	Descripción
Tipo De Llave	PK, FK	PK	
Nulos/Únicos	NN-U	NN-U	NN
Ejemplos	1	1	Esta compuesto de: Tablas Base, User accesible y Views
	1	2	Controla concurrencia, aplica restricciones de seguridad, evita redundancia e inconsistencia
	1	3	Conjunto de información organizada para servir a una aplicación o más, de la mejor manera posible

IMAGÉNES

Columna	Num_Preg	Id	NombreArchivo	RutaArchivo
Tipo De Llave	FK	PK	PK	
Nulos/Únicos	NN	NN-U	NN-U	NN
Ejemplos	4	1	Exam1_4.gif	./Images/
	5	2	Exam1_5.gif	./Images/
	6	1	Exam1_6.gif	./Images/

RESPUESTAS

Columna	Clave	Descripción	Letra
Tipo De Llave	PK		
Nulos/Únicos	NN-U	NN	
Ejemplos	1	Base de datos	a
	2	Sistema manejador de base de datos	b
	3	Entidad	c

PREGUNTAS_RESPUESTAS

Columna	Id	Num_Preg	Preg_Sub	Subpregunta	Clave_Resp
Tipo De Llave	PK	FK1	FK2	FK2	FK3
Nulos/Únicos	NN-U	N	N	N	NN
Ejemplos	1	NULL	1	1	7
	2	NULL	1	2	5
	3	NULL	1	3	1

EXÁMENES CONTESTADOS

Columna	ClaveAlumno	NoExamen	Versión	Fecha
Tipo De Llave	PK, FK1	PK, FK2	PK, FK2	PK
Nulos/Únicos	NN-U	NN-U	NN-U	NN
Ejemplos	1	1	1	21-Septiembre-2003
	1	2	1	26-Septiembre-2003
	2	1	1	29-Agosto-2003

RESPUESTAS ALUMNOS

Columna	ClaveAlumno	NoExamen	Versión	Fecha	NoPregunta	CalificacionObtenida
Tipo De Llave	PK, FK1	PK, FK1	PK, FK1	PK, FK1	PK, FK2	
Nulos/Únicos	NN	NN	NN	NN	NN-U	NN
Ejemplos	1	1	1	21-Septiembre-2003	1	3
	1	1	1	21-Septiembre-2003	2	2
	1	1	1	21-Septiembre-2003	3	0

3.2 Diseño de la aplicación

El Software Educativo se desarrollo con ASP's (Active Server Pages) propuestas en diciembre de 1996 por Microsoft, es una tecnología que nos permite crear páginas Web dinámicas que han sido comercializadas como un anexo a Internet Information Services (IIS).

La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante; intenta ser solución para un modelo de programación rápida, permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML.

El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del web, es ejecutado, por lo que el usuario no tiene acceso más que a la página resultante en su navegador

Como base de datos SQL Server. Es un sistema de gestión de base de datos relacionales basado en el lenguaje Transact-SQL y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea, así como de tener las siguientes ventajas:

- 1) Soporte de transacciones
- 2) Escalabilidad, estabilidad y seguridad
- 3) Soporta procedimientos almacenados
- 4) Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente
- 5) Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información
- 6) Además permite administrar información de otros servidores de datos.

La decisión de utilizar SQL Server fue que al inicio del desarrollo de la tesis MySQL que es un manejador de base de datos de licenciamiento gratuito carecía de elementos como integridad referencial, subconsultas, procedimientos almacenados, no era transaccional. En contraparte el uso de Oracle estaba sobrado.

Como resultado de la elección del manejador de base de datos opté por ASP's para que la conectividad fuera transparente, ambos productos son del mismo fabricante.

3.3 Especificaciones De Los Programas

Describe las pantallas que conforman el sistema, su funcionamiento y las tablas que se afectan.

Mantener los Catálogos

Responsable: Olga Cisneros

Fecha: _____

Se programo un menú en donde se podrán seleccionar las nueve opciones siguientes:

Administración de Capítulos
Administración de Temas
Administración de Exámenes
Administración de Alumnos
Administración de Tipo de Preguntas
Administración de Preguntas, Subpreguntas y Respuestas
Administración de Asignación de Respuestas
Consulta de Historial Académico
Gráfica de Resultados de Exámenes

Tablas

	Insert	Update	Delete	Select
Capítulos	x	x	x	x
Temas	x	x	x	x
Exámenes	x	x	x	x
Alumnos	x	x	x	x
Tipo De Preguntas	x	x	x	x
Preguntas	x	x	x	x
Subpreguntas	x	x	x	x
Respuestas	x	x	x	x
Preguntas_Respuestas	x	x	x	x

- Por medio de los botones se activarán las funciones respectivas de Alta, Baja, Cambio y Consulta.
- Se mantendrá informado al usuario mediante mensajes amigables la respuesta a la acción.
- Si los catálogos tienen información dependiente, se realizará una eliminación lógica

Tutorial		1.1 / 3.4																												
<table border="1"> <thead> <tr> <th>Capítulo</th> <th>Modificar</th> <th>Eliminar</th> </tr> </thead> <tbody> <tr> <td>Bienvenida</td> <td></td> <td></td> </tr> <tr> <td>Teoría</td> <td></td> <td></td> </tr> <tr> <td>Análisis</td> <td></td> <td></td> </tr> <tr> <td>Álgebra Relacional</td> <td></td> <td></td> </tr> <tr> <td>Diseño</td> <td></td> <td></td> </tr> <tr> <td>Construcción</td> <td></td> <td></td> </tr> <tr> <td>Herramientas Case</td> <td></td> <td></td> </tr> <tr> <td>Administración</td> <td></td> <td></td> </tr> </tbody> </table>				Capítulo	Modificar	Eliminar	Bienvenida			Teoría			Análisis			Álgebra Relacional			Diseño			Construcción			Herramientas Case			Administración		
Capítulo	Modificar	Eliminar																												
Bienvenida																														
Teoría																														
Análisis																														
Álgebra Relacional																														
Diseño																														
Construcción																														
Herramientas Case																														
Administración																														
<input type="button" value="Nuevo Capítulo"/>																														

Tutorial		1.2 / 3.4																																											
<table border="1"> <thead> <tr> <th>TEORÍA</th> <th>Mofificar</th> <th>Eliminar</th> </tr> </thead> <tbody> <tr> <td>Cronología de las bases de datos</td> <td></td> <td></td> </tr> <tr> <td>Definición de una base de datos</td> <td></td> <td></td> </tr> <tr> <td>Objetivos de una base de datos</td> <td></td> <td></td> </tr> <tr> <td>Definición de un sistema de base de datos</td> <td></td> <td></td> </tr> <tr> <td>Ventajas y desventajas de un sistema de base de datos</td> <td></td> <td></td> </tr> <tr> <td>Sistema Manejador de Base de Datos (DBMS)</td> <td></td> <td></td> </tr> <tr> <td>Administrador de la base de datos</td> <td></td> <td></td> </tr> <tr> <td>Diccionario de datos</td> <td></td> <td></td> </tr> <tr> <td>Estructura del diccionario de datos</td> <td></td> <td></td> </tr> <tr> <td>Introducción al análisis, diseño y construcción</td> <td></td> <td></td> </tr> <tr> <td>Teoría del modelo de datos Entidad-Relación</td> <td></td> <td></td> </tr> <tr> <th>ANÁLISIS</th> <th>Mofificar</th> <th>Eliminar</th> </tr> <tr> <td>Identificar entidades</td> <td></td> <td></td> </tr> </tbody> </table>				TEORÍA	Mofificar	Eliminar	Cronología de las bases de datos			Definición de una base de datos			Objetivos de una base de datos			Definición de un sistema de base de datos			Ventajas y desventajas de un sistema de base de datos			Sistema Manejador de Base de Datos (DBMS)			Administrador de la base de datos			Diccionario de datos			Estructura del diccionario de datos			Introducción al análisis, diseño y construcción			Teoría del modelo de datos Entidad-Relación			ANÁLISIS	Mofificar	Eliminar	Identificar entidades		
TEORÍA	Mofificar	Eliminar																																											
Cronología de las bases de datos																																													
Definición de una base de datos																																													
Objetivos de una base de datos																																													
Definición de un sistema de base de datos																																													
Ventajas y desventajas de un sistema de base de datos																																													
Sistema Manejador de Base de Datos (DBMS)																																													
Administrador de la base de datos																																													
Diccionario de datos																																													
Estructura del diccionario de datos																																													
Introducción al análisis, diseño y construcción																																													
Teoría del modelo de datos Entidad-Relación																																													
ANÁLISIS	Mofificar	Eliminar																																											
Identificar entidades																																													
<input type="button" value="Nuevo Tema"/>																																													

Tutorial						1.3 / 3.3	
<input type="button" value="Nuevo Examen"/>							

N°	Versión	Descripción	Modificar	Eliminar	Consultar	Examen Contestado
1	1	Examen de teoría y análisis Sem 2004-2			Consultar	Ver
1	2	Examen de teoría y análisis Sem 2005-1			Consultar	Ver
2	1	Examen de Álgebra Relacional			Consultar	Ver
2	2	Examen de Prueba de Algebra			Consultar	Ver
3	1	Examen de diseño			Consultar	Ver
3	2	Examen de diseño de Prueba			Consultar	Ver
4	1	Examen de construcción			Consultar	Ver
5	1	Examen de Herramientas Case			Consultar	Ver

Tutorial						1.4	
<input type="button" value="Nuevo Alumno"/>							

Clave	Nombre	Apellido Paterno	Apellido Materno	Usuario	Contraseña	Modificar	Eliminar
1	Olga	Cisneros	Monroy	ocisneros	sistemas		
2	Donovan	Estrella	Cisneros	destrella	prueba		
3	Cristina	Armenta	Rodriguez	carmenta	9zeb96j		
4	Cinthy	Cisneros	Monroy	ccisneros	3kek03w		
5	Brandon	Montoya	Cisneros	bmontoya	7shb20a		
6	Felipe	Cisneros	Rangel	fcisneros	2kvt13r		
7	Maricela	Monroy	Anaya	mmonroy	6jkk54f		

Tutorial				1.5
	Tipo De Pregunta	Ejemplo	Modificar	Eliminar
	Relacionar Columnas	Ver 		
	Opción Múltiple - 1 respuesta	Ver 		
	Opción Múltiple - 1 ó más respuestas	Ver 		
	Respuesta en texto			
	Generación de diseño (tablas)			
	Álgebra Relacional			
<input type="button" value="Nuevo Tipo de Pregunta"/>				

Tutorial								1.6		
Exam.	Ver.	Capítulo	Tema	Tipo No.		Descripción	Puntos	Imagen	Modificar	Eliminar
1	1	Teoría	Cronología de las bases de datos	1	1	Relaciona cada pregunta con su respuesta correcta.	4			
1	1	Teoría	Cronología de las bases de datos	1	2	Relaciona cada pregunta con su respuesta correcta.	2			
1	1	Teoría	Definición de una base de datos	2	3	¿Cuándo se utiliza barra UID?	2			
<input type="button" value="Nueva Pregunta"/>										

Tutorial					1.7
N°	Pregunta	Asignación	Modificar Asignación	Eliminar Asignación	
1	Relaciona cada pregunta con su respuesta correcta.				
2	Relaciona cada pregunta con su respuesta correcta.				
3	¿Cuándo se utiliza barra UID?				
5	Determina si todos los atributos de la entidad CURSO son atributos. Si no es así seleccionalo(s).				
6	Para el siguiente MER evaluar las reglas de normalización. Identificar si cada atributo está en el lugar correcto; si no lo está, identificar qué regla de normalización se está violando.				
7	¿Qué tipo de relaciones se utilizan para modelar jerarquías?				

Controlar las Evaluaciones del Alumno

Responsable: Olga Cisneros

Fecha: _____

Tablas

	Insert	Update	Delete	Select
Exámenes_Contestados	x			
Respuestas_Alumnos	x			

Para el Control de Evaluaciones cada vez que un alumno conteste un examen, se registrará: la clave del alumno, número y versión del examen con la fecha de realización de manera automática; al igual que la calificación obtenida de cada una de sus respuestas

Reportes

Responsable: Olga Cisneros

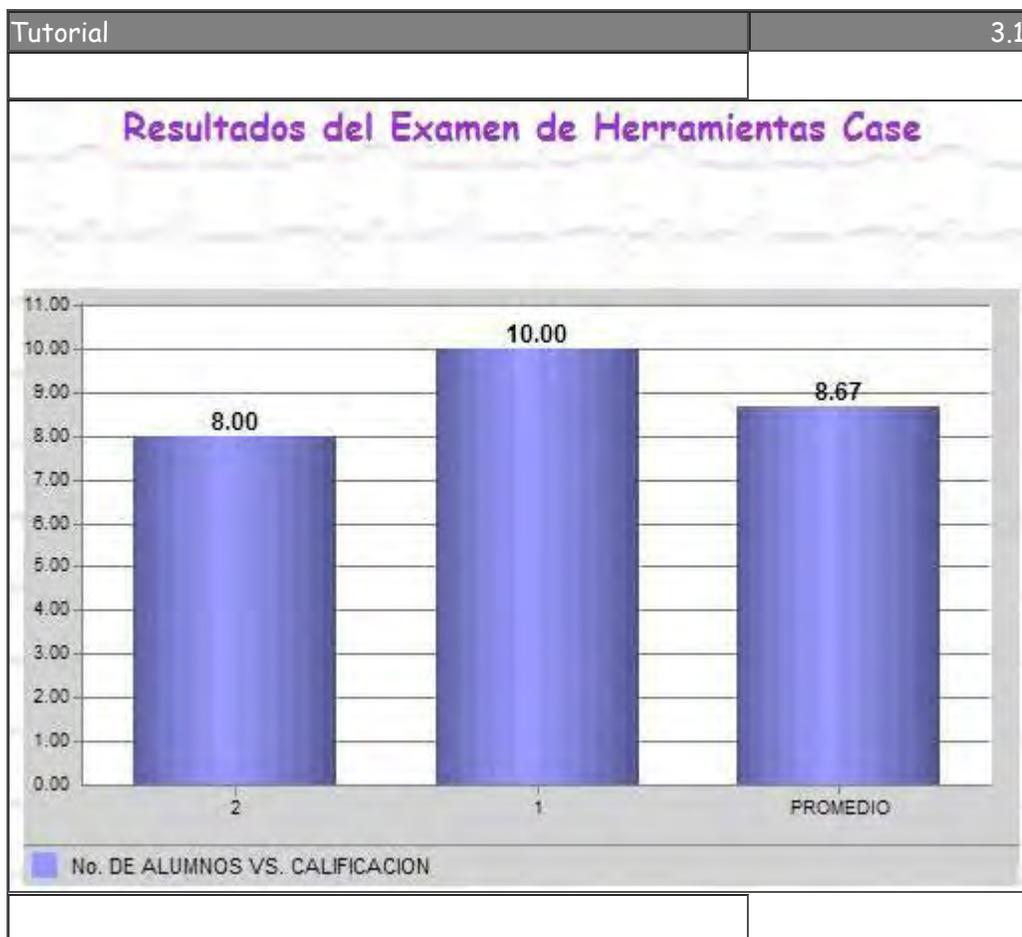
Fecha: _____

Reportar estadísticas de resultados de exámenes y evaluaciones por alumno

Tablas

	Insert	Update	Delete	Select
Exámenes_Contestados				x
Respuestas_Alumnos				x

Se podrá consultar mediante parámetros el historial académico de los alumnos, así como los resultados de los exámenes.



Tutorial	2.1/2.2/3.2	
Consulta de Historial Académico		
Alumno: Brandon Montoya Cisneros		
Descripción	Fecha	Calificación
Examen de Herramientas Case	24-Noviembre-2008	8

IV. Construcción del software educativo interactivo

El software educativo esta comprendido por las siguientes secciones:

- Registro: Cada alumno debe proporcionar su nombre y apellidos para la asignación de usuario y contraseña. Existe un usuario tipo administrador el cual se agrego directamente a las tablas.
- Acceso: El alumno ó administrador debe proporcionar usuario y contraseña para entrar al sistema.
- Administración: Esta sección solo la visualiza el profesor (usuario tipo administrador) y puede realizar consultas, altas, bajas y/o modificaciones de :
 - ❑ Capítulos
 - ❑ Temas
 - ❑ Exámenes
 - ❑ Alumnos
 - ❑ Tipos de preguntas
 - ❑ Preguntas, Subpreguntas, Respuestas
 - ❑ Asignar respuestas
 - ❑ Consulta de historiales académicos
 - ❑ Visualización de las gráficas de resultados de exámenes
- Consulta de Historial académico. En esta sección el alumno puede visualizar los exámenes que ha contestado con sus respectivas evaluaciones, así como la fecha en la que resolvió el examen.
- Exámenes. En esta sección se muestran todos los exámenes disponibles que el alumno puede resolver. El alumno puede contestar cada examen una sola vez al día.
- Contenido: Los capítulos que se desarrollaron en el software educativo son:
 - Teoría. Trata sobre la historia y principales definiciones de las bases de datos
 - Análisis. Nos enseña como modelar los requerimientos del cliente a través de entidades, relaciones, atributos, etc.
 - Álgebra relacional. Nos explica como utilizar éste lenguaje para obtener información de la base de datos.
 - Diseño. Nos muestra como trasladar nuestros modelos a tablas normalizadas, llaves primarias, foráneas, índices, vistas, etc.
 - Construcción. Nos orienta a través de los lenguajes DDL, DCL y DML como crear y consultar la base de datos, así como otorgar permisos para usar o modificar los datos.
 - Herramientas Case. Presenta las etapas de esta metodología para el desarrollo de sistemas.

En cada uno de estos capítulos se presenta un examen para que el alumno se auto-evalúe.

Visualización del Home del software educativo (Perfil alumno)

Curso de Base de Datos en Línea - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost/Tutorial/Home.asp> Ir Vínculos >>

[Teoría](#) [Análisis](#) [Algebra Relacional](#) [Diseño](#) [Construcción](#) [Herramientas Case](#) [Historial](#) [Exámenes](#)

II. Teoría

- Cronología de las bases de datos
- Definición de una base de datos
- Objetivos de una base de datos
- Definición de un sistema de base de datos
- Ventajas y desventajas de un sistema de base de datos
- Sistema Manejador de Base de Datos (DBMS)
- Administrador de la base de datos
- Diccionario de datos
- Estructura del diccionario de datos
- Introducción al análisis, diseño y construcción
- Teoría del modelo de datos Entidad-Relación
- Examen

Definición De Base De Datos

Conjunto de información organizada (Archivo lógico (uno o varios archivos físicos))

Objetivos De Una Base De Datos

Servir a una aplicación o más, de la mejor manera posible.

Niveles en que puede observarse la base de datos, esto se conoce como Abstracción de la información:

- Nivel Físico. Describe como se almacenan los datos (bytes) DBA = Administrador de la Base de Datos
- Nivel Conceptual. Esquematización de los datos y sus relaciones (diagramas) Analista y Desarrollador
- Nivel De Visión. Abstracción más alta, como interpretan la información los usuarios finales.

[Salir](#)

Visualización del Home del software educativo (Perfil administrador)

Curso de Base de Datos en Línea - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost/Tutorial/Home.asp> Ir Vínculos >>

Teoría Análisis Álgebra Relacional Diseño Construcción Herramientas Case Administración

Capítulos Existentes

Capítulo	Modificar	Eliminar
Bienvenida		
Teoría		
Análisis		
Álgebra Relacional		
Diseño		
Construcción		
Herramientas Case		

Nuevo Capítulo

Salir

- Administración de Capítulos
- Administración de Temas
- Administración de Exámenes
- Administración de Alumnos
- Administración de Tipo de Preguntas
- Administración de Preguntas, Subpreguntas y Respuestas
- Administración de Asignación de Respuestas
- Consulta de Historial Académico
- Gráfica de Resultados de Exámenes

TEORIA DE LOS CAPITULOS DEL SOFTWARE EDUCATIVO

a) BIENVENIDA

La presente tesis está diseñada para auxiliar en el proceso enseñanza-aprendizaje, al docente que imparte la materia de Base de Datos y a los estudiantes que cursan esta materia.

Surge por la necesidad de brindar nuevas herramientas didácticas en esta área del conocimiento, siendo un instrumento interactivo de apoyo en la labor educativa.

Objetivos Generales

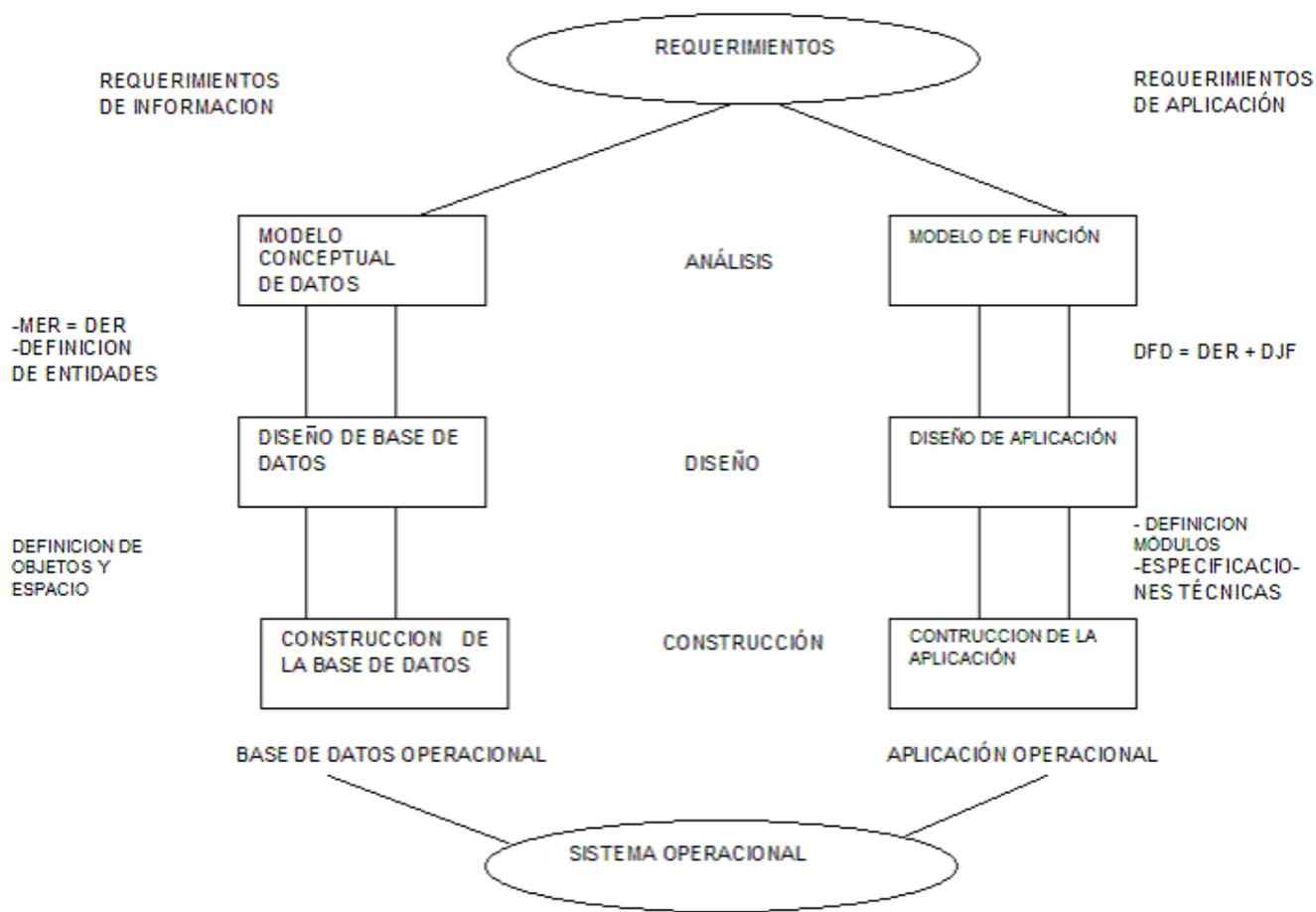
Los profesores se podrán apoyar de este instrumento para complementar su actividad educativa, enriqueciendo su método de enseñanza.

Los alumnos reforzarán sus conocimientos de la materia, a través de los diversos ejemplos y ejercicios propuestos en el curso.

b) HERRAMIENTAS CASE

- *Metodología Case*

El desarrollo de base de datos es un enfoque top-down (descendiente), que transforma los requerimientos de información en una base de datos operacional como se muestra en la siguiente figura.



Dado que el desarrollo de sistemas de información es una actividad compleja, ésta puede dividirse para su estudio en las siguientes etapas:

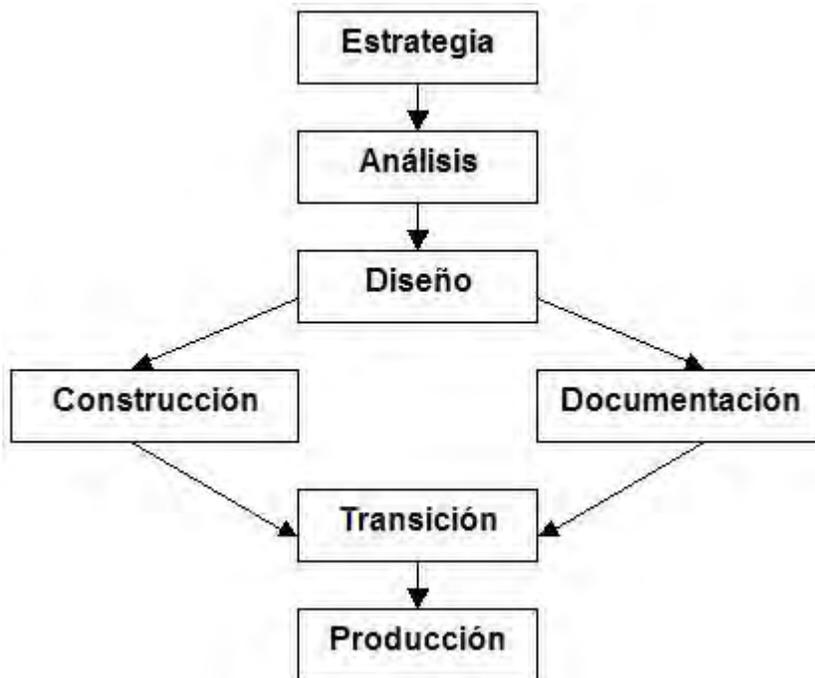
1. Definición y análisis de los requerimientos del usuario.
2. Diseño del sistema y de la base de datos.
3. Implantación y prueba de módulos.
4. Integración y prueba del sistema.
5. Operación y mantenimiento.

Como estas etapas a su vez son muy elaboradas, han surgido varias metodologías que permiten realizarlas de una manera estructurada. El método CASE (Computer Aided Systems Engineering/ Ingeniería de Sistemas Asistida por Computadora/ Ingeniería de Sistemas Automatizada) plantea una secuencia de etapas que es aproximadamente equivalente a la citada arriba, pero es más detallada, y además proporciona para cada etapa su descripción, definición de objetivos y metas, productos de la etapa, factores críticos de éxito, y la lista de tareas que conviene realizar. Además es posible auxiliarse de herramientas CASE que facilitan grandemente la puesta en práctica del método. A continuación presentamos la metodología CASE.

Etapas en el método CASE

El ciclo de vida de un sistema se compone de las siguientes etapas 1, 2 y 3:

1. Estrategia
2. Análisis
3. Diseño
- 4.1. Construcción
- 4.2. Documentación
5. Transición
6. Producción

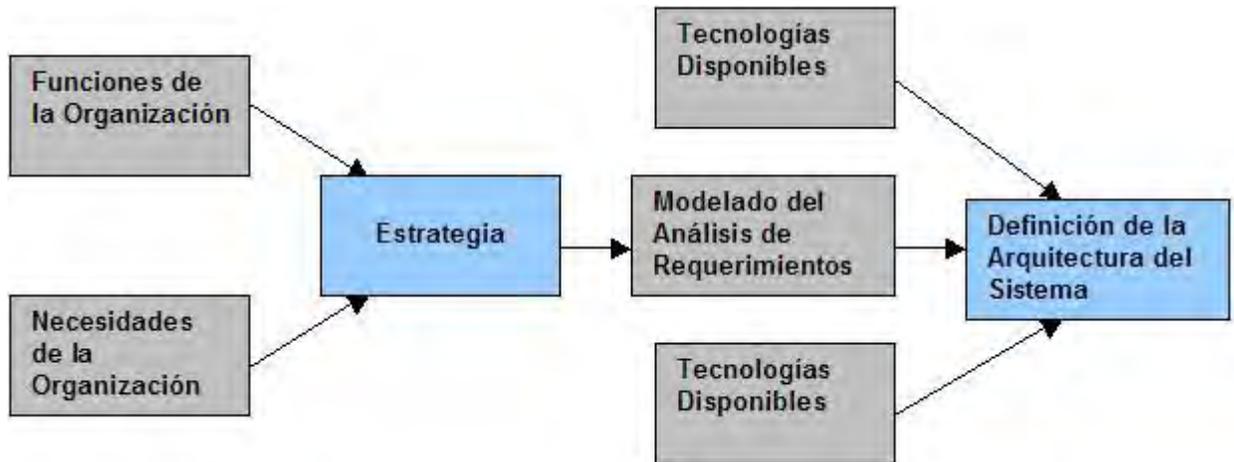


ETAPAS DE LA METODOLOGÍA CASE

- *Fase de estrategia*

Esta es una de las etapas mas importantes, ya que tiene por objetivo lograr un entendimiento claro de las necesidades de la organización y del ambiente en que operará el sistema o sistemas a implantar.

Los resultados de esta etapa son, un conjunto de modelos de la empresa, un conjunto de recomendaciones, y un plan acordado de desarrollo de los sistemas de información, la elaboración de este último se hará de acuerdo las necesidades actuales y futuras de la organización, tomando en cuenta restricciones operativas, financieras y técnicas.



ETAPA DE ESTRATEGIA

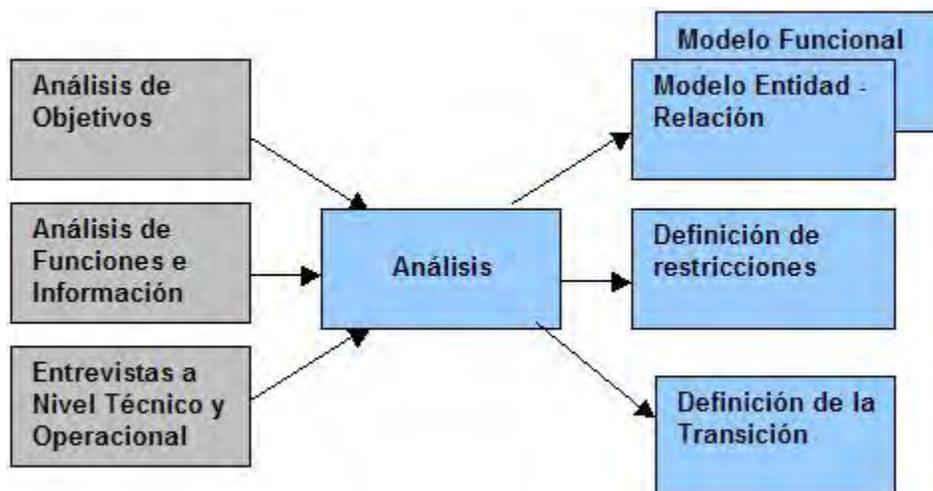
- *Fase de análisis*

La etapa de análisis toma y verifica los descubrimientos de la etapa de estrategia; además de que realiza otra serie de entrevistas ya no a un nivel directivo como en la anterior, sino a un nivel operativo y técnico. Con la participación de los responsables de la operación de las funciones que serán automatizadas se realiza un análisis detallado de sus requerimientos específicos en cuanto a objetivos, sub-funciones, información, datos, etc.

Así, en esta etapa a partir de los modelos de la organización obtenidos en la anterior y del producto del análisis de ésta, se genera el modelado del sistema. Los modelos básicos de esta etapa son:

- El Modelo entidad-relación, que modela mediante relaciones lógicas todos los datos involucrados en el sistema, de tal manera que cualquier tipo de explotación (consulta o modificación) sean posibles.
- El Modelo funcional, que modela los diferentes servicios que ofrecerá el sistema mediante una organización y clasificación de las diversas funciones y subfunciones que fueron identificadas en el análisis.

Como resultados de esta etapa, además del modelo de entidad-relación y el funcional, se definen las restricciones que tendrá el sistema y la estrategia que se seguirá en la etapa de transición.

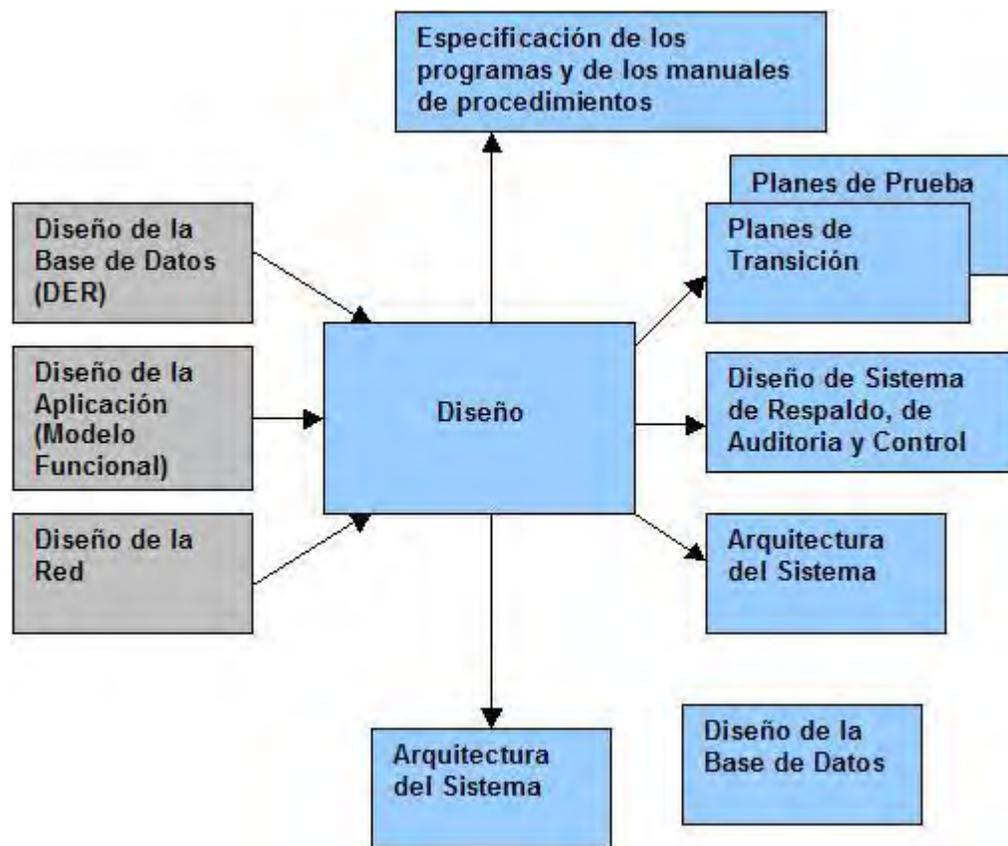


ETAPA DE ANÁLISIS

- *Fase de diseño*

La etapa de diseño toma los requerimientos y el modelado de la etapa de análisis y determina la mejor manera de satisfacerlos, logrando niveles de servicios acordados, dados el ambiente técnico y las decisiones previas en los niveles requeridos de automatización. Es decir que del diseño conceptual se pasa al diseño final que será utilizado para la implantación, por ejemplo en esta etapa, el modelo entidad-relación será transformado en un diseño de base de datos, y en especificaciones de almacenamiento y el modelo de funcional, en módulos y manuales de procedimientos.

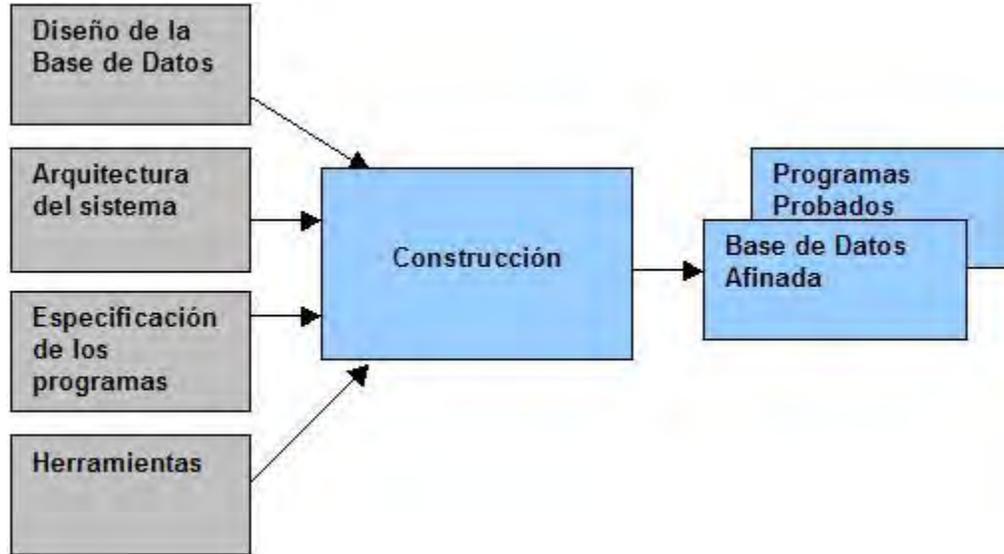
El diseño final del sistema integra tres diseños, el de la base de datos, el de la aplicación y el de la red además se elaboran los planes de prueba y de transición y se realizan los diseños de los sistemas de auditoría y control, y el de respaldos y recuperación. Los resultados de esta etapa lo constituyen, la arquitectura del sistema, el diseño de la base de datos, la especificación de los programas, la especificación de los manuales de procedimientos. En la figura 4 se muestra esta etapa.



ETAPA DE DISEÑO

- *Construcción*

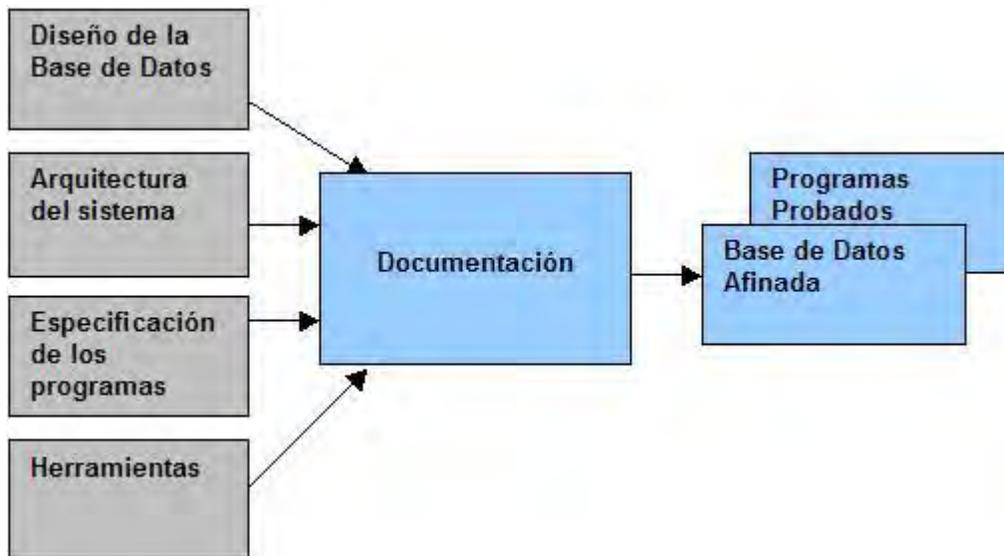
A partir del diseño final generado en la etapa anterior, en esta, se codificarán y probarán los nuevos programas, usando herramientas apropiadas. Los resultados de la construcción son los programas probados y la base de datos afinada.



ETAPA DE CONSTRUCCIÓN

- *Documentación*

Uno de los productos fundamentales para un uso y un mantenimiento efectivo y eficiente de los sistemas programados son los manuales. Esta etapa se realiza al mismo tiempo que la de construcción. Los manuales, resultados de esta etapa, se elaboran a partir de las especificaciones de diseño y de los programas realizados.

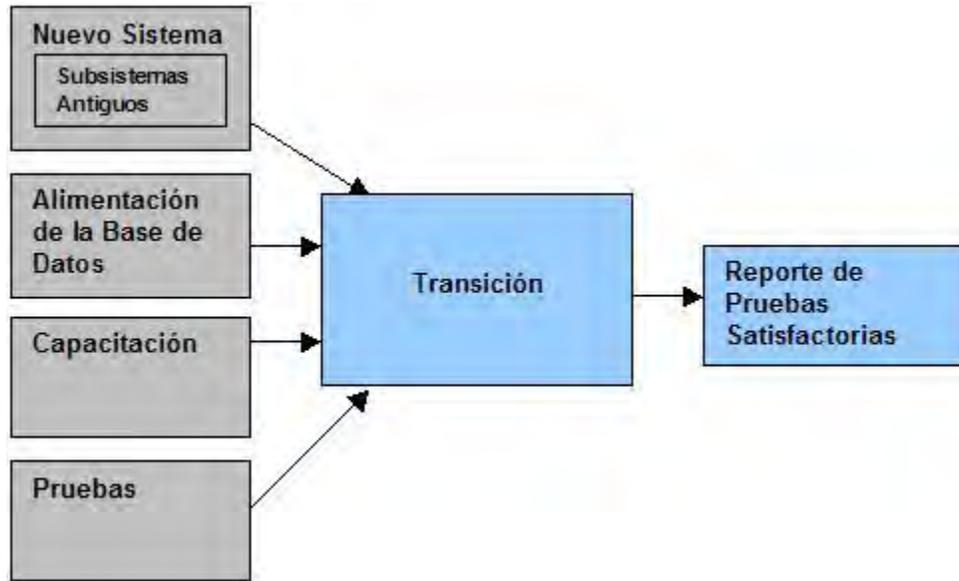


ETAPA DE DOCUMENTACIÓN

- *Transición*

La introducción ya sea de un sistema completamente nuevo o un sistema que integra subsistemas existentes implica un nuevo tipo de uso y de operación que deberá ser asimilado y aprendido por los usuarios y operadores. Por esta razón, el desarrollo de un sistema no se termina con su programación; antes de su liberación para su uso, se debe prever un período de transición que deberá incluir la alimentación de la nuevas bases de datos, la capacitación de los usuarios y el desarrollo de pruebas.

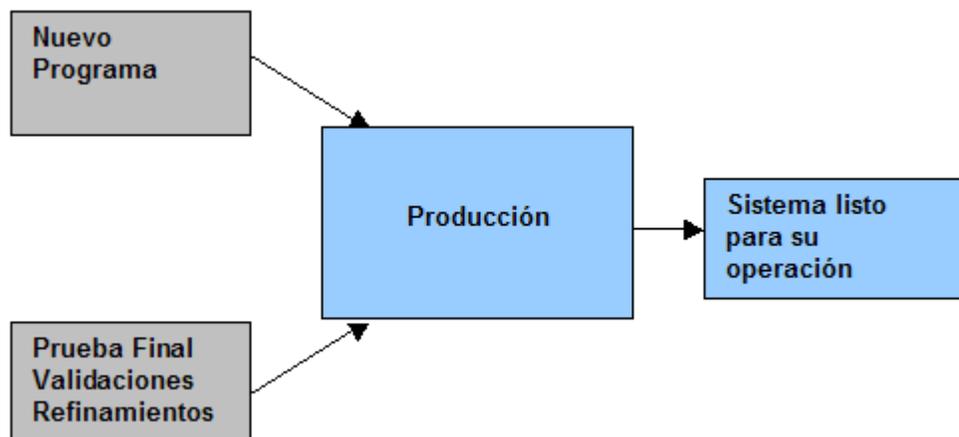
La transición debe llevarse a cabo con una interrupción mínima de la organización, y debe dejar a los usuarios confiados y listos para explotar el nuevo sistema. El resultado final de esta etapa es un reporte que muestre que las pruebas fueron satisfactorias.



ETAPA DE TRANSICIÓN

- *Producción*

Finalmente, en la etapa de producción se asegura que el sistema funcione correctamente en la mayoría de los casos, y con intervención mínima de los administradores del sistema. Para esto se realizan nuevas pruebas, se reevalúan los resultados y se hacen refinamientos del sistema, los cambios necesarios deberán ser introducidos sin afectar a los usuarios, y deberá conseguirse la máxima confianza de los usuarios. El resultado de esta etapa un sistema listo para su operación.



ETAPA DE PRODUCCIÓN

c) TEORÍA DE BASE DE DATOS

- *Cronología de las bases de datos*

Antes de existir las Bases de Datos existieron los sistemas de procesamiento de archivos basados en registros guardados en archivos, creándose respectivamente programas para su manipulación (obtener y guardar información).

Estos tenían varias desventajas:

- Redundancia de información (repetición de información).
- Inconsistencia en la información (problemático actualizarla, por tanto las distintas copias no concuerdan).
- Falta de manejo de concurrencia (múltiples usuarios con actualizaciones al mismo tiempo).
- Dependencia entre programa con archivo.

Breve Historia De Las Bases De Datos

- 1945 – 1960 Surge el Lenguaje Cobol, Surgen archivos de acceso secuencia, archivos de acceso directo. Surge concepto de llave (Key).
- 1960 – 1965 Evoluciona el Modelo Jerárquico a partir del lenguaje Cobol.
- 1968 Evoluciona el Modelo de Red.
- 1980 Evoluciona el Modelo Relacional.

Cronología de las bases de datos

1960	Uso de LIS archivos separados ISAM (Index Sequential Access Method) método para almacenar información a la que se puede acceder rápidamente, fue reemplazado por VSAM (Virtual Storage Access Method) que es un sistema de archivos usado en mainframes de IBM en la actualidad.
1964	El término Bases De Datos, primero usado en las fuerzas militares de EEUU aparece en las publicaciones Honeywell introduce IDS (Almacén De Datos Integrado).
1969	IBM introduce IMS, una base de datos jerárquica (con el fin de programas de conquista espacial, permite construir cadenas de registros entre ficheros y recorre dichas cadenas).
1970	E.F. Codd hizo un documento describiendo modelo relacional (definición de modelo relacional) basado en la simplicidad matemática del álgebra relacional.
1971	Se produjo un informe (DBTG) sugiriendo por primera vez dos lenguajes de descripción de datos, uno para el esquema y otro para el subesquema. Se forma una comisión llamada Comisión de Lenguajes de Descripción de Datos (DDLDC).
1972	Finales de 1971 y principios de este año aparece el primer borrador del modelo Codasyl con una arquitectura de dos niveles: un esquema que proporciona la vista del sistema y un subesquema que proporciona la vista del usuario.
1973	La DDLDC publicó su propuesta en Journal of Development, reconociendo que el proceso de desarrollo en lenguajes de bases de datos debería ser evolutivo, como en el Cobol.
1976	DER Chen. Surge el SQL. Chen introduce el modelo entidad – vínculo (ER)
1977	Oracle Corp. se establece. Surge QUEL (Query Language) del sistema INGRES.
1978	Oracle hace la primera implementación real del modelo relacional. Surge el lenguaje QBE (Query By Example) desarrollado por IBM. La comisión técnica X3H2 ha comenzado a estandarizar el modelo de red. Se lanzó una definición borrador de un esquema de almacenamiento pero jamás ha sido aprobada ninguna versión final.

1979	Primer RDBMS comercial Oracle V2.
1980	Se funda informix (3ª generación de DBMS). Cerca de 10,000 usando DBMS en EEUU. Para PC Paradox. Evolución del modelo relacional.
1981	IBM comenzó a comercializar el SQL. La pequeña compañía Ashton – Tate introduce DBASE II (no es DBMS, es de archivos).Las facilidades de Codasyl incluyen un Lenguaje de Descripción de Datos (DDL) de subesquema y un correspondiente Lenguaje de Manipulación de Datos (DML) Cobol, esto es, el Cobol JoD de 1981.
1982	Inician los esfuerzos de ANSI SQL.
1983	Oracle libera sus primero DBMS que corren en mainframes, minicomputadoras y PC's. IBM lanzó al mercado su producto DATABASE 2, más conocido por su abreviatura DB2.Se disolvió la principal comisión Codasyl que se ocupaba del modelo porque terminó su objetivo al crear un modelo con éxito, que ANSI tomó a su cargo para estandarizar.
1984	Ashton – Tate introduce DBASE III. El trabajo sobre el modelo de red (ANSI, 1984 a, b, c, d); basado en el modelo Codasyl, está casi completo, mientras que el modelo relacional (ANSI/SPARC 1984) está todavía en la etapa inicial (Gallagher, 1984).
1985	Comenzaron a conectar las computadoras por medio de redes de área local (LAN). IBM introduce DB2 para maxicomputadoras o mainframes bajo Sistema Operativo MVS.
1986	Informix pasa a ser una compañía pública. Oracle libera su primer DBMS con capacidades de distribución. Se publica el estándar ANSI SQL. El Dr. Date define las Base de Datos Distribuidas. SQL fue adoptado como estándar ANSI (American National Standards Institute) para los lenguajes de Base de Datos Relacionales. Los lenguajes usados son: SQL (Structure Query Language), QUEL (Query Language) y QBE (Query By Example).Base de Datos orientadas a objetos. Primera generación Gbase francesa Grápale. Surge la primera generación SGBDOO (Sistema Gobernador de Base de Datos Orientada a Objetos).
1987	ISO (International Standards Organization) adopta el estándar SQL.Serv. Corp. Introdujo Gemstone.
1988	Se libera SQL Server Para OS/2 por parte de Microsoft y Sybase. Libera Oracle V6.Ashton – Tate introduce DBASE IV. Ontologic lanzó Vbase. Symbolics introdujo Static (apoyo a inteligencia artificial como LISP - Lenguaje de Procesamiento de Listas).
1989	ANSI – 89 SQL. El estándar SQL fue actualizado. Se forma SQL Access Group. Se utiliza arquitectura cliente/servidor y plataforma común. Segunda etapa del SGBDOO.
1990	Ashton – Tate introduce DBASE IV ver 1.1.Tercera generación de Base de Datos Orientada a Objetos (integridad en forma más rápido y simple), sobre todo en áreas de investigación. Programación Orientada a Objetos simula – 67 y smalltalk – 80.
1991	Modelo Orientado a Objetos (Autores: Jeffcoate, Guilfoyle y Deutsch)Oracle alcanza el poder de 1,000 TPS en una máquina de cómputo paralelo. Las bases de objetos (también conocidas como bases de datos orientadas a objetos o bases de datos objetuales).El ODMG (Object Database Management Group) hoy día (Object Data Management).El uso creciente del Unified Modeling Language (UML), para la especificación y diseño de sistemas de información propicia que sus datos se gestionen en bases de objetos de una manera más natural – tecnología de objetos del principio al final – que en SGBDs.
1992	ANSI – 92 SQL. Triggers (Disparador: acciones ejecutadas automáticamente).SQL Access Group publica las especificaciones CLIENTE. Microsoft libera ODBC 1.0 para Windows. Se renueva el estándar para SQL, ANSI – 92 SQL.

1993	Oracle implanta base de datos distribuidas. Access puede trabajar con datos dependientes de DBASE, Paradox, FoxPro).Paradox 4.0 de Borland anuncia PAL (Paradox Applications Language for Windows).Sybase anunció la versión 10.0, una versión completamente nueva de SQL Server y sin intervención de Microsoft. Libera Oracle 7 para UNIX. Se desarrolla el Ambiente de Desarrollo Cooperativo (CDE) de Oracle. Microsoft libera su versión de SQL Server de 32 para Windows NT.
1994	FoxPro Microsoft. Microsoft libera ODBC 2.0 para Windows. Libera Oracle 7 para PC: En Abril Microsoft anunció que desarrollaría su propio producto servidor de base de datos. Las versiones posteriores de Microsoft SQL Server traían recuerdos de SQL Server de Sybase. Microsoft permite portar las especificaciones de ODBC a otras plataformas no Windows por parte de Visigenic Software. ODBC 2.0 SDK es disponible para los UNIX de Sun, HP e IBM.
1995	Se anuncia DB2/2 de IBM. Oracle 7.1SQL Server 6.0 de Microsoft (puede trabajar con Access, Lotus Approach y Borland Paradox).En este año Informix tenía el 34.5% del mercado RDBMS para UNIX. El SDK de ODBC está disponible para SCO UNIX, OS/2, Macintosh y Power Macintosh.
1996	Mejor base de datos Oracle 7.2Oracle Parallel Server incrementa performance introduciendo operaciones en paralelo y asíncronas. Oracle Universal Server transfiere desde PC a poderosos servidores de redes. Soporta Web, multimedia. Oracle 8 está en versiones beta. Sybase e Informix. JavaSoft está preparando una especificación para unir aplicaciones Java con bases de datos relacionales. La subsidiaria de Sun Microsystems ha anunciado Java Database Connection. Sun Microsystems dispone de una herramienta que permite unir aplicaciones Java a base de datos ODBC. Java Database Connectivity (JDBC) crea un cambio uniforme para que las aplicaciones Java accedan a diferentes bases de datos SQL.
1997	Herramientas OLAP (On Line Analytical Processing) ventajas de base de datos en bloques de tres dimensiones. Oracle se encuentra en la versión 8.1, SQL Server de Sybase en la 11 y SQL Server de Microsoft en la 6.5. Hoy en día las aplicaciones multimedia son vía Internet o Intranet para base de datos y envuelven un gran número de redes públicas o privadas. En los últimos años se han introducido nuevos conceptos en las bases de datos como: Datawarehousing (base de datos formada por base de datos que pueden manejar una cantidad enorme de información).Base de Datos Orientadas a Objetos (En un inicio se tenían implementaciones de éstas que no cumplían del todo con la ideología de objetos, pero con el tiempo se han ido acoplando a ésta).JBDC (Java Database Connectivity) este es un API (Interfaz de Aplicación) que une a Java con las bases de datos. Demanda para aprovechar Procesadores Paralelos Masivos (MPP).Las bases de datos FoxPro con 24%, Access con un 21%, Oracle con un 15% y FileMaker 6%, son las más utilizadas entre los encuestados para poder administrar la información sobre biodiversidad. Así mismo un 15% de las instituciones que participaron en la encuesta no cuentan con ninguna base de datos para administrar su información. IBM, Informix, Microsoft, Oracle y Sybase, anuncian la creación del middleware necesario para introducir y extraer datos en y desde sus DBMS mediante tecnología Web. Informix se alianza con Netscape, dando lugar a la integración de su Online Workgroup Server. Sybase Iberia organiza un curso sobre diseño de bases de datos relacionales. Sybase integra colas de mensajes en sistemas de bases de datos con el lanzamiento del sistema dbQ. Las bases de datos relacionales de Sybase ofrecerán soporte a Java. Lanza su arquitectura de componentes Adaptive. Informix reorienta su estrategia y unifica sus cuatro bases de datos. La compañía intenta recuperar su rentabilidad. Bases de datos relacionales extendidas. Las bases de datos relacionales y sus SGBDS's, por su parte, han ido evolucionando esto últimos años para soportar objetos y reglas y para ampliar el lenguaje SQL y hacerlo extensible – nuevos tipos, nuevas operaciones – y computacionalmente completo. El soporte de los Persistent Stored Modules es una mejora importante, tanto para la productividad del desarrollo de aplicaciones y la seguridad, como para la eficiencia en entornos cliente/servidor. El

	<p>desarrollo del nuevo estándar de SQL ha sido más largo de lo previsto y no sólo por lo ambicioso de sus objetivos o como consecuencia directa de los intentos de convergencia con el modelo ODMG y su lenguaje OQL. También aquí, además de esta colaboración en el desarrollo del nuevo estándar, ha habido una fuerte competencia, sobre todo entre los tres grandes: DB2 de IBM con sus extenders, Informix con sus datablades y Oracle con sus cartridges. Cabe destacar que el impacto de Java da lugar a JDBC y al SQLJ. Minería de datos. Análogamente como en minería se obtienen pequeñas cantidades de metal por beneficio de grandes cantidades de mena. Integra técnicas no sólo de base de datos, sino también de la estadística y de la inteligencia artificial. Ha habido progresos recientes en base de datos activas y de tiempo real, en bases de datos espaciales, temporales y espacio-temporales, en bases de datos con información incompleta o incierta y en bibliotecas digitales, entre otras técnicas y han aparecido las bases de datos móviles o nómadas. Las nuevas técnicas son tratadas en libros como [Zaniolo97] y [Yu98].</p>
1998	<p>Microsoft lanzó la versión 7.0 de SQL Server con una reestructura de ingeniería de base de datos</p>
1999	<p>Sybase lanzó al mercado SQL Anywhere Studio versión 6.0, un paquete completo para la gestión de datos, para uso en ambientes móviles, de escritorio y de trabajo en grupo. Este sistema incluía el Sistema administrador de base de datos relacionales SQL Anywhere, pero rebautizado como Adaptive Server Anywhere (ASA) y un conjunto de herramientas para el diseño y gestión de base de datos, creación de reportes y formas, uso de base de datos en la Web y la replicación de base de datos a usuarios móviles. Al mismo tiempo lanzó ASE 12.0 brindando soporte para Java en la base de datos, alta disponibilidad y gestión de transacciones distribuidas</p>
2001	<p>IBM compró a Informix la tecnología de base de datos. Borland libera Interbase 6 bajo una variante de la Mozilla Public License. Sybase lanzó Adaptive Server Enterprise (ASE) 12.5, con características tales como Aplicación dinámica de memoria, soporte para XML en la base de datos y conexiones seguras con SSL.</p>
2002	<p>A finales de este año Borland lanzó InterBase versión 7, incluyendo soporte para SMP, soporte mejorado para aplicación y controlar los servidores. Se lanza Oracle 9i con un enfoque e-business</p>
2005	<p>A mediados de este año dos compañías anunciaron planes para comercializar PostgreSQL con énfasis en nichos separados de mercados. EnterpriseDB añadió funcionalidades que le permitían a las aplicaciones escritas para trabajar con Oracle ser más fáciles de ejecutar con PostgreSQL. Greenplum contribuyó mejoras directamente orientadas a Aplicaciones de Data Warehouse e Inteligencia de negocios, incluyendo el proyecto BizGres. En noviembre de 2005, Sun Solaris 10 (lanzamiento 6/06) incluía PostgreSQL. Borland lanza el 1º de junio la versión 7.5.1 de Interbase</p>
2006	<p>En noviembre junto al lanzamiento de SQL Server 2005, Microsoft lanza la versión 3.0 renombrada a SQL Server Mobile 3.0 con acceso simultáneo, soporte transaccional.</p>
2007	<p>EnterpriseDB anunció el Postgres Resource Center y EnterpriseDB Postgres, diseñados para ser una completamente configurada distribución de PostgreSQL incluyendo muchos módulos contribuidos y agregados. Oracle lanza la versión estable 11g Release 1</p>
2008	<p>El 18 de abril de 2008 se libera la versión 2.1 de Firebird, un sistema de administración de base de datos relacional de código abierto, basado en la versión 6 de Interbase.</p>
2009	<p>Desde abril una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desarrolla MySQL como software libre en un esquema de licenciamiento dual</p>

- *Definición de una base de datos*

Conjunto de información organizada (Archivo lógico (uno o varios archivos físicos))

- *Objetivos de una base de datos*

Servir a una aplicación o más, de la mejor manera posible.

Niveles en que puede observarse la base de datos, esto se conoce como Abstracción de la información:

- Nivel Físico. Describe como se almacenan los datos (bytes) DBA = Administrador de la Base de Datos
- Nivel Conceptual. Esquematización de los datos y sus relaciones (diagramas) Analista y Desarrollador
- Nivel De Visión. Abstracción más alta, como interpretan la información los usuarios finales.

- *Definición de un sistema de base de datos*

Servir a una aplicación o más, de la mejor manera posible.

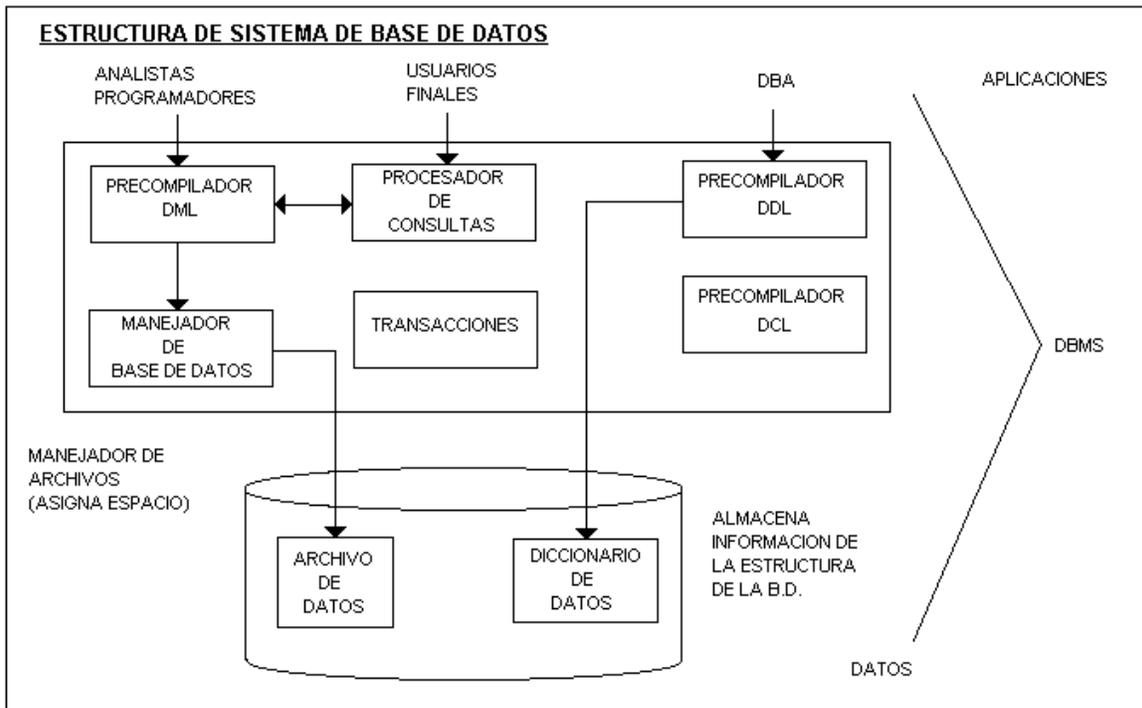
- *Ventajas de un sistema de base de datos*

- Reducir redundancia
- Evitar inconsistencia
- Mantener integridad
- Controlar concurrencia
- Aplicar restricciones de seguridad
- Los datos son independientes de los programas

- *Desventajas de un sistema de base de datos*

- La seguridad y la integridad pueden ser contraproducentes sin buenos controles

- *Sistema Manejador de Base de Datos (DBMS)*



El Sistema Manejador de Base de Datos (DBMS *DataBase Management System*) está formado por:

1. Base de Datos
2. Software para manipular los datos (SQL)

Funciones:

- Almacena, recupera, elimina y modifica datos
- Guarda la consistencia de los datos
- Soluciona problemas de concurrencia
- Tiene seguridad
- Creación y modificación de la base de datos

Objetivo: Crear un ambiente en que sea posible guardar y recuperar información de la base de datos en forma eficiente.

El manejo de datos del DBMS incluye tanto la definición como la manipulación y la seguridad de los mismos:

- Lenguaje de Definición de Datos (DDL = Data Definition Language). Define y maneja los objetos de la base de datos
- Lenguaje de Manipulación de Datos (DML = Data Manipulation Language). Manipulación: Inserción, Recuperación, Eliminación y Actualización (Lenguajes de consulta SQL "Structured Query Language", PAL, XBASE).
- Lenguaje de Control de Datos (DCL = Data Control Language). Controla accesos a la base de datos.

- *Administrador de la base de datos*

Administrador De La Base De Datos (DBA = DataBase Administrator)

Sus actividades son:

- Definir y modificar la estructura de almacenamiento (espacio).
- Definir permisos de accesos (DCL).
- Definir estrategia de respaldo y recuperación en caso de fallas.
- Afinar la Base de Datos

- *Diccionario de datos*

Conjunto central de tablas y vistas de la base de datos de sólo lectura; es la base de datos maestra.

Un diccionario de datos proporciona la siguiente información:

- Nombre de los usuarios.
- Privilegios (permisos) que tienen los usuarios.
- Nombres de los objetos (Tablas, Vistas, Índices, Sinónimos, Clusters, Secuencias, Procedimientos, Triggers, tablespace,...).
- Defaults.
- Espacio ocupado por los objetos

- *Estructura del diccionario de datos*

Esta compuesto de:

- Tablas Base: En las cuales escribe únicamente el DBMS
- User accesible: Vistas accesibles de los objetos de los usuarios Views: Vistas accesibles.
- El dueño del diccionario es SYS (superusuario) Los prefijos de estas tablas y vistas son: all_ dba_ usr_

Por ejemplo:

```
all_tables
dba_tables
usr_tables
```

El DBMS accede al diccionario de datos cada vez que se ejecuta un DDL, DCL o DML.

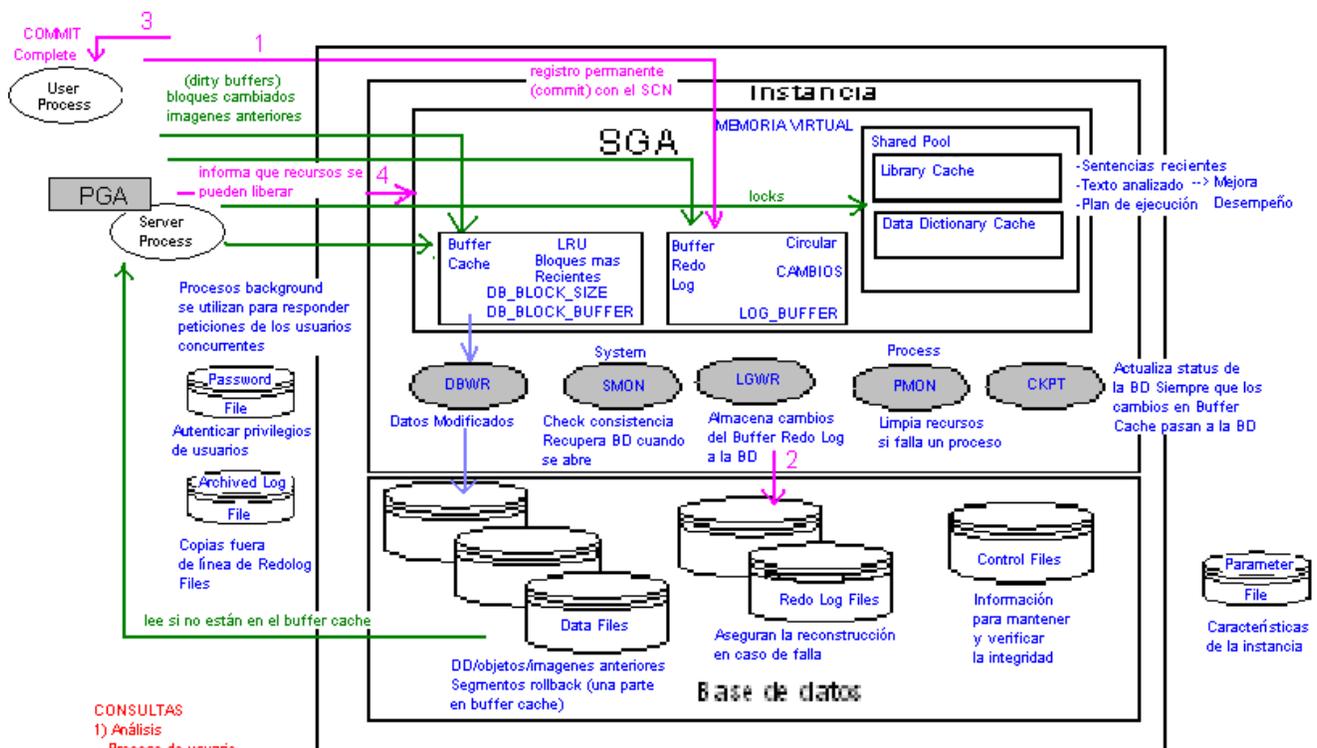
¿Cómo se usa el diccionario de datos?

Durante la operación de la base de datos el DBMS lee el diccionario de datos para ver que objetos hay y quién los accede y para actualizar estructuras del desarrollador.

Ejemplo: Si el usuario X crea una tabla; se crean registros de la nueva tabla en el diccionario de datos, sus columnas, sus segmentos, extents y privilegios que el usuario X tiene en la tabla.

Acceso Rápido Al Diccionario De Datos

Debido a que el diccionario es accesado por el DBMS constantemente (durante la operación de la base de datos para verificar el estado de los objetos) mucha información del diccionario se encuentra en el SGA (Shared Global Area). Toda la información almacenada en memoria usando el algoritmo LRU (Least Recently Used, Menos Reciente Uso).



CONSULTAS
 1) Análisis
 Proceso de usuario manda consulta al proceso servidor y usa el SGA para compilar -> éxito o no
 2) Ejecución
 El proceso del servidor se prepara para recuperar los datos
 3) Obtención
 Los renglones son regresados

DML
 1) =
 2) **COMMIT**
 en caso de falla. El servidor asigna un commit SCN a la transacción para sincronizar datos de los data files

Nunca se crean nuevos objetos perteneciendo al usuario SYS, excepto algún script (conjunto de comandos guardado es un archivo *.sql) provisto por la empresa proveedora del DBMS.

Única excepción para borrar información en una tabla es SYS.AUD\$ por crecer tanto.

¿Cómo acceden los usuarios al diccionario?

El diccionario reside en el tablespace SYSTEM el cual siempre está en línea cuando la base de datos esta abierta (activa).

- *Introducción al análisis, diseño y construcción*

En ANÁLISIS de los requerimientos de información del usuario se desarrolla un Modelo Entidad-Relación para expresar esos requerimientos (Modelar, Esquematar). Los requerimientos son proporcionados por el usuario.

Ejemplo:

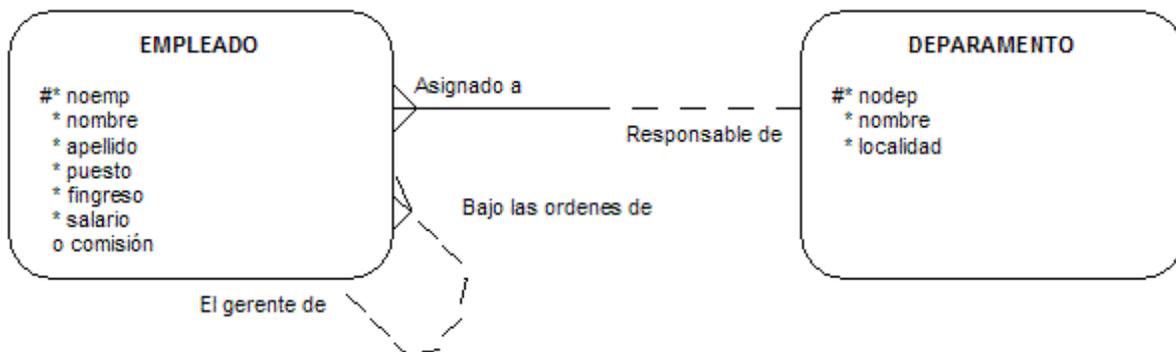
“El departamento de recursos humanos de una compañía necesita tener la información de cada uno de sus empleados, nombres, apellidos, puesto, fecha de contratación y salario de cada uno. Hay algunos empleados que tiene comisión. A cada empleado se le asigna un número único.

La compañía está dividida en departamentos. Cada empleado está asignado a un departamento, por ejemplo, contabilidad, ventas o desarrollo. Se necesita conocer el nombre del departamento responsable de cada empleado y la localización del departamento. Cada departamento tiene un número único, por ejemplo, contabilidad es 10 y ventas tiene el número 30.

Algunos de los empleados son gerentes. También se requiere saber quién es el gerente de cada empleado y que empleados tiene cada gerente.”

Introducción Al Modelo De Datos Conceptual

“Es el resultado del análisis que define y modela los aspectos importantes de la información que el negocio necesita saber o tener y las relaciones entre dicha información.” Como se muestra en la siguiente figura.



Sintaxis de escritura de relaciones

- Cada EMPLEADO debe estar asignado a uno y solo un DEPARTAMENTO*
- Cada DEPARTAMENTO puede ser responsable de uno o más EMPLEADOS*
- Cada EMPLEADO puede ser el gerente de uno o más EMPLEADOS*
- Cada EMPLEADO puede estar bajo las ordenes de uno y solo un EMPLEADO*

En el **DISEÑO** de base de datos se mapean los requerimientos de información reflejados en un modelo E-R. En tablas.

El diseño de la base de datos de recursos humanos se muestra a continuación:

Nombre de la tabla: DEPARTAMENTOS

Columna	NODEP	NOMDEP	LOCALIDAD
Tipo Llave			
Nulos/Únicos			
Ejemplos			

Nombre de la tabla: EMPLEADOS

Columna	NOEMP	NOMBRE	APELLIDO	PUESTO	FINGRESO	SALARIO	COMISION	GERENTE	NODEP
Tipo Llave	PK							FK1	FK2
Nulos/Únicos	NN-U	NN	NN	NN	NN	NN			NN
Ejemplos									

Introducción a la **CONSTRUCCIÓN** de la base de datos

En la construcción de la base de datos se crean físicamente las tablas en la base de datos relacional, implementándolas de acuerdo al diseño de base de datos.

```
SQL> CREATE TABLE DEPARTAMENTOS
2  (NODEP NUMBER(2) NOT NULL,
3  NOMDEP CHAR(20) NOT NULL,
4  LOC CHAR(15) NOT NULL,
5  CONSTRAIN PK_DEP PRIMARY KEY (NODEP)
6  USING INDEX PCTFREE 10
7  STORAGE (INITIAL 64K NEXT 64K PCTINCREASE 0 ))
8  PCTFREE 40 PCTUSED 40 INITRANS 1 MAXTRANS 255
9  STORAGE(
10 INITIAL 10K NEXT 10K PCTINCREASE 50
11 MINEXTENTS 1 MAXEXTENTS 121) NOCACHE.
```

```
SQL> CREATE TABLE EMPLEADOS
2  (NOEMP NUMBER(5) NOT NULL CONSTRAINT PK_emp PRIMARY KEY,
3  NOMBRE CHAR(15) NOT NULL,
4  APELLIDO CHAR(15) NOT NULL,
5  PUESTO CHAR(9) NOT NULL,
6  FINGRESO DATE NOT NULL,
7  SALARIO NUMBER(7.2) NOT NULL,
8  COMISION NUMBER(7.2) NOT NULL,
9  GERENTE NUMBER(5) CONSTRAINT FK_emp FOREIGN KEY(noemp)
10 REFERENCES EMPLEADOS(noemp)
11 NODEP NUMBER(2) NOT NULL CONSTRAINT FK_dep FOREIGN KEY(nodep)
12 REFERENCES DEPARTAMENTOS(nodep)
13 USING INDEX
14 TABLESPACE BRITTEP01 PCTFREE 10
15 STORAGE (INITIAL 100, NEXT 10K PCTINCREASE 0 ))
16 TABLESPACE BRITTEP01
17 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
18 STORAGE(
19 INITIAL 10K PCTINCREASE 50
20 MINEXTENTS 1 MAXEXTENTS 112) NOCACHE;);
```

```
SQL> ALTER TABLE EMPLEADOS
2  ADD CONSTRAINT FK_dep FOREIGN KEY(noemp)
3  REFERENCES DEPARTAMENTOS (nodep)
4  ON DELETE CASCADE
```

```
SQL> ALTER TABLE EMPLEADOS
2  ADD CONSTRAINT FK_dep FOREIGN KEY(noemp)
3  REFERENCES EMPLEADOS (noemp)
4  ON DELETE CASCADE
```

Opciones para los constraints de Delete y Update (Cascade, Restricted, Nullify)

Recomendación:

- Delete con Restricted
- Update con Cascade
- Cuando es Nula la relación se puede utilizar Nullify
- Para borrar un constraint de una tabla es necesario utilizar una sentencia ALTER con una cláusula DROP. Por Ejemplo:
SQL> ALTER TABLE EMPLEADOS
2 DROP CONSTRAINT FK_emp

Para borrar una tabla con toda la integridad referencial

```
SQL> DROP TABLE EMPLEADOS CASCADE CONSTRAINTS
```

Definición De Modelo De Datos

Es el resultado del análisis de la información y consiste en la representación conceptual de esta. Debe ser claro.

Tipos De Modelos De Datos Y Ejemplos Comerciales De DBMS

- Jerárquico. Se puede especificar relaciones de un registro padre y múltiples registros hijos. Ejemplo: IDMS DE CULLNET, IMS DE IBM 1960
- De Red. Se puede especificar relaciones entre múltiples registros padres y múltiples registros hijos. Ejemplo: CODASYL 1970.
- Relacional. Se representa lógicamente la base de datos mediante entidades y sus relaciones entre sí. Ejemplo: INFORMIX, INGRESS, ORACLE (1979), DB2, PARADOS, SQLSERVER, SYBASE, PROGRESS, SQLBAE, ADABAS O NATURAL, FOXPRO, CLIPPER 1980.

Ejemplo Relacional y red: RAIMA

- *Teoría del modelo de datos Entidad-Relación*

Se basa en la percepción del mundo real, que consiste en un conjunto de objetos llamadas entidades y las relaciones entre ellas.

Representa la estructura lógica general de la base de datos gráficamente

Entidad. Es un objeto concreto como una persona o abstracto como un día festivo.

Es una persona, cosa, evento de interés a la comunidad usuaria.

Siempre son sustantivos tangibles o intangibles.

Ejemplo: Juan López

Una entidad está formada por un conjunto de atributos (características propias).

Para cada atributo existe un rango de valores permitidos llamado dominio del atributo.

Ejemplo: El dominio del atributo Nombre podría ser el conjunto de todas las letras del abecedario. El dominio del atributo Número de Cuenta podrían ser el conjunto de todos los enteros positivos

Conjunto de entidades

Es un grupo de entidades del mismo tipo. Por ejemplo: Personas que tiene cuenta en un banco.

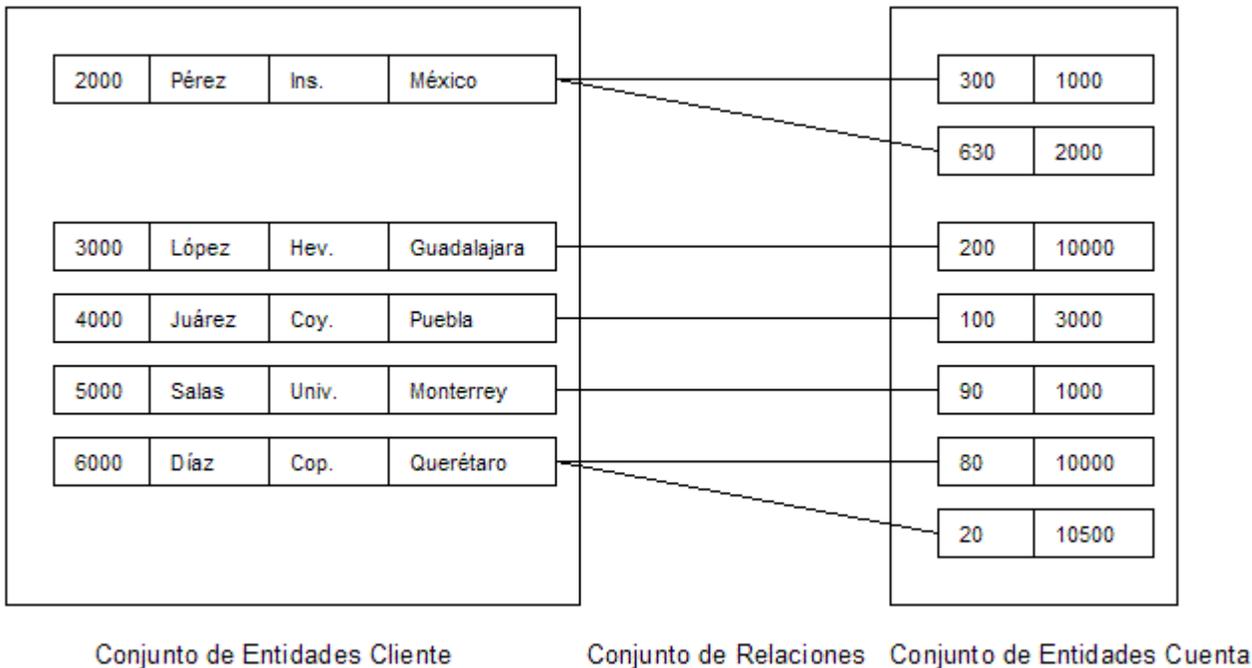
Ejemplos de conjuntos de entidades de aplicación bancaria

- Conjunto de todas las sucursales de un banco
Sucursal(Clave, Nombre, Ciudad_Sucursal, activo)
- Conjunto de todas las personas que tiene una cuenta
Cliente (Seguro_Social, Nombre, Calle, Ciudad)
- Conjunto de todas las cuentas que mantienen en el banco
Cuenta(Número, Saldo)
- Conjunto de todos los empleados que laboran en un banco
Empleado(Id, Nombre, Teléfono)
- Conjunto de transacciones para una cuenta determinada
Transacción(Número, Fecha, Importe)

Relación. Asociación entre entidades. Ejemplo: Asociación del Cliente López con Número de Cuenta 200.

Conjunto de relaciones

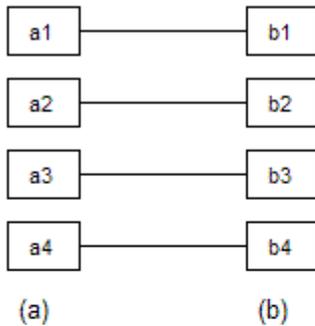
Es un grupo de relaciones del mismo tipo. Ejemplo: Las relaciones Cliente-Cuenta para denotar la relación entre Cliente y Cuenta



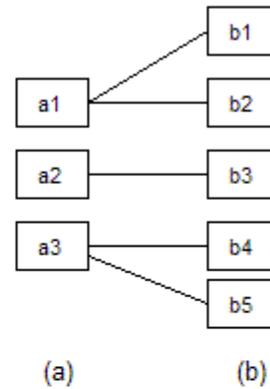
Relaciones Binarias. Solo implica dos conjuntos de entidades

Tipos De Relaciones

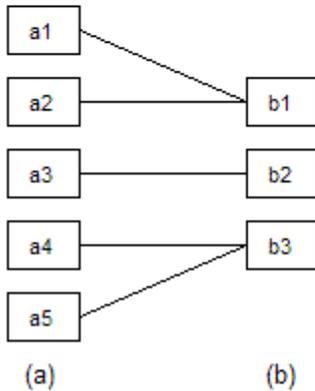
Una a Una



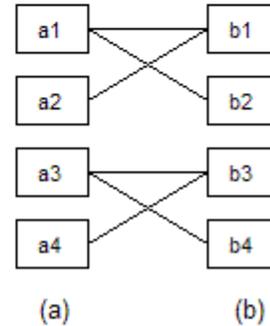
Una a Muchas



Muchas a Una



Muchas a Muchas



Llave Candidato

Conjunto de uno o más atributos que juntos identifican a una entidad como única

Ejemplo: El atributo Seguro_Social es una llave candidato porque es suficiente para distinguir una entidad Cliente de otra.

El atributo Seguro_Social + Nombre también es una llave candidato.

El apellido no es una llave candidato porque varios clientes pueden tener el mismo apellido.

Llave Primaria

Es la llave candidato elegida para identificar una entidad como única. La llave candidato más pequeña posible.

Dependencia Por Existencia

Cuando la existencia de la entidad X depende de la existencia de la entidad Y. X es una subordinada y Y es dominante.

Ejemplo: Cuenta (dominante) y Transacción (subordinada)
Futbolista (dominante) y Gol (subordinada)

Entidad Fuerte O Dominante

Entidad que tiene suficientes atributos para formar una llave primaria

Entidad Débil O Subordinada

Entidad que no tiene suficientes atributos para formar una llave primaria.

Ejemplo: Transacción (2 transacciones pueden ser iguales con diferentes cuentas).

Nota: Para tener un identificador único necesita ayuda de la entidad fuerte o dominante, está formada por la llave primaria de la entidad dominante y el discriminador de la débil (Conjunto de atributos que permiten hacer esta distinción)

Ejemplo: la llave primaria (PK) de Transacción: Número_Cuenta, Número_Transacción.

Llave Foránea

Es la llave primaria en una entidad y es la que nos sirve para relacionar a otra entidad.

Ejemplo: Numero_Cuenta en Transacción (FK).

Nota: A veces también forma parte de la llave primaria en la entidad Destino (llamada llave primaria compuesta)

d) ANÁLISIS

- *Identificar entidades*

Pasos Para Análisis Relacional

1. Identificar entidades

Mediante los siguientes pasos se puede llegar a identificar y modelar las entidades de las notas de entrevistas a los usuarios, de formatos, etc.

Examinar los sustantivos (son aspectos importantes)

Poner un nombre a cada entidad

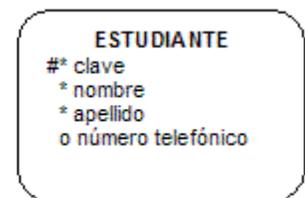
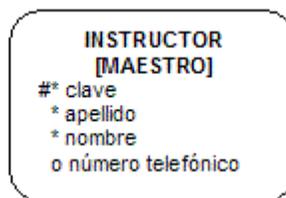
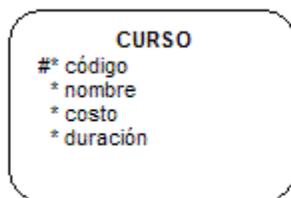
¿Existe información de interés para la compañía acerca de la entidad?

¿Cada entidad es identificable de manera única? ¿Cuál o cuáles atributos sirven como UID?

Ejemplo: Identificar las entidades de los siguientes requerimientos.

"Soy el administrador de una compañía de capacitación que provee de cursos técnicos guiados por nuestros instructores. Enseñamos muchos cursos, cada uno tiene un código, nombre y un costo.

Introducción a UNIX y Programación en C son 2 de nuestros cursos más populares. Los cursos varían en duración desde 1 día hasta 4 días. Un instructor puede enseñar varios cursos. Alberto Suárez y María González son 2 de nuestros mejores instructores. Registramos el nombre de cada instructor y su número telefónico. Cada curso es enseñado por sólo un instructor. Se designa el curso y en base a ello asignamos al instructor. Los estudiantes pueden tomar varios cursos a través del tiempo. Jaime Martínez de AT&T toma cada curso de los que ofrecemos. También llevamos registro de los nombres de los estudiantes y su número telefónico. Varios de nuestros estudiantes e instructores no nos dan su número telefónico."



- *Descripción de entidades*

2. *Descripción de entidades*

Escribir la descripción de la entidad.

Ejemplo:

1. Un CURSO es un evento de entrenamiento ofrecido por la compañía. Ejemplo: Introducción a Unix
2. Un ESTUDIANTE es un participante en uno o varios cursos. Ejemplo: Jaime Martínez
3. Un INSTRUCTOR es un maestro de uno o varios cursos. Ejemplo: Alberto Suárez

- *Establecer las relaciones entre entidades*

3. Establecer las relaciones entre unidades

Ejemplo: La relación entre un INSTRUCTOR y un CURSO es:

Cada CURSO debe ser enseñado por uno y solamente un INSTRUCTOR

Cada INSTRUCTOR puede ser asignado a uno o mas CURSOS

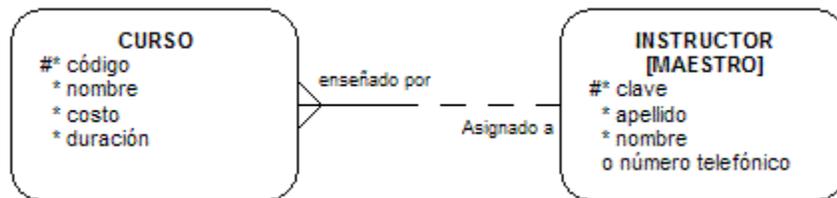
Cada dirección de una relación tiene:

Un nombre. Ejemplo: Enseñado por o asignado a

Una opción. Ejemplo: Debe ser/estar o puede ser/estar

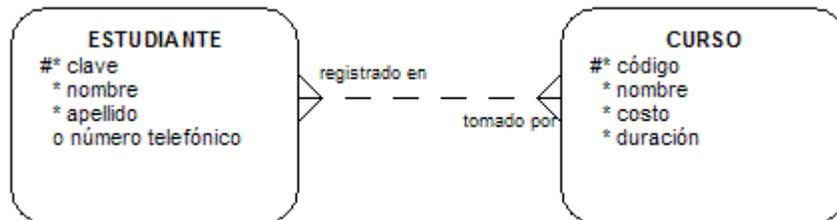
Un grado (cardinalidad). Uno y solamente uno ó uno o mas

Primero se debe leer la relación en una dirección y después leer la relación en la otra dirección.



Cada CURSO debe ser enseñado por uno y solamente un INSTRUCTOR

Cada INSTRUCTOR puede ser asignado a uno o mas CURSOS



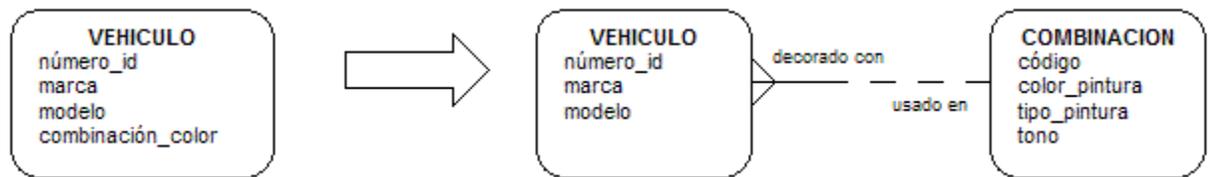
Cada ESTUDIANTE puede estar registrado en uno o mas CURSOS

Cada CURSO puede ser tomado por uno o mas ESTUDIANTES

- *Diferencias entre atributos y entidades*

Si un atributo tiene atributos propios, entonces es una entidad.

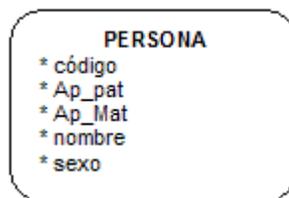
Ejemplo: Determinar si todos los atributos de la entidad VEHÍCULO son atributos



Inicialmente el usuario identifica la combinación de color como un atributo del VEHÍCULO. Después el usuario define los requerimientos para color, tipo y tono para cada color. Opcionalidad De Atributos

Para analizar y modelar relaciones se deben: Los atributos de una entidad pueden ser opcionales (o) u obligatorios (*)

Ejemplo: Opcionalidad de los atributos de la entidad PERSONA



El mapa de instancias sirve para validar que sean correctas las marcas para atributos (opcionales y obligatorios) se utiliza para almacenar ejemplos de datos de atributos y cada mapa corresponde a una entidad y debemos relacionarlos para la integridad referencial.

Ejemplo:

Nombre de la tabla: PERSONAS

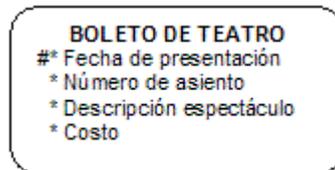
Nombre del atributo	código	Ap_Pat	Ap_Mat	nombre	sexo
Etiqueta	*	*	*	*	*
Ejemplo de datos	1	Juárez	García	Jorge	M
	301	lópez	Márquez	Octavio	M
	134	García	Lara	Celia	F

- *Identificadores únicos*

En un negocio cada ocurrencia de DEPARTAMENTO se identifica de manera única por el número de departamento



Para un teatro, cada boleto es identificado como único con la fecha de presentación y el número de asiento

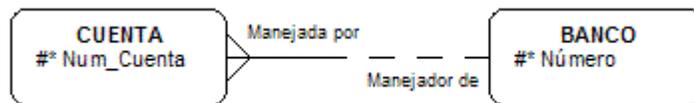


Nota: Si tuviera un número de folio único ese sería el UID. En este caso si se presentan varias funciones el mismo día entonces para el UID es importante que la fecha incluya la hora.

Una entidad debe tener un UID, o ésta no es una entidad.

Todos los componentes de un UID deben ser Marcados con #* ó con # únicamente Una entidad puede ser identificada de manera única a través de una relación.

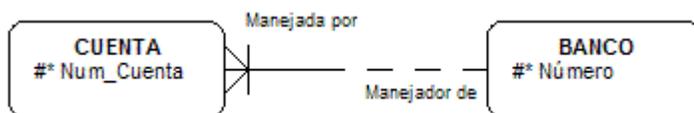
Ejemplo: En la industria bancaria, a cada banco se le asigna número único de banco. Dentro de cada banco cada cuenta tiene un número secuencial de cuenta. ¿Cuál es el UID de la entidad CUENTA?



La CUENTA es identificada como única por su atributo número y el BANCO específico a la que está relacionada.

Usar una barra UID para indicar que la relación es parte del identificador único de la entidad.

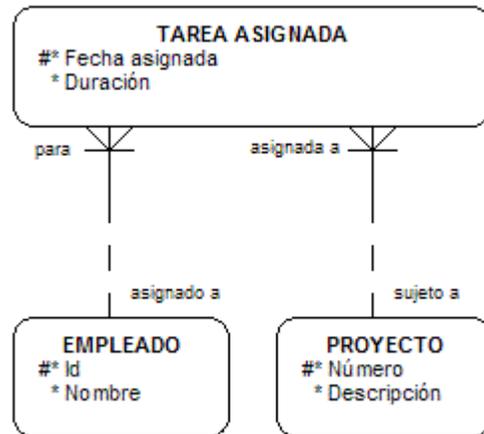
Ejemplo: La barra UID indica que la relación con el BANCO es parte del UID de CUENTA



Nota: Una relación incluida dentro de un UID debe de ser obligatoria del lado de la entidad débil y una y solamente una del lado de la entidad dominante.

Una entidad puede ser identificada de manera única a través de múltiples relaciones.

Ejemplo: En un negocio se necesitan conocer las tareas asignadas a sus empleados. A los empleados se les asignan tareas de los proyectos. Un empleado puede tener múltiples tareas asignadas en uno o varios proyectos, cada uno con diferente fecha de asignación.



¿Cual es el UID de la entidad TAREA ASIGNADA?

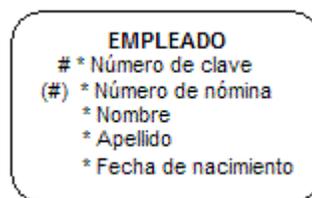
Una TAREA ASIGNADA es identificada de manera única para el EMPLEADO a quien se le dio la TAREA ASIGNADA, el PROYECTO que contiene a la TAREA ASIGNADA y por la Fecha asignada.

* Ambas relaciones son obligatorias y una y sólo una en la dirección incluida en el UID.

- *Identificadores únicos secundarios*

Cada UID secundario se marca con (#)

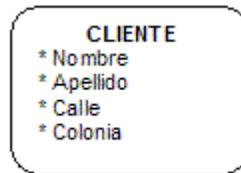
Ejemplo:



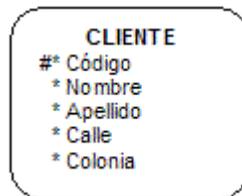
Atributos adicionales para identificar una entidad

Hay casos en los que es necesario crear un atributo adicional para ayudar a identificar a una entidad en caso de no existir.

¿Cuál es el identificador único de la entidad CLIENTE?



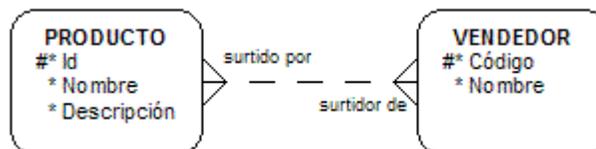
Crear un atributo adicional llamado código, el cual sea único para cada instancia del CLIENTE



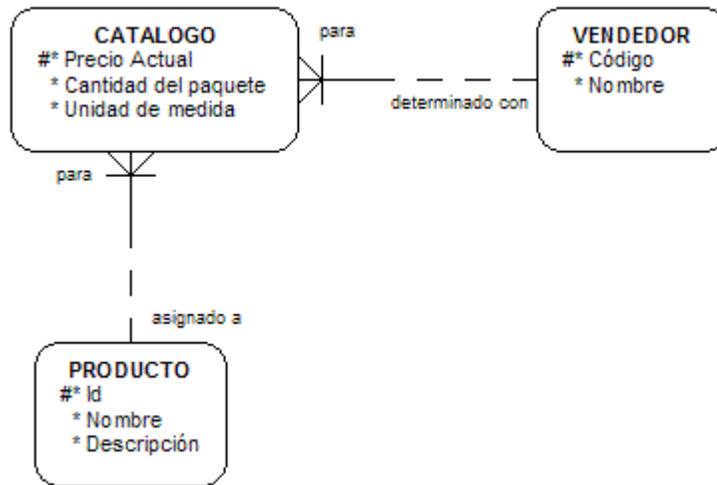
- *Resolución de relaciones M:M*

Se debe agregar una entidad intersección con los atributos que identifican a cada una de las entidades.

¿Cuál es el precio actual de un PRODUCTO específico para un VENDEDOR específico?



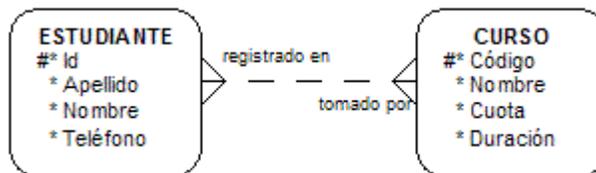
El precio actual, cantidad del paquete y unidad de medida son atributos de la relación entre PRODUCTO y VENDEDOR. Cuando se crea una Entidad Intersección nueva se crean 2 relaciones M:1



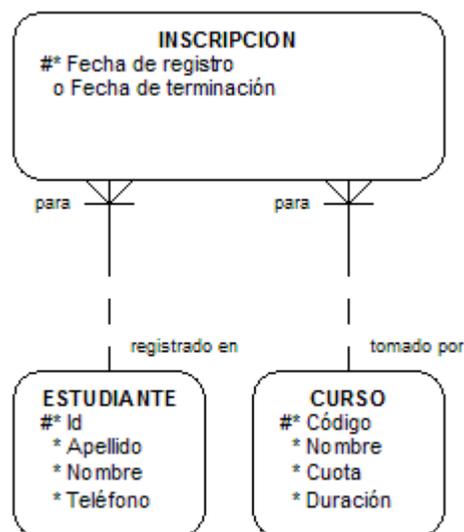
El UID del CATALOGO esta compuesto de estas 2 relaciones, observe las 2 barras, las relaciones desde una entidad intersección son siempre obligatorias; éstas entidades de INTERSECCIÓN generalmente tienen atributos adicionales como cantidades y fechas. Estas tienden a ser entidades volátiles y de volumen alto.

Nota: Una entidad intersección sin atributos es una lista de referencia cruzada.

Ejemplo: "Cada ESTUDIANTE tiene una fecha de registro en el CURSO, una fecha de terminación del curso"



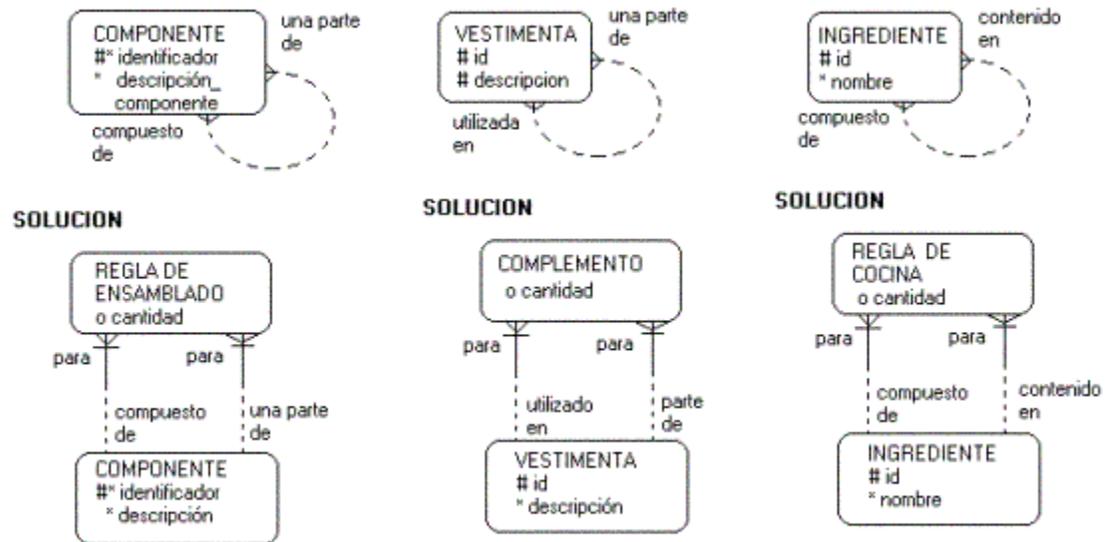
Solución



Relaciones Recursivas M:M

Resolución de la relación recursiva M:M con una entidad intersección y 2 relaciones M:1 hacia diferentes instancias de la entidad original.

Ejemplo: Una organización de manufactura de automóviles necesita guardar información de partes, subensambles, ensambles y productos, por ejemplo partes que componen un ventilador.



- *Normalización en Análisis*

Se debe evaluar y validar los atributos y las relaciones para identificar cada entidad

Normalizar el MODELO DE DATOS nos sirve para validar cada atributo usando las siguientes reglas de normalización.

REGLA DE FORMA NORMAL DESCRIPCION

1FN Todos los atributos deben tener un sólo valor

2FN Un atributo debe ser dependiente del identificador único completo

3FN Ningún atributo no-UID puede ser dependiente de otro atributo no-UID (Su objetivo es eliminar redundancia)

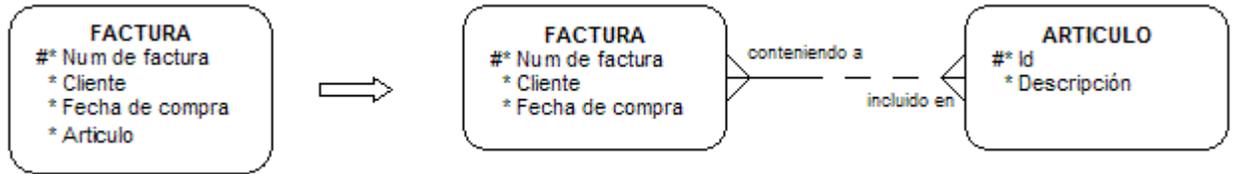
Un modelo de datos entidad-relación normalizado se traslada automáticamente en un diseño de B.D. Después de la 3FN no es comúnmente utilizado.

1FN. Valor único para cada ocurrencia

Ejemplo: ¿La entidad FACTURA cumple con 1FN? No

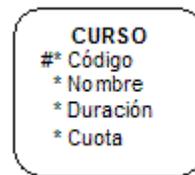
SI UN ATRIBUTO TIENE MÚLTIPLES VALORES, SE CREA UNA ENTIDAD ADICIONAL Y SE RELACIONA CON LA ENTIDAD ORIGINAL MEDIANTE UNA RELACION M:1

Ejemplo:

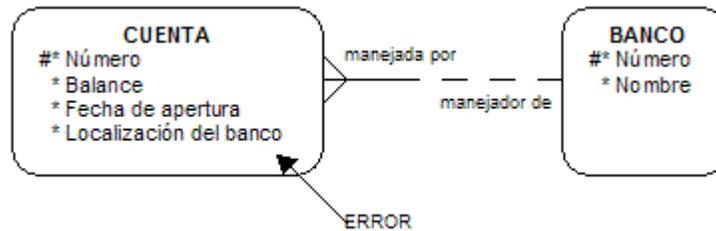


2FN. Un atributo debe ser dependiente del identificador único completo

Ejemplo: ¿La entidad CURSO cumple con 2FN? Si, cada instancia del código de curso determina un valor específico para nombre, duración y cuota.



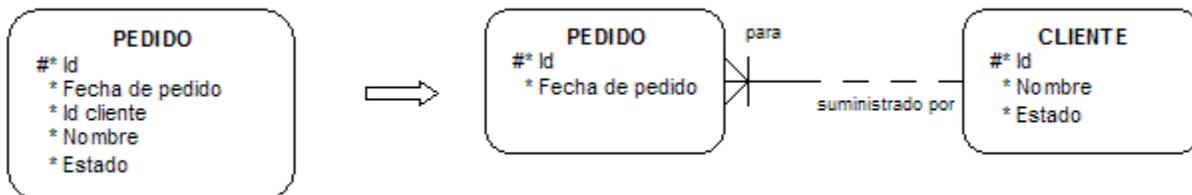
Validar la ubicación de los siguientes atributos. Si un atributo no es dependiente del UID completo, deberá ser reubicado.



3FN. Ningún atributo no-UID puede ser dependiente de otro atributo no-UID

- Validar que ningún atributo no-UID no dependa de otro no-UID
- Reubicar cualquier atributo no-UID que dependa de otro no-UID

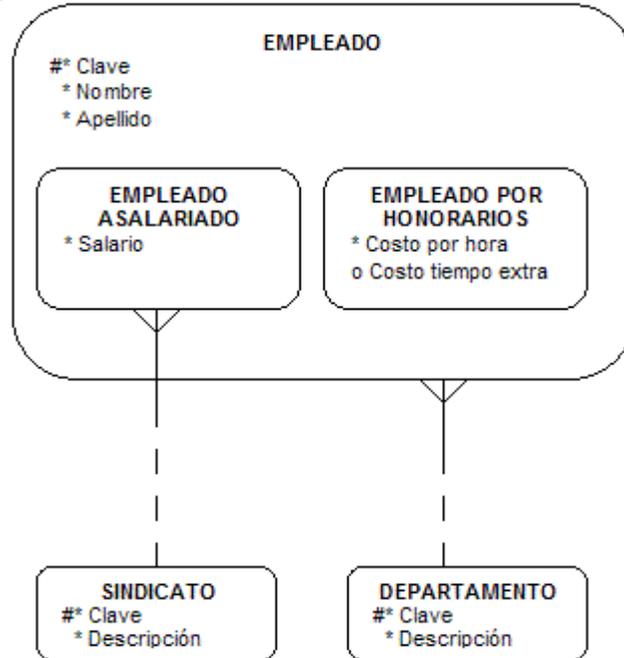
Ejemplo: ¿Algún atributo no-UID de la siguiente entidad depende de otro atributo no-UID? Si, los últimos dos dependen de Id cliente. Se deben trasladar ambos atributos dependientes del UID



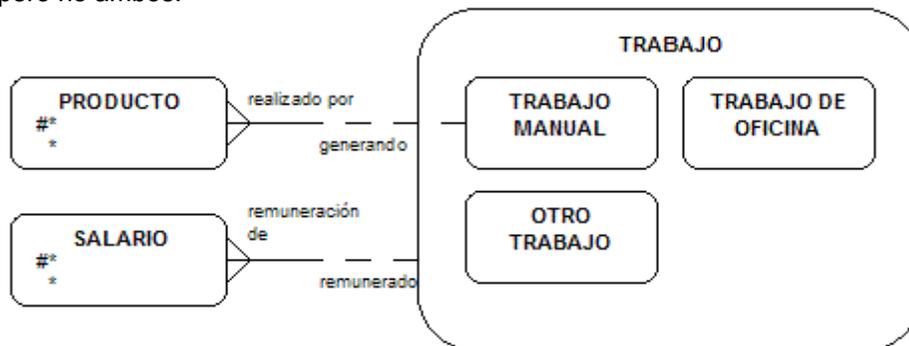
- *Subtipos o subentidades y supertipos o superentidades*

Los subtipos sirven para modelar exclusivamente tipos de entidad que tienen atributos o relaciones comunes. Un supertipo es una entidad que tiene subtipos. Un supertipo puede ser separado en dos o más subtipos mutuamente excluyentes.

Ejemplo: Un negocio tiene dos tipos de empleados: asalariados y por honorarios. Para todos los empleados se guarda su número, nombre, apellido y departamento asignado. Para los empleados asalariados, se guarda el salario del empleado y sindicato. Para los empleados por honorarios, se guarda el costo por hora y costo por tiempo extra.



Nota: Un empleado debe ser EMPLEADO ASALARIADO ó EMPLEADO POR HONORARIOS, pero no ambos.



Estándares De Lectura De Supertipos

Cada TRABAJO MANUAL el cual es un TRABAJO puede estar generando uno o más PRODUCTOS.

Cada PRODUCTO debe ser realizado por uno y solo un TRABAJO MANUAL el cual es un TRABAJO

Cada SALARIO debe ser remuneración de un y solo un TRABAJO que puede ser MANUAL, OFICINA u OTRO TRABAJO

Cada TRABAJO que puede ser MANUAL, OFICINA u OTRO TRABAJO puede ser remunerado por uno o más SALARIOS

Nota: Siempre se utilizan subtipos OTROS cuando no se está seguro.

- *Relaciones excluyentes (arco)*

Modelar dos o más relaciones mutuamente excluyentes de la misma entidad usando un arco.

Ejemplo: Se usa un arco para modelar la relación de una CUENTA BANCARIA que debe pertenecer a una COMPAÑÍA o ser PERSONAL.

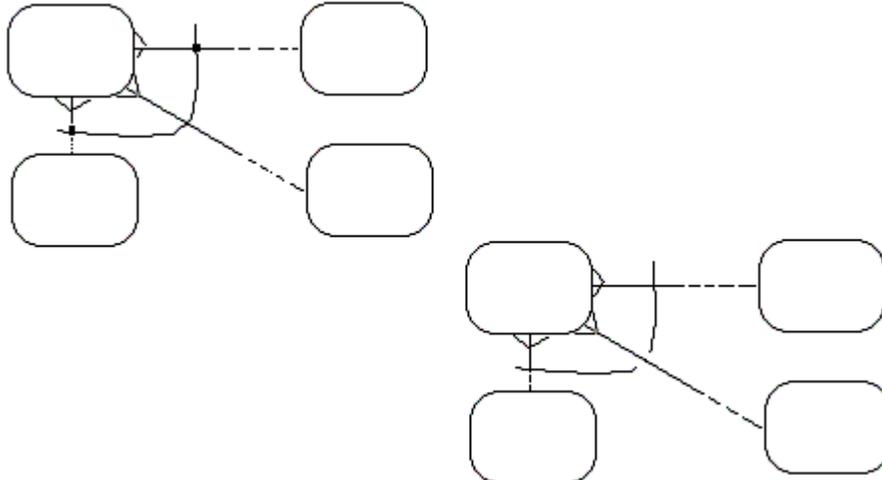


Reglas De Lectura Para La Relación Exclusiva

Cada CUENTA BANCARIA debe estar perteneciendo a una y sólo una PERSONA o estar perteneciendo a una y sólo una COMPAÑIA

Tipos Para Dibujo De Arcos

Un arco con puntos opcionales Rompimiento de arco.



e) **ÁLGEBRA RELACIONAL**

- *Teoría*

El Diseño de Base de Datos es el mapeo a tablas del DER

TABLA SUCURSALES

CLAVE	NOMBRE	ACTIVO	CIUDAD
Primary Key			
1	Norte	1000	México
2	Sur	2000	Puebla
3	Este	3000	México
4	Oeste	500	Querétaro
5	Centro	1000	México
6	Poniente	1000	México

TABLA CUENTAS

CLAVE_SUC	NUM_CLIENTE	NUM_CUENTA	SALDO
Primary Key Y FK1	Primary Key Y FK2	Primary Key	
1	500	500	1000
1	2000	401	10500
1	3200	210	10000
3	0003	220	3000
4	400	400	3000
4	850	300	1000
5	100	630	2000
6	1000	1000	10000

TABLA CLIENTES

NUM_CLIENTE	NOMBRE	CALLE_NUM	CIUDAD
Primary Key			
0003	Acosta Adriana	Madero 23	México
0100	Pérez Rodolfo	Rev 1020	Puebla
0400	Valencia Sergio	Libertad 398	Querétaro
0500	Juárez José Alfonso	Copilco 15	Chiapas
4	400	400	3000
0850	Lara Susana	Xola 125	México
1000	Ruiz Gonzalo	Universidad 80	México
1008	Fernando Marco	Ins 122	México
2000	López Alejandra	Allende 11	Guadalajara
3200	Gómez Verónica	Progreso 37	Monterrey
7000	Hernández Luis	Xola 94	México

TABLA DEPOSITOS

CLAVE_SUC	NUM_CLIENTE	NUM_CUENTA	FECHA	SALDO
Primary Key Y FK1	Primary Key Y FK2	Primary Key Y FK3	Primary Key	
1	2000	401	2-Ene-98	1000
1	3200	210	5-Mar-98	450
2	1008	560	7-Jun-98	100
3	0003	220	1-Jul-97	180
4	0850	300	3-Nov-97	290

TABLA PRESTAMOS

CLAVE_SUC	NUM_CLIENTE	NUM_CUENTA	NUM_PRESTAMO	IMPORTE
Primary Key Y FK1	Primary Key Y FK2	Primary Key Y FK3	Primary Key	
1	0500	500	23	3000
1	2000	401	012	500
2	1008	560	50	300
4	0400	400	17	100
5	0100	630	10	1000
6	1000	1000	18	850

EXISTE UNA CORRESPONDENCIA DIRECTA ENTRE CONCEPTO DE UNA **TABLA** Y EL CON-CEPTO MATEMATICO DE UNA **RELACION**, DEL CUAL RECIBE EL NOMBRE DE MODELO RELACIONAL.

TABLA DEPOSITOS = RELACION DEPOSITOS

CLAVE_SUC	NUM_CLIENTE	NUM_CUENTA	FECHA	SALDO
Primary Key Y FK1	Primary Key Y FK2	Primary Key Y FK3	Primary Key	
1	2000	401	2-Ene-98	1000
1	3200	210	5-Mar-98	450
2	1008	560	7-Jun-98	100
3	0003	220	1-Jul-97	180
4	0850	300	3-Nov-97	290

D1 (Dominio1) Es el conjunto de todas las claves de sucursal

D2 (Dominio2) Es el conjunto de todos los números de cliente

D3 (Dominio3) Es el conjunto de todos los números de Cuenta

D4 (Dominio4) Es el conjunto de todas las fechas de Depósito de los Clientes

D5 (Dominio5) Es el conjunto de todos los Saldos depositados

Cada Tupla (renglón) se compone de (V1,V2,V3,V4,V5) donde V1 pertenece a D1 y así sucesivamente.

Por tanto DEPOSITOS es un subconjunto de

$$\times_{i=1}^5 D_i$$

En general una tabla de n columnas es un subconjunto de

$$\times_{i=1}^n D_i$$

Los matemáticos llaman RELACION al subconjunto del producto cartesiano de una lista de dominios, esto es similar a una tabla.

- Operadores fundamentales y adicionales

Álgebra Relacional. Es un lenguaje de consulta (sirve para obtener la información de la Base de Datos). Es menos poderoso que sql (Inserción, Actualización y Eliminación). Consta de 5 operaciones fundamentales:

- | | |
|----------------------------|----------|
| 1. Elegir | σ |
| 2. Proyectar | π |
| 3. Producto Cartesiano | \times |
| 4. Unión | \cup |
| 5. Diferencia de conjuntos | - |

Y 2 operaciones adicionales:

- | | |
|---------------------|-----------|
| 1. Intersección | \cap |
| 2. Producto Natural | \bowtie |

Se permiten operadores

<> (diferente) = \geq \leq < > \wedge (AND) \vee (OR)

Producto Cartesiano X (SUCURSALES X CUENTAS)

CLAVE	NOMBRE	ACTIVO	CIUDAD	CLAVE_SUC	NUM_CLIENTE	NUM_CUENTA	SALDO
1	Norte	1000	México	1	500	500	1000
1	Norte	1000	México	1	2000	401	10500
1	Norte	1000	México	1	3200	210	10000
1	Norte	1000	México	3	0003	220	3000
1	Norte	1000	México	4	400	400	3000
1	Norte	1000	México	4	850	300	1000
1	Norte	1000	México	5	100	630	2000
1	Norte	1000	México	6	1000	1000	10000
2	Sur	2000	Puebla	1	500	500	1000
...							
3	Este	3000	México	1	500	500	1000
...							
4	Oeste	500	Querétaro	1	500	500	1000
...							
5	Centro	1000	México	1	500	500	1000
...							
6	Poniente	1000	México	1			
...							
6	Poniente	1000	México	6	1000	1000	10000

- *Ejemplos*

- i. Proyectar el número de cliente, número de cuenta y número de préstamo PERO las tuplas de la relación PRESTAMOS de la sucursal 1

Π NUM_CLIENTE, NUM_CUENTA, NUM_PRESTAMO
 σ CLAVE_SUC = 1 (PRESTAMOS) = 'Norte'

NOTA: Cuando se utiliza más de una tabla, se realiza un JOIN entre las columnas en común.

NUM_CLIENTE	NUM_CUENTA	NUM_PRESTAMO
0500	500	23
2000	401	012

- ii. Proyectar el número de cliente, número de cuenta y número de préstamo PERO las tuplas de la relación PRESTAMOS de la sucursal Norte

Π NUM_CLIENTE, NUM_CUENTA, NUM_PRESTAMO
 σ CLAVE_SUC = CLAVE
 NOMBRE = 'Norte' (PRESTAMOS SUCURSALES)

- iii. Proyectar los nombres de clientes y su nombre de sucursales en las que han hecho préstamos.

Π CLIENTES.nombre, SUCURSALES.nombre
 σ CLIENTES.NUM_CLIENTE = PRESTAMOS.NUM_CLIENTE
 CLAVE_SUC = CLAVE (PRESTAMOS X CLIENTES X SUCURSALES)

CLIENTES.nombre	SUCURSALES.nombre
Juárez José Alfonso	Norte
López Alejandra	Norte
Fernández Marco	Sur
Valencia Sergio	Oeste
Pérez Rodolfo	Centro
Ruiz Gonzalo	Poniente

CONDICIÓN DE UNIÓN: Unión Compatible

El orden (Número de Atributos) y los dominios de cada atributo deben ser iguales. Las dos relaciones deben tener el mismo número de argumentos.

Ejemplo:

R10 = { (1,a), (1,b), (1,c) }
 R20 = { (1,a), (2,b), (3,b) }
 R30 = { (a), (b), (1) }

R10 \cup R30 = No se puede realizar porque no son compatibles

- iv. Proyectar los clientes que han hecho operaciones en la sucursal Norte.
 Π CLIENTES.nombre,
 σ SUCURSALES.nombre = 'Norte'
 \wedge clave = clave_suc
 \wedge CLIENTES.num_cliente = DEPOSITOS.num_cliente
 (DEPOSITOS X CLIENTES X SUCURSALES)

∪

- Π CLIENTES.nombre
 σ SUCURSALES.nombre = 'Norte'
 \wedge clave = clave_suc \wedge CLIENTES.num_cliente = PRESTAMOS.num_cliente
 (PRESTAMOS X CLIENTES X SUCURSALES)

CLIENTES.nombre
López Alejandra
Gómez Verónica
Juárez José Alfonso

- v. Obtener los clientes que no han hecho ningún préstamo en la sucursal Norte
 Π CLIENTES.nombre
 σ SUCURSALES.nombre \neq 'Norte'
 \wedge PRESTAMOS.num_cliente = CLIENTES.num_cliente
 \wedge clave = clave_suc (PRESTAMOS CLIENTES SUCURSALES)

CLIENTES.nombre
Pérez Rodolfo
Fernández Marco
Ruiz Gonzalo

Una expresión en Álgebra Relacional se puede construir con subexpresiones.

Sean E_1 y E_2 expresiones de Álgebra Relacional:

$E_1 \cup E_2$
 $E_1 - E_2$
 $E_1 \times E_2$

$\sigma_p(E_1)$ donde p = Predicado con atributos de E_1
 $\Pi_s(E_1)$ donde s = Lista de atributos que aparecen de E_1

OPERADORES ADICIONALES. \cap y (X)

vi. Obtener a todos los clientes que tienen tanto un préstamo, como un depósito en la QZ

Π CLIENTES.nombre
 σ SUCURSALES.nombre = 'Norte'
 \wedge clave_suc = clave
 \wedge PRESTAMOS.NUM_CLIENTE = CLIENTES.NUM_CLIENTE
(PRESTAMOS X SUCURSALES X CLIENTES)
 \cap

Π CLIENTES.nombre
 σ SUCURSALES.nombre = 'Norte'
 \wedge clave_suc = clave
 \wedge DEPOSITOS.NUM_CLIENTE = CLIENTES.NUM_CLIENTE
(DEPOSITOS X SUCURSALES X CLIENTES)

CLIENTES.nombre
López Alejandra

vii. Obtener todos los nombres y las ciudades de residencia de los clientes de la sucursal Norte que han hecho préstamos

Π CLIENTES.nombre, ciudad
 σ_p (PRESTAMOS X CLIENTES X SUCURSALES)

donde

$p =$ CLIENTES.NUM_CLIENTE = PRESTAMOS.NUM_CLIENTE
CLAVE_SUC = CLAVE
SUCURSALES.NOMBRE = 'Norte'

CLIENTES.nombre	ciudad
Juárez José Alfonso	Chiapas
López Alejandra	Guadalajara

Utilizando Producto Natural

Π CLIENTES.nombre, ciudad (PRESTAMOS (X)* CLIENTES (X) SUCURSALES)

donde

θ es el predicado
 $\theta =$ SUCURSALES.nombre = 'Norte'

- viii. Obtener los nombres de los clientes y las ciudades de residencia de todos los clientes que tienen un préstamo. Con producto natural:

Π nombre, ciudad (PRESTAMOS (X) CLIENTES)

Aquí se toman únicamente las parejas que tienen el atributo en común número_cliente.

nombre	ciudad
Juárez José Alfonso	Chiapas
López Alejandra	Guadalajara
Fernández Marco	México
Valencia Sergio	Querétaro
Pérez Rodolfo	Puebla
Ruiz Gonzalo	México

- ix. Obtener el nombre de todas las sucursales y el activo que tienen depositantes que vivan en México.

Π SUCURSALES.nombre, activo

σ CLIENTES.ciudad = 'México'

(CLIENTES (X) DEPOSITOS (X) SUCURSALES)

SUCURSALES.nombre	activo
SUR	2000
ESTE	3000
OESTE	500

- x. Obtener los clientes que tienen tanto un depósito, como un préstamo en la sucursal "Norte". (Similar al ejercicio 6)

Π CLIENTES.nombre

σ SUCURSALES.nombre = 'Norte'

(PRESTAMOS (X) SUCURSALES (X) CLIENTES)

\cap

Π CLIENTES.nombre σ SUCURSALES.nombre = 'Norte'

(DEPOSITOS (X) SUCURSALES (X) CLIENTES)

CLIENTES.nombre
López Alejandra

- *Estándares*

Los estándares son especificaciones establecidas con el fin de regular la realización correcta de diagramas, relaciones, normalización, diseño, etc.

Estándares Para Entidades

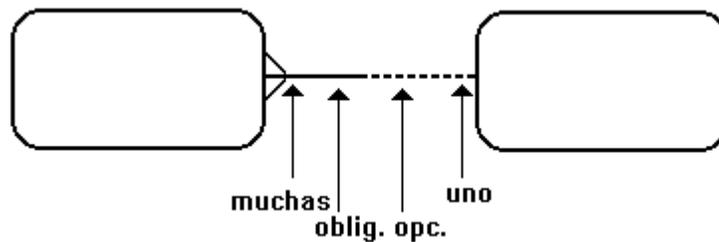
1. Cajas de cualquier dimensión con las esquinas redondeadas
2. Nombre único para cada entidad
3. Nombre de la entidad en mayúsculas
4. Nombre de sinónimo (nombre alternativo para una entidad) opcional (entre paréntesis)
5. Si una entidad no puede tener un identificador único (UID) ésta no puede ser una entidad, # o barra(s) UID
6. Nombre de entidad en singular

Estándares Para Relaciones

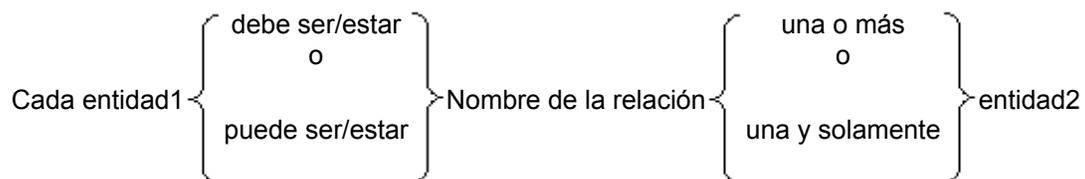
Definir una línea entre dos entidades.

Una relación es bidireccional y representa la asociación entre dos entidades, o entre una entidad consigo misma.

1. Nombres de relaciones en minúsculas
2. Opcionalidad
 - Opcional (Puede ser/estar)
 - Obligatorio (Debe ser/estar)
3. Grado ó Cardinalidad
 - Una ó más
 - Una y solamente una
4. Nombres de Relación alternado y cercano a la entidad

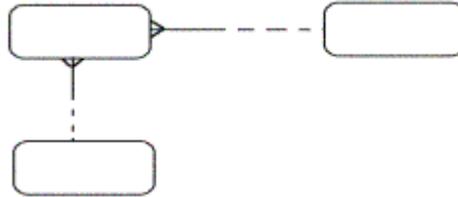


Sintaxis de una relación



Estándares Del Diagrama E-R

1. Alinear las cajas de las entidades
2. Dibujar las líneas de relación como rectas horizontales o verticales
3. Hacer todo el texto claro (entidades, relaciones, atributos, sinónimos)
4. Evitar abreviaciones y modismos (modo de hablar propio de una lengua), en entidades, relaciones, atributos
5. Poner los nombres de la relación próximos a su grado o cardinalidad y en lados opuestos de la línea
6. Hacer el Diagrama E-R fácil de recordar
7. Compactar en la medida de lo posible las cajas de las entidades para ayudar a la visualización del diagrama
8. Dibujar patas de gallo apuntando hacia arriba o a la izquierda Tratar de posicionar la pata de gallo en la línea en la parte izquierda para líneas horizontales y en la parte superior para las líneas verticales



NOTA: Hasta que una relación M:M sea resuelta, al menos un fin de la relación debe apuntar abajo y a la derecha

9. Posicionar las entidades más volátiles arriba y a la izquierda del diagrama y posicionar las entidades menos volátiles abajo y a la derecha del diagrama (catálogos abajo)
10. Nombres de entidades y atributos máximo 29 caracteres
11. Omitir ciclos, excepto roles

Estándares De Atributos

Definición de atributos : describen una entidad para calificar, identificar, clasificar, cuantificar o expresar el estado de una entidad.

¿Cuáles son algunos de los atributos de la entidad EMPLEADO?

El número de nómina que identifica a un EMPLEADO

El nombre y apellidos califican a un EMPLEADO

El tipo de nómina (ejemplo semanal) clasifica a un EMPLEADO

La edad cuantifica a un EMPLEADO

El estatus de empleo (Activo, jubilado) expresa el estatus de un EMPLEADO

Notas:

1. Los nombres de los atributos deben ser claros para un usuario, más no codificado para el desarrollador.
El nombre de la entidad es siempre un calificador de atributos del nombre del atributo. Ejemplo curso. Por lo tanto, los nombres de los atributos no deberían incluir el nombre de la entidad.
2. Los nombres de los atributos deben ser específicos. Ejemplo en el caso de una cantidad, es cantidad vendida o cantidad comprada
3. Clarificar siempre la fecha de un atributo con una descripción o una frase, ejemplo fecha de contrato, fecha de orden
4. Los atributos deben estar en singular (excepto atributos que sí lo necesitan, por ejemplo num_puertas en ENTIDAD auto) y minúsculas

5. Verificar que cada atributo tenga un sólo valor para cada entidad. Un atributo multivalor no es un atributo válido (Normalización). Un atributo repetido indica que falta una entidad (multivalor)
6. Verificar que un atributo no sea derivado o calculado de los valores existentes de otros atributos. No incluir atributos derivados en un modelo E-R
 - Los atributos derivados son redundantes.
 - La redundancia de datos puede ocasionar una inconsistencia de valores de datos. El dato derivado debe ser actualizado cada vez que se modifican los datos en los que se basa.

Ejemplos de CONSTRAINTS de usuario:

- * Contadores (Ejemplo: el número de vendedores en una región)
- * Totales (Ejemplo: el total de ventas mensuales de cada vendedor)
- * Max/Min/Promedio (Ejemplo: Estadísticas de ventas de un grupo de vendedores)
- Otros cálculos (Ejemplo: La comisión de un vendedor calculada al 10% de las ventas)

7. Los atributos que identifican de manera única una entidad y pertenecen a los UID de las entidades son precedidos por #* ó #, los atributos obligatorios por un * y los atributos opcionales son precedidos por una o.

Estándares De Columnas (Diseño)

1. Todas las columnas se deben descomponer hasta su mínimo componente con significado. Ejemplo: El nombre de una PERSONA puede ser descompuesto en nombre y apellidos. Las columnas como fechas, horas, números del seguro social, códigos postales generalmente no se descomponen. Ejemplo ¿Cómo se dividiría la columna dirección? Puede ser descompuesto en múltiples columnas: número, departamento, calle, ciudad, estado y código postal. Generalmente se descompone durante el diseño. El nivel de la descomposición de columnas depende de los requerimientos del negocio.

Estándares Para Modelos De Arco

1. Las relaciones en arco frecuentemente tienen los mismos nombres de relación
2. Las relaciones de arco deben ser o todas obligatorias o todas opcionales
3. Un arco pertenece a una sola entidad, y solamente debe incluir relaciones originadas de esta entidad
4. Una entidad puede tener múltiples arcos, pero una relación específica, solamente puede participar en un sólo arco

Estándares Para DJF

1. Los nodos finales van a ser distribuidos entre 7 + -2

Estándares De subtipos

1. Para la lectura de subtipos
Cada SUPERENTIDAD debe ser SUBTIPOS debe/puede...
Cada SUBTIPO es un tipo de SUPERTIPO...
2. Agregar un subtipo OTROS

Normalización

- 1F Todos los atributos no deben ser multivalor
- 2F Todos los atributos deben depender del UID completo
- 3F Un atributo no UID no deben depender de otro no UID

f) DISEÑO

- *Teoría de diseño*

El diseño es la siguiente etapa después del análisis, en el diseño las entidades se convierten en tablas, los atributos en columnas, los identificadores únicos en llaves primarias y las relaciones en llaves foráneas, las marcas de atributos de la siguiente manera # a PK, o a N, * a NN.

Conceptos De Base De Datos Relacional

TABLA (matriz de renglones y columnas)

Ejemplo: EMPLEADOS

No_Emp	Apellido	Nombre	No_Depto
100	Soto	Juan	10
310	Alba	José	15

Una B.D. relacional debe tener integridad de datos, es decir sus datos deben ser consistentes.

- *Llaves primarias*

Una llave primaria (PK) es una columna o grupo de columnas que identifican de manera única a cada renglón en una tabla. Cada tabla debe tener una llave primaria.

Ejemplo:

llave primaria



No_Emp	Apellido	Nombre	No_Depto
100	Soto	Juan	10
310	Alba	José	15

Una llave primaria que consta de múltiples columnas se llama llave primaria compuesta.

Las columnas de una llave primaria compuesta deben de ser únicas en combinación. Las columnas pueden tener duplicados en forma individual, pero en combinación, no se permiten duplicados. Ninguna parte de la llave primaria puede ser NULA.

Ejemplo de llave primaria compuesta para la Tabla CUENTAS

NO_BANCO, NO_CUENTA

Los UID secundarios corresponden a llaves alternas en el diseño y en la construcción se intercalan su existencia (debido a que no se pueden tener 2 PK al mismo tiempo).

Ejemplo: No_emp (PK)

No_nomina (PK2)

- *Llaves foráneas*

Una llave foránea (FK) es una columna o combinación de columnas en una tabla, que se refieren a una llave primaria en otra tabla.

Ejemplo

TABLA EMPLEADOS

llave primaria			llave foránea	
NO_EMP	APELLIDO	NOMBRE	NO_DEPTO	NOMINA

TABLA DEPARTAMENTOS

llave primaria	
NO_DEPTO	NOMBRE

Las llaves foráneas son utilizadas para hacer "JOIN" entre tablas

- *Integridad de datos*

Exactitud y consistencia de los datos Los Constraints (Restricciones) de integridad de datos aseguran que los usuarios realicen únicamente operaciones en las cuales dejarán a la B.D. en un estado correcto y consistente.

Tipo de Constraint	Explicación
Integridad de Entidades	Ninguna parte de la llave primaria puede ser NULA
Integridad Referencial	Una llave foránea debe coincidir con un valor de una llave primaria
Integridad de Columnas	Una columna debe contener sólo valores consistentes con el formato de datos definido para la columna
Integridad definida por el Usuario	Los datos almacenados en la B.D. deben cumplir con las reglas del negocio

Ejemplos de constraints de integridad definida por el usuario:

- i. A un empleado de confianza no se le paga por las primeras 5 horas de trabajo de tiempo extra
- ii. Ningún empleado del Departamento. de Finanzas lleva el título de Programador
- iii. La comisión de los vendedores no puede exceder del 50% de su salario

COMENTARIO: Estas restricciones hacen que los valores que entren por medio de una forma, sean los correctos.

Un dato es inconsistente si existen múltiples copias de un registro y no todas las copias han sido actualizadas. Una B.D. inconsistente provee información incorrecta o contradictoria a los usuarios.

- *Diseño inicial de la base de datos*

El diseño de B.D. se lleva a cabo por medio de dos actividades:

1. Pasar el modelo E-R a tablas
2. Refinar el diseño inicial para producir un diseño completo de la B.D.

En el diseño se definen las tablas , índices, vistas y espacio de almacenamiento.

Introducción al diseño inicial de la B.D.

Documentar cada tabla relacional en un mapa de instancias.

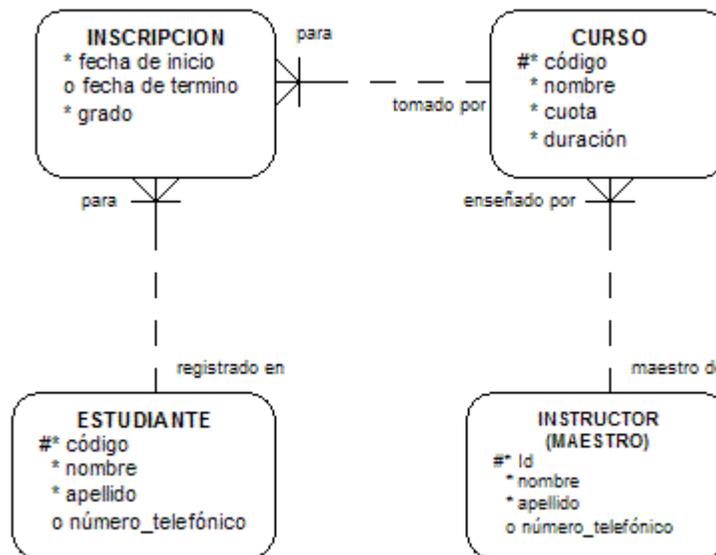
Nombre de la Tabla: EMPLEADOS

COLUMNA	NOEMP	NOMBRE	APELLIDO	PUESTO	FINGRESO	SAL	COM	GERENTE	NODEP
TIPO LLAVE	PK							FK1	FK2
NN-U	NN	NN	NN	NN	NN	NN			NN
EJEMPLOS	1	JOSE	LOPEZ	ARCHIVISTA	10-MAR-90	1000		7920	20
.

Usar sufijos para distinguir entre múltiples columnas FK en una tabla, por ejemplo, FK1 y FK2. Etiquetar múltiples columnas con el mismo sufijo.

Usar NN para una columna que debe ser definida como NO NULA.

Usar U para la columna que debe ser UID, Etiquetar múltiples columnas con el mismo sufijo, por ejemplo U1.



Pasos En El Diseño Inicial De La Base De Datos

1. Mapear las entidades para las tablas
 Crear un mapa de instancias para la nueva tabla. Registrar únicamente el nombre de la tabla

Ejemplo:

NOMBRE DE LA TABLA: INSTRUCTORES

COLUMNA				
TIPO DE LLAVE				
NULOS/UNICOS				
EJEMPLO				

2. Mapear atributos para columnas y establecer los atributos obligatorios para columnas NO NULAS (NN). Documentar datos de ejemplo
 Ejemplo: NOMBRE DE LA TABLA: INSTRUCTORES

COLUMNA	INST_ID	NOMBRE	APELLIDO	NOTEL
TIPO DE LLAVE				
NULOS/UNICOS	NN	NN	NN	
EJEMPLO	10	Juan	Pérez	7812251
	18	Maria	López	8881196

Para Cada Atributo Seleccionar Un Nombre Corto Pero Significativo

El nombre de las columnas debe ser fácil de identificar en un modelo E-R.

No utilizar palabras reservadas de SQL para nombres de columnas por ejemplo

NUMBER

Utilizar abreviaciones consistentes que no causen confusión como NUMERO = NUM ó

DEPARTAMENTO = DEPTNO DEPTNUM

Los Nombres De Columnas Cortos O Pequeños Reducirán El Tiempo Requerido Para El Comando de SQL "parsing"

Las INSTANCIAS se pueden obtener de notas de entrevistas, otros documentos de análisis.

3. Mapear identificadores únicos a llaves primarias y agregar columnas de foráneas con sus respectivos ejemplos.

NOMBRE DE LA TABLA: INSTRUCTORES

COLUMNA	INST_ID	NOMBRE	APELLIDO	NOTEL
TIPO DE LLAVE	PK			
NULOS/UNICOS	NN-U	NN	NN	
EJEMPLO	10	Juan	Pérez	7812251
	18	Maria	López	8881196

Si una entidad incluye una relación, agregar columnas de llaves foráneas para la tabla y señalarlas como parte de la llave primaria

Ejemplo:
 NOMBRE DE LA TABLA: INSCRIPCIONES

COLUMNA	FECHA_INI	FECHA_TER	GRADO	CODIGO_CUR	INS_ID	ID_EST
TIPO DE LLAVE				PK, FK1	PK, FK1	PK, FK2
NULOS/UNICOS	NN-U			NN-U	NN-U	NN-U
EJEMPLO	20-JUL-91	29-JUL-91	344	10	47593
.

Si existen múltiples columnas FK en una tabla, usar sufijos para distinguirlos, por ejemplo FK1 y FK2.
 Etiquetar múltiples columnas llave con el mismo sufijo.

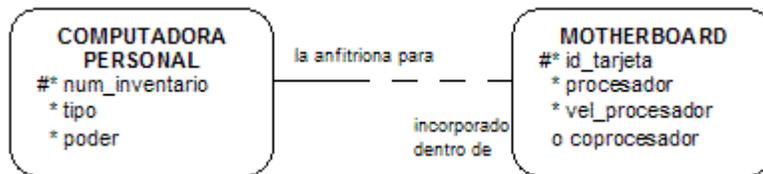
4. Mapear relaciones a llaves foráneas
 Para una relación de entidades M:1 tomar el PK de la tabla (1) y ponerlo en la tabla (M)
 Ejemplo:

Poner el PK INST_ID de la tabla 1 y ponerlo en la tabla M CURSOS

NOMBRE DE LA TABLA: CURSOS

COLUMNA	CODIGO_CURSO	NOMBRE	CUOTA	DURACION	INST_ID
TIPO LLAVE	PK				PK, FK
NULOS/UNICOS	NN-U	NN	NN	NN	NN-U
EJEMPLOS	344	SQL * FORMS	1000	5	81

Para una relación 1:1 obligatoria, de un solo lado colocar las FK como únicas en el mapa de instancias en donde la relación es obligatoria y usar el constraint NO NULO para forzar la condición obligatoria
 Ejemplo:



NOMBRE DE LA TABLA: COMPUTADORAS_PERSONALES

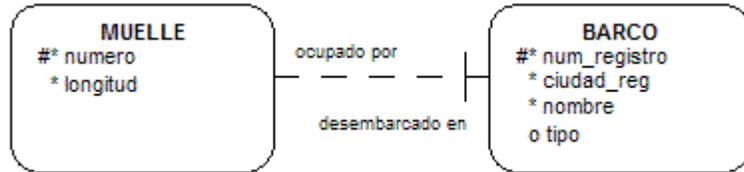
COLUMNA	NUM_INV	TIPO	PODER	ID_TAR
TIPO LLAVE	PK			FK
NULOS/UNICOS	NN-U	NN	NN	NN
EJEMPLOS	1045	tower	150	4579

NOMBRE DE LA TABLA: MOTHERBOARDS

COLUMNA	ID_TAR	PROC	VEL_PROC	COPRO
TIPO LLAVE	PK			
NULOS/UNICOS	NN-U	NN	NN	N
EJEMPLOS	9978	486	33	Y

Si una relación 1:1 es opcional en ambas direcciones mover la FK a cualquiera de la relación.

Ejemplo:



NOMBRE DE LA TABLA: MUELLES

COLUMNA	M_NUM	LON
TIPO DE LLAVE	PK	
NULOS/UNICOS	NN-U	NN
EJEMPLOS	344	100

NOMBRE DE LA TABLA: BARCOS

COLUMNA	NUM_REG	CIU_REG	NOMBRE	TIPO	M_NUM
TIPO LLAVE	PK				PK,FK
NULOS/UNICOS	NN-U	NN	NN		NN-U
EJEMPLOS	134X89	PANAMA	SUN KING	CRUISE	344

Para una relación recursiva 1:M, agregar una columna FK a la tabla. Esta columna FK debe referenciar valores de la columna PK.

Ejemplo:



NOMBRE DE LA TABLA: EMPLEADOS

COLUMNA	EMP_ID	NOMBRE	APELLIDO	GER_ID
TIPO LLAVE	PK			FK
NULOS/UNICOS	NN-U	NN	NN	
EJEMPLOS	1000	JOSE	SUAREZ	7450
7450

Para una relación recursiva 1:1, agregar una FK única a la tabla. Esta columna FK debe referenciar un valor de la columna PK.

Ejemplo:



NOMBRE DE LA TABLA: PERSONAS

COLUMNA	PERS_ID	NOMBRE	APELLIDO	ID_ESP
TIPO Llave	PK			FK
NULOS/UNICOS	NN-U	NN	NN	
EJEMPLOS	7450	MARIA	SUAREZ
	5579	SUSANA	LOPEZ	9450

5. Diseño Avanzado

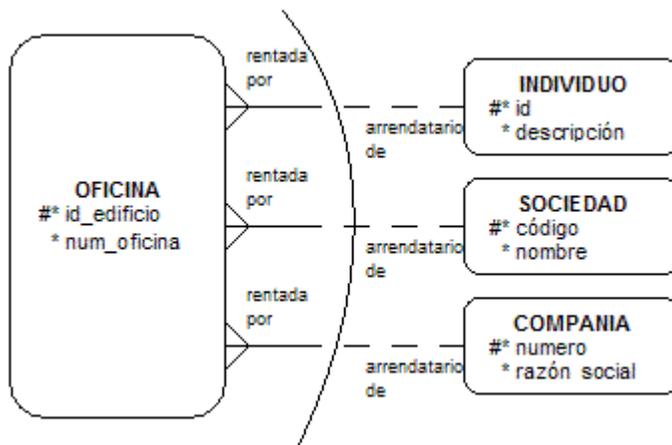
Los arcos representan un tipo de llave foránea de alternativa múltiple (Excluyente).

Alternativas de Diseño

- Diseño de Arco Explícito
- Diseño de Arco Genérico

Los arcos solamente pueden atravesar el final de las relaciones que son o todas obligatorias o todas opcionales. El diseño de Arco Explícito soportará llaves foráneas múltiples con diferentes formatos.

Ejemplo: Este Modelo E-R mapeará 4 tablas. La entidad OFICINA tiene un arco que atraviesa los finales de 3 relaciones, y las correspondientes columnas de FK deben ser agregadas a la tabla OFICINA. Usar un Diseño de Arco Explícito o un Diseño de Arco Genérico para agregar estas múltiples alternativas de llaves foráneas.



NOMBRE DE LA TABLA: OFICINAS

COLUMNA	ID_EDIF	NUM_OFIC	ID_IN	CO_SOC	NU_CIA
TIPO Llave	PK	PK	FK1	FK2	FK3
NULOS/UNICOS	NN-U	NN-U			
EJEMPLO	1024	101	30045		
	512	210		A4431	
	977	144			10844

El diseño de Arco Genérico crea una columna de llave foránea y una columna para marcar la relación del arco (distintivo). Solamente un valor de FK existirá para cada renglón en la tabla, para indicar cual de las 3 tablas está referenciada por la columna FK en cada renglón (así se crea un catálogo nuevo de tipos de arrendatario).

Este Modelo E-R mapeará 3 tablas.

NOMBRE DE LA TABLA: OFICINAS

COLUMNA	ID_EDIF	NUM_OFIC	ARREN_INQUILINO	TIP_ARREN
TIPO LLAVE	PK	PK	FK	FK
NULOS/UNICOS	NN-U	NN-U	NN	NN
EJEMPLO	1024	101	30045	1
	512	210	A4431	2
	977	144	10844	3

NOMBRE DE LA TABLA: TIPOS_ARRENDATARIOS

TIP_ARREN	DESCRIPCION
1	INDIVIDUAL
2	SOCIEDAD
3	COMPAÑÍA

Una solución es homologar el ARRENDATARIOS, de otra forma no habría llave foránea y por programación se protegería la información.

Las llaves foráneas deben de compartir el mismo formato para todas las tablas referenciadas.

- Elegir opciones de subtipo.
Opciones de Mapeo de Subtipos a Tablas
 - Diseño de una sola Tabla
 - Diseño de Tablas separadas
 - Implementación del Arco

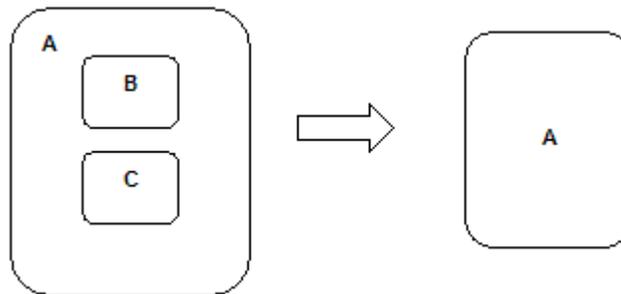
Ejemplo: En el siguiente supertipo/subtipo, las entidades EMPLEADO, EMPLEADO ASALARIADO y EMPLEADO POR HONORARIOS pueden ser mapeadas en una, o dos tablas, dependiendo de la opción seleccionada.



- Los subtipos heredan todos los atributos del supertipo y relaciones.
- Los subtipos pueden tener atributos y relaciones propias.
- Los subtipos deben de ser mutuamente excluyentes.

Opción 1.- Diseño de Subtipo de una Sola Tabla

Mapea los subtipos dentro de una sola tabla para el supertipo. La tabla sencilla contendrá instancias de todos los subtipos. Se deberá crear otra tabla (catálogo) para el distintivo.



Se usa el diseño de una sola tabla cuando el supertipo tiene pocas relaciones y atributos propios

Pasos de Diseño

1. Crear una sola tabla para el supertipo
2. Crear una columna TIPO para identificar a que subtipo pertenece cada renglón y agregar un catálogo para los distintivos.
3. Crear una columna para cada uno de los atributos del supertipo
4. Crear una columna para cada uno de los atributos del subtipo
5. Crear columnas FK para cada una de las relaciones del supertipo
6. Crear columnas FK para cada una de las relaciones del subtipo

NOMBRE DE LA TABLA: EMPLEADOS

COLUMNA	CVE	NOMBRE	APELLIDO	TIP_EMP	SAL	CTO_HRS	CTO_EX	CLA_SIN	COD_DEPT
LLAVE	PK			FK3				FK1	FK2
N/U	NN-U	NN	NN	NN	N	N	N	N	NN
EJEMPLOS	4579	Jaime	López	1	29000				40
	7147	Juan	Pérez	2		8.5	12.75	201	35

NOMBRE DE LA TABLA: TIPO EMPLEADOS

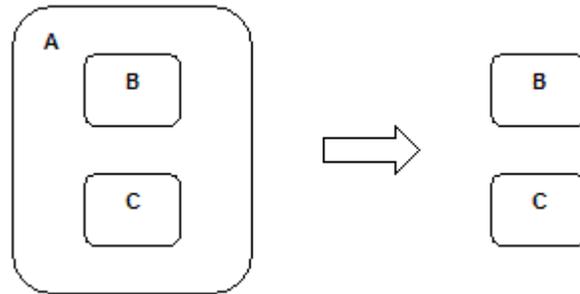
TIPO	DESCRIPCIÓN
1	Empleado Asalariado
2	Empleado Por Honorarios

Las columnas de la tabla EMPLEADOS son derivadas de los atributos y las relaciones del supertipo y de todos sus subtipos.

Tipo de Entidad	Columnas para Atributos	Columnas FK para Relaciones
Supertipo	NO_CLAVE NOMBRE APELLIDO	COD_DEPTO
Subtipo1	SALARIO	NUM_SINDICATO
Subtipo2	CTO_HORA HR_EXTRA	

Opción 2.- Diseño de Subtipo de tabla separadas

Mapear el subtipo en tablas separadas (Uno para cada subtipo). Cada tabla contendrá solamente instancias de un subtipo.



Pasos de Diseño

1. Crear una tabla para cada subtipo
2. En cada tabla subtipo, crear columnas para los atributos del subtipo
3. En cada tabla subtipo, crear columnas para los atributos del supertipo
4. En cada tabla subtipo, crear columnas FK para las relaciones del subtipo
5. En cada tabla subtipo, crear columnas FK para las relaciones del supertipo

Ejemplo: Hacer el supertipo EMPLEADO en 2 tablas (Una para cada subtipo). Primero crear una tabla separada para el subtipo EMPLEADO ASALARIADO

NOMBRE DE LA TABLA: EMPLEADOS ASALARIADOS

Columna	NUM_CVE	NOMBRE	APELLIDO	SAL	COD_DEPT	NU_SIN
Tipo Llave	PK				FK2	FK1
Nulos/Unicos	NN-U	NN	NN	NN	NN	NN
Ejemplos	4579	Jaime	Juárez	29000	40	201

Crear una tabla separada para el subtipo EMPLEADO POR HONORARIOS

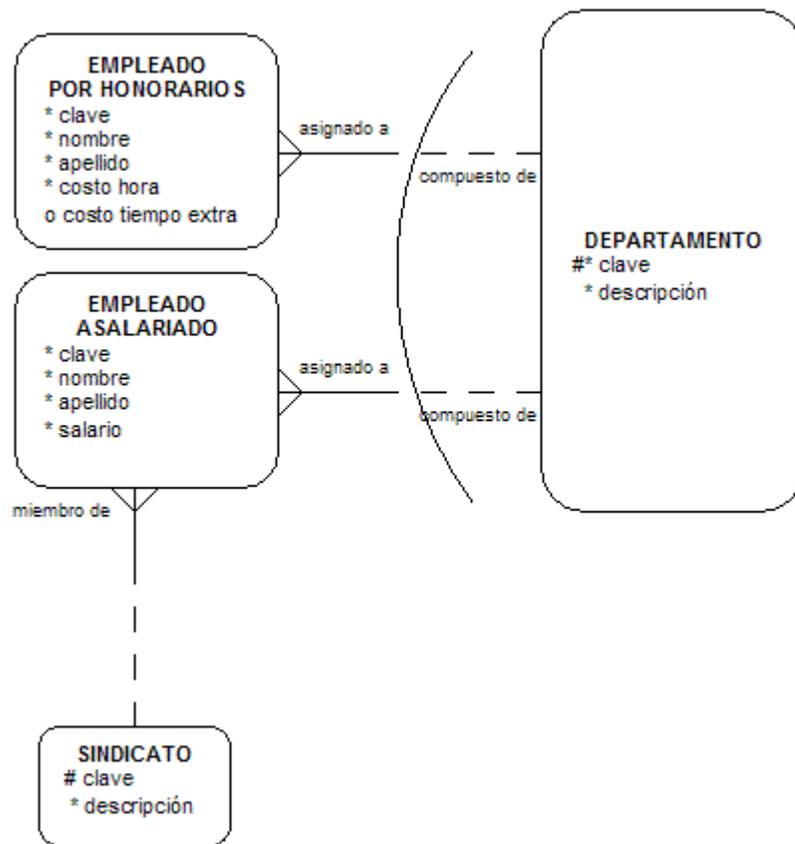
NOMBRE DE LA TABLA: EMPLEADOS POR HONORARIOS

COLUMNA	NUM_CVE	NOMBRE	APELLIDO	CTO_HORAS	H_EXT	COD_DEPT
Tipo Llave	PK					FK1
NULOS/UNICOS	NN-U	NN	NN	NN	N	NN
EJEMPLO	7147	Juan	López	8.5	12.75	35

Se usa un diseño de Subtipo de Tablas separadas cuando hay muchas relaciones y atributos de subtipos específicos.

Desventajas

- El acceso al supertipo requiere de un operador de UNION o una vista con un operador de UNION
- El mantenimiento de los UID's de los subtipos es difícil de implementar.



NOMBRE DE LA TABLA: EMPLEADOS ASALARIADOS

COLUMNA	NUM_CVE	NOMBRE	APELLIDO	SAL	COD_DEPT	NU_SIN
TIPO LLAVE	PK				FK2	FK2
NULOS/UNICOS	NN-U	NN	NN	NN	NN	NN
EJEMPLOS	4579	Jaime	Juárez	29000	40	
						201

NOMBRE DE LA TABLA: EMPLEADOS POR HONORARIOS

COLUMNA	NUM_CVE	NOMBRE	APELLIDO	CTO_HORAS	H_EXT	COD_DEPT
TIPO LLAVE	PK					FK1
NULOS/UNICOS	NN-U	NN	NN	NN	N	NN
EJEMPLO	7147	Juan	López	8.5	12.75	35

- *Normalización en diseño*

Normalización de tablas

Validar cada columna en las tablas mediante las reglas de normalización, en caso de violarlas se deben reubicar.

Regla de la Forma Normal	Descripción
Primera Forma Normal (1FN)	- La tabla no puede contener grupos de repetición
Segunda Forma Normal (2FN)	- La tabla debe de estar en 1FN. - Cada columna que no es llave debe de ser dependiente de la llave primaria como un todo.
Tercera Forma Normal (3FN)	- La tabla debe de estar en 2FN. - Una columna que no es llave primaria no debe ser funcionalmente dependiente de otra columna no llave primaria.

¿Por qué hay que normalizar las tablas?

- minimiza la redundancia de los datos. Un dato sin normalizar es redundante
- La redundancia de datos causa problemas de integridad. Las transacciones de actualización y borrado pueden no ser consistentes en todas las copias de los datos causando inconsistencia en la B.D.
- La normalización ayuda a identificar entidades, relaciones (ANALISIS) y tablas faltantes (DISEÑO).

Notas:

- La 3FN es un objetivo normalmente aceptado para un diseño de B.D. para eliminar la redundancia
- Las formas normales mayores no son utilizadas.

Reconocer Datos Sin Normalizar

Un dato sin normalizar no cumple con ninguna regla de normalización.

Ejemplo: Considerar los siguientes registros ¿Cuáles son los datos no normalizados ?

ID_ORDEN (FACTURA)	FECHA	ID_CLIENTE	NOMBRE CLIENTE	ESTADO	NUM ITEM	DESCRIP_ITEM	CANTIDAD	PRECIO
2301	6/23/97	101	MARTI	CA	3786	RED	3	35
2301	6/23/97	101	MARTI	CA	4011	RAQUETA	6	65
2301	6/23/97	101	MARTI	CA	9132	PAQ-3	8	4.75
2302	6/25/97	107	HERMAN	WI	5794	PAQ-6	4	5.0
2303	6/26/97	110	WE-SPORTS	MI	4011	RAQUETA	2	65
2303	6/26/97	110	WE-SPORTS	MI	3141	FUNDA	2	10

Contiene un grupo de repetición para NUMITEM, DESCRIP ITEM, CANTIDAD y PRECIO.

La 1FN prohíbe los grupos de repetición.

- *Conversión a la primera forma normal en diseño*

Pasos

1. Eliminar los grupos de repetición.
2. Crear una nueva tabla con la PK de la tabla base y el grupo de repetición.

Ejemplo:

ORDENES

ID_ORDEN (FACTURA)	FECHA	ID_CLIENTE	NOMBRE CLIENTE	ESTADO
PK				
2301	6/23/97	101	MARTI	CA
2302	6/25/97	107	HERMAN	WI
2303	6/26/97	110	WE-SPORTS	MI

ID_ORDEN (FACTURA)	NUM_ITEM	DESCRIP_ITEM	CANTIDAD	PRECIO
PK,FK	PK			
2301	3786	RED	3	35
2301	4011	RAQUETA	6	65
2301	9132	PAQ-3	8	4.75
2302	5794	PAQ-6	4	5.0
2303	4011	RAQUETA	2	65
2303	3141	FUNDA	2	10

- *Conversión a la segunda forma normal en diseño*

Eliminar cualquier columna no llave que no dependa de la llave primaria completa de la tabla

Pasos

1. Determinar cuales columnas que no son llave no dependen de la llave primaria completa de la tabla
2. Eliminar esas columnas de la tabla base
3. Crear una segunda tabla con esas columnas y la(s) columna(s) de la PK de la cual dependen

Ejemplo: Poner la siguiente tabla en 2FN

ORDENES

ID_ORDEN	FECHA	ID_CLIENTE	NOMBRE_CLIENTE	ESTADO
PK				
2301	6/23/97	101	MARTI	CA
2302	6/25/97	107	HERMAN	WI
2303	6/26/97	110	WE-SPORTS	MI

La tabla ORDENES está en 2FN. Cualquier valor único de ID ORDEN determina un sólo valor para cada columna. Por lo tanto, todas las columnas son dependientes de la llave primaria ID_ORDEN.

Nota: Cualquier tabla con una llave primaria de una sola columna está automáticamente en la 2FN

Ejemplo: Poner la siguiente tabla en la 2FN

ARTICULOS_ORDENES

ID_ORDEN	NUM_ITEM	DESCRIP_ITEM	CANTIDAD	PRECIO
PK,FK	PK			
2301	3786	RED	3	35
2301	4011	RAQUETA	6	65
2301	9132	PAQ-3	8	4.75
2302	5794	PAQ-6	4	5.0
2303	4011	RAQUETA	2	65
2303	3141	FUNDA	2	10

Esta tabla no se encuentra en 2FN ya que PRECIO y DESCRIP_ITEM son dependientes de NUM_ITEM, pero no son dependientes de ID ORDEN.

Eliminar parcialmente cualquier columna dependiente. Crear una tabla ARTICULOS con estas columnas y la columna de la PK de la que dependen.

ARTICULOS_ORDENES

ID_ORDEN	NUM_ITEM	CANTIDAD
PK,FK	PK,FK	
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

ARTICULOS

NUM_ITEM	DESCRIP_ITEM	PRECIO
PK		
3786	RED	35
4011	RAQUETA	65
9132	PAQ-3	4.75
5794	PAQ-6	5.0
3141	FUNDA	10

- *Conversión a la tercera forma normal en diseño*

Eliminar cualquier columna no llave que sea dependiente de otra columna no llave
Pasos

1. Determinar que columnas son dependientes de otra columna no llave
2. Eliminar esas columnas de la tabla base
3. Crear una Segunda Tabla con esas columnas y con la columna no llave de la cual son dependientes.

Ejemplo: Poner la tabla ORDENES en 3FN

ORDENES

ID_ORDEN	FECHA	ID_CLIENTE	NOMBRE_CLIENTE	ESTADO
PK				
2301	6/23/97	101	MARTI	CA
2302	6/25/97	107	HERMAN	WI
2303	6/26/97	110	WE-SPORTS	MI

NOMBRE CLIENTE y ESTADO son dependientes de ID CLIENTE. ID CLIENTE no es la PK. Por eso, la tabla ORDENES no está en 3FN.

Mover las columnas no llave dependientes con la columna no llave de la cual son dependientes dentro de una nueva tabla CLIENTES

ORDENES

ID_ORDEN	FECHA	ID_CLIENTE
PK		
2301	6/23/97	101
2302	6/25/97	107
2303	6/26/97	110

CLIENTES

ID_CLIENTE	NOMBRE_CLIENTE	ESTADO
PK		
101	MARTI	CA
107	HERMAN	WI
110	WE-SPORTS	MI

Ejemplo:

ARTICULOS_ORDENES

ID_ORDEN	NUM_ITEM	CANTIDAD
PK,FK	PK,FK	
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

¿Está en 3FN? ¿Por qué si o por qué no?. La Tabla ARTICULOS_ORDENES está en 3FN; todos los atributos no llave son dependientes solamente de la llave.

- *Definir la integridad referencial*

Puntos Adicionales En El Diseño De Base De Datos

1. Especificar La Integridad Referencial

El valor de una columna de llave foránea debe "coincidir" con un valor existente en otra columna que sea llave primaria (o ser NULO).

Constraint de borrado

- Qué sucede si un renglón que contiene una llave primaria referenciada es borrada?

Constraint de actualización

- Qué sucede si una llave primaria referenciada es actualizada?

Especificar el Constraint de borrado para definir qué debe suceder si un renglón que contiene a una llave primaria referenciada es borrada.

OPCIONES: CASCADE, RESTRICTED o NULLIFY (Sólo si se permiten NULOS)

Ejemplo: Considerar las tablas EMPLEADOS y DEPARTAMENTOS. ¿Qué deberá suceder si el NO_DEPT para el cual el empleado trabaja es actualizado de la tabla DEPARTAMENTOS?

Tabla EMPLEADOS

COLUMNA	NO_EMP	NOM_EMP	NO_DEPT_EMP
TIPO LLAVE	PK		FK
NULOS/UNICOS	NN-U		NN
EJEMPLO	100	Suárez	10
	310	Almada	30
	210	Gómez	20
	405	Juárez	50
	378	Vázquez	50

Tabla DEPARTAMENTOS

COLUMNA	NO_DEPT	NOM_DEPT
TIPO LLAVE	PK	
NULOS/UNICOS	NN-U	NN
EJEMPLO	10	Contabilidad
	20	Ventas
	40	Finanzas
	50	Operaciones

Opción Explicación del Constraint

-
- CASCADE El borrado deberá ser en cascada para todos los empleados que coincidan. Los renglones que coincidan de EMPLEADOS también deberán ser borrados.
 - RESTRICTED El borrado deberá restringirse sólo a DEPARTAMENTOS sin empleados.
 - NULLIFY Se deberá colocar un valor NULO a la llave foránea (válido sólo para FK's que permiten nulos) Cuando la PK referenciada es borrada.

Especificar el Constraint de actualización para definir qué debe suceder si un renglón que contiene a una llave primaria referenciada es actualizada.

OPCIONES : CASCADE, RESTRICTED o NULLIFY (Sólo si se permiten NULOS).

Ejemplo: Considerar las tablas EMPLEADOS y DEPARTAMENTOS. ¿Qué deberá suceder si el NO_DEPT para el cual el empleado trabaja es actualizado de la tabla DEPARTAMENTOS?

Opción	Explicación del Constraint
CASCADE	La actualización deberá hacerse en cascada para todos los empleados que coincidan. Los renglones que coincidan de EMPLEADOS también deberán ser actualizados para reflejar el nuevo valor de PK.
RESTRICTED	La actualización deberá restringirse sólo a DEPARTAMENTOS sin empleados
NULLIFY	Se deberá colocar un valor NULO a la llave foránea (válido sólo para FK's que permiten nulos) Cuando la PK referenciada es actualizada a un nuevo valor de PK.

Constraint De Dominio

Este tipo de constraints son usados para asegurar reglas simples de negocio sobre el contenido de los datos en las tablas.

Los check pueden referenciar a otras columnas en la fila que está siendo revisada, pero no pueden referenciar a otras filas o a otras tablas, o llamar a funciones como SYSDATE, UID, USER o USERENV.

No se puede asociar un check a una columna de tipo LOB, object, nested table, VARRAY o REF. Una columna puede estar protegida por más de un constraint y un check puede proteger a más de una columna.

La siguiente tabla CG_REF_CODES contiene el rango de valores de cada dominio.

```
SQL> CREATE TABLE CG_REF_CODES
2> (RV_DOMAIN, VARCHAR2(100) NOT NULL,
3> RV_LOW_VALUE, VARCHAR2(240) NOT NULL,
4> RV_HIGH_VALUE, VARCHAR2(240) NULL,
5> RV_ABBREVIATION, VARCHAR2(240) NULL,
6> RV_MEANING VARCHAR2(240) NULL);

SQL> Insert into CG_REF_CODES
2> (RV_DOMAIN,
3> RV_LOW_VALUE,
4> RV_HIGH_VALUE,
5> RV_ABBREVIATION,
6> RV_MEANING);
7> VALUES ('CONFIRMACION',
8> 'S',
9> NULL,
10> 'SI',
11> 'SI');

SQL> ALTER TABLE X ADD CONSTRAINT NAME CHECK
2> (COLUMN IN ('S', 'N'));
```

- *Diseño de índices*

Un índice es asociado con una sola tabla y contiene los valores de una o más columnas de esa tabla.

Tabla: CURSOS

COLUMNA	COD_CURSO	NOMBRE	DURACIÓN	CUOTA	ID_INST
TIPO LLAVE	PK				FK
NULOS/UNICOS	NN-U	NN	NN	NN	NN
EJEMPLO	344	SQL *Forms	5	1000	81
	974	SQL *RW	2	400	73
	401	DB DESIGN	2	400	95
	717	DBA	3	900	73
	659	SQL *Plus	2	400	301

Tabla: CURSOS

Pseudocolumna IDREN (rowid)

IDREN	COD_CURSO	NOMBRE	DURACIÓN	CUOTA	ID_INST
1010	344	SQL *Forms	5	1000	81
1011	974	SQL *RW	2	400	73
1012	401	DB DESIGN	2	400	95
1013	717	DBA	3	900	73
1014	659	SQL *Plus	2	400	301

Índice I_CURSOS_PRIN

(Único)

COD_CURSO	IDREN
344	1010
401	1012
659	1014
717	1013
974	1011

Índice I_CURSOS_2

(No único)

ID_INST	IDREN
73	1011
73	1013
81	1010
95	1012
301	1014

Usar índices para mejorar considerablemente el acceso de datos.

Índices

- Proveen acceso rápido a los renglones de datos y evitan búsquedas en toda la tabla
- Facilitan uniones (join) entre tablas
- Aseguran que no existe un valor duplicado si se define como único
- Son usados automáticamente cuando se referencian en las cláusulas WHERE de las instrucciones de SQL

Un índice concatenado es un índice creado con un grupo de columnas.

Ejemplo: Tabla INSCRIPCIONES

COLUMNA	EST_ID	COD_CURSO	FEC_INIC	FEC_TERM	GRADO
TIPO LLAVE	PK, FK1	PK, FK2			
NULOS/UNICOS	NN-U	NN-U	NN		
EJEMPLO	47592	344	20-JUL-91	19-JUL-91	A
	15402	401	05-SEP-91	B
	51394	717	14-JUN-91	20-JUL-91	A
	94572	717	08-MAY-91	28-JUL-91	B
	51394	401	05-MAY-91	21-MAY-91	A

Tabla física INSCRIPCIONES (Aquí no importa orden)

ID_REN	FEC_INIC	FEC_TERM	GRADO	COD_CURSO	EST_ID
5011	20-JUL-91	19-JUL-91	A	344	47592
5012	05-SEP-91	B	401	15402
5015	14-JUN-91	20-JUL-91	A	717	51394
5013	08-MAY-91	28-JUL-91	B	717	94572
5014	05-MAY-91	21-MAY-91	A	401	51394

Índice I_INSCRIPCION_PRN
(Única)

COD_CURSO	EST_ID	ID_REN
344	47592	5011
401	15402	5012
401	51394	5014
717	51394	5015
717	94572	5013

Crear índices para:

- Llaves Primarias (índices únicos)
- Llaves Foráneas (Generalmente índices no-únicos)

Considerar la indexación para:

- Llaves alternativas (índices únicos)
- En columnas no-llaves críticas muy usadas en las cláusulas WHERE
- Cualquier llave de búsqueda

Los índices requieren espacio y generan una sobrecarga al momento de hacer actualizaciones

- *Establecer vistas*

Una vista es un objeto de B.D. para tener un subconjunto de información de una o más tablas, mediante un SELECT

Establecer las vistas de la B.D. para cubrir los requerimientos de acceso.

Las vistas se pueden utilizar para:

- Restringir accesos
- Presentar las tablas a los usuarios en cualquier forma
- Pre-empacar consultas complejas
- Producir prototipos rápidos en SQL*Forms
- Checar la entrada de datos, cuando la vista es creada por una sola tabla.

Se puede ver como una ventana predefinida en una B.D.

Notas:

- Una vista no tiene datos propios y únicamente cuenta con información de tablas
- Una vista es definida por una instrucción SELECT a la cual se pone un nombre y es almacenado en el diccionario de datos
- Una vista es consultada como si fuera tabla.

Una vista puede restringir lo que ve el usuario, el diseñador o la herramienta.

Ejemplo: Una vista de la tabla EMPLEADOS puede evitar ver los salarios de empleados

Tabla Base: EMPLEADOS

EMP_NO	NOMBRE	EMP_GER	SAL
101	JUAREZ	50	55,010

Vista: V_EMPLEADOS

EMP_NO	NOMBRE	EMP_GER
101	JUAREZ	50

Una vista puede usarse para presentar datos normalizados en forma desnormalizada

Ejemplo:

ORDENES

NUM_ORD	FECHA_ORD	ORD_NO_CLI
10145	10-JUN-91	111
32157	08-SEP-91	342
43010	30-JUL-91	111
27444	07-AUG-91	550
98710	10-MAY-91	375

CLIENTES

NO_CLIEN	NOM_CLI	APELLIDO_CLI
111	SAMUEL	JUAREZ
342	JUAN	DIAZ
550	ROSA	LARA
375	JOSE	LOPEZ

Una vista definida entre las 2 tablas puede ser usada para hacer una pre-uni6n (pre-join) entre las tablas, as6 el usuario s6lo podr6 ver una sola tabla.

Vista: V_PEDIDOS

NUM_ORD	FECHA_ORD	ORD_NO_CLI	NOM_CLI	APELLIDO_CLI
10145	10-JUN-91	111	SAMUEL	JUAREZ
32157	08-SEP-91	342	JUAN	DIAZ
43010	30-JUL-91	111	SAMUEL	JUAREZ
27444	07-AUG-91	550	ROSA	LARA
98710	10-MAY-91	375	JOSE	LOPEZ

Usar las vistas con precaución. El acceso a través de las vistas es más lento porque se requiere un acceso extra al diccionario de datos y puede causar que la optimización de la consulta sea más lenta.

Limitaciones de las vistas

- Para una vista basada en una sola tabla, los comandos INSERT, UPDATE y DELETE de SQL no tienen limitaciones
- Para vistas de múltiples tablas con columnas virtuales, INSERT, UPDATE y DELETE están restringidos
- *Desnormalización de la base de datos*

Siempre comenzar con tablas en Tercera Forma Normal. ¡Hay que ser cuidadosos con la desnormalización ! Porque puede causar problemas de inconsistencia de datos.

La desnormalización puede ser la solución para transacciones con requerimientos de performance como Tiempo de respuesta rápido.

Considerar otras opciones antes de la desnormalización, especialmente el agregar o cambiar la estructura de los índices.

Combinar las tablas es la forma en que desnormalizamos.

Ejemplo : Considerar las tablas CUENTAS y BANCOS.

Tabla: CUENTAS

COLUMNA	NUM_CUENTA	BALANCE	NUM_BANCO
TIPO Llave	PK		PK, FK
NULOS/UNICOS	NN-U	NN	NN-U
EJEMPLO	04157	100	301
	75731	2500	410
	61134	9791	301
	94174	3740	501
	53109	5949	410

Tabla: BANCOS

COLUMNA	NUM_BANCO	NOM_BANCO
TIPO LLAVE	PK	
NULO/UNICO	NN-U	NN
EJEMPLO	301	UNITED BANK
	410	HARVARD TRUST
	501	AMERICAN
	257	FIRST NATIONAL
	380	SECOND NATIONAL

Si las consultas de las cuentas de gran volumen siempre accesan el nombre del banco, una tabla combinada puede ser la solución pero generará redundancia de datos.

Tabla : CUENTAS2

COLUMNA	NUM_BANCO	NUM_CUEN	BALANCE	NOM_BANCO
TIPO Llave	PK	PK		
NULOS/UNICOS	NN-U	NN-U	NN	NN
EJEMPLO	301	04157	100	UNITED BANK
	410	75731	2500	HARVARD TRUST
	301	61134	9791	UNITED BANK
	501	94174	3740	AMERICAN
	410	53109	5949	HARVARD TRUST

Ejemplo: Las sig. tablas de códigos se requieren para un sistema de aplicación. Son usados para proveer la lista de valores de SQL*Forms y para validar los valores de las tablas para INSERT o UPDATE.

COD_COLORES

CODIGO	DESCRIPCION
010	amarillo
020	Oro
030	Rojo
050	Gri

COD_REGIONES

CODIGO	DESCRIPCION
C	Central
E	Este
N	Norte

COD_ESTAT_REPARACIONES

CODIGO	DESCRIPCION
R	reparable
I	irreparable
E	Espera en reparación

COD_FABRICA

CODIGO	DESCRIPCION
RD	Centro R&D
CR	Centro de Reparación
A	Almacén

Combinar todas las tablas en una sola tabla con una columna adicional, TIPO_COD, que define a que grupo de valores pertenece el código. Crear una tabla con cada TIPO_COD para validar la longitud máxima de las descripciones de los códigos.

CODIGOS

CODIGO	DESCRIPCION	TIPO_COD
010	Amarillo	1
C	Central	2
R	Reparable	3
020	Oro	1
E	Este	2
030	Rojo	1
RD	Centro R&D	4
N	Norte	2
CR	Centro de Reparación	4
050	Gris	1

Tabla: TIPOS_CODIGOS

TIPO_COD	DESCRIPCION
1	Color
2	Región
3	Estado reparación
4	Fábrica

Información Por Cada Año De Los Diferentes Modelos De Autos

Un vector es un arreglo de una dimensión con un número compuesto de valores (un grupo repetido de un tamaño definido). Representar un vector de datos como un conjunto de renglones o un conjunto de columnas.

Diseño de Tabla de Vectores como Columnas (3FN)

MODEL	MES	CANTIDAD
COUPE	ENE	1000
COUPE	FEB	1001
COUPE	MAR	1002
COUPE	ABR	1003
COUPE	MAY	1004
COUPE	JUN	1010
COUPE	JUL	1001
COUPE	AGO	1005
COUPE	SEP	1004
COUPE	OCT	1003
COUPE	NOV	1001
COUPE	DIC	1000

Diseño de Tabla de Vectores como Renglones

MODEL	ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	NOV	DICE
COUPE	1000	1001	1002	1003	1004	1010	1001	1005	1004	1003	1001	1000
VAGON	850	850	852	853	854	860	861	865	864	863	851	850
CONVE												

Escoger el diseño de tabla para un vector de datos se basa en los requerimientos de acceso.

Ventajas de incluir el vector como columnas

1. Las funciones de grupo de SQL actúan en las columnas, por Ejemplo: SUM,AVG

Ventajas de incluir el vector como renglón

1. El requerimiento de espacio de almacenamiento es menor
2. Los reportes de salida con valores horizontales son más fáciles de producir (En este ejemplo si van por año)

Reconsiderar el almacenar datos derivados para apoyar a que sean funcionales los requerimientos del acceso.

Ejemplo: Un vendedor regional tiene 200 vendedores a su cargo. El frecuentemente consulta las cuotas de ventas y las ventas actualizadas por cada región. Las cuotas de ventas se establecen trimestralmente. El dato de ventas es actualizado semanalmente.

Tabla: VENDEDORES

COLUMNA	ID	APELLIDO	NOMBRE	CUOTA	VENTAS	NUM_REG
TIPO DE LLAVE	PK					FK
NULOS/UNICOS	NN-U	NN	NN	NN	NN	NN
EJEMPLO	110	Suárez	Juan	500000	10000	50
	250	Juárez	Tomas	700000	350000	35
	310	González	Maria	1200000	500000	40
	400	Díaz	Martha	650000	250000	37
	550	Bravo	María	800000	420000	40

Tabla: VENTAS

COLUMNA	NUM_REGION	CUOTA_VENTAS	VENTAS
TIPO DE LLAVE	PK		
NULOS/UNICOS	NN-U		
EJEMPLO	50	500000	10000
	35	700000	350000
	40	2000000	920000
	37	650000	250000

Datos Consolidados, Derivados

NOTA: Mediante TRIGGER se garantiza la información actualizada.

- *Planeación de uso del almacenamiento físico*

Se deberá planear con el DBA para índices y tablas.

- Se realizará una estimación de cada tabla e índice
- Decidir la localización del lugar de las tablas e índices en tablespaces lógicos y en discos físicos separados
- Definir parámetros de actualización de crecimiento.

g) CONSTRUCCIÓN

- *Integridad referencial*

La integridad referencial es un sistema de reglas que relacionan las entidades entre sí a través de un campo, convirtiéndose éste en llave primaria o foránea; garantizando así que los datos sean correctos, sin repeticiones innecesarias, que no se modifiquen ni borren datos relacionados de forma accidental.

- *DDL (Data Definition Language / Lenguaje de Definición de Datos)*

Define y maneja los objetos de la base de datos:

- **Create.** El siguiente script crea una tabla llamada PRODUCTOS

```
CREATE TABLE PRODUCTOS (  
    ID_Produc int Not Null,  
    Nóm_Produc char(40),  
    Precio number(7,2))
```

- **Alter.** El siguiente ejemplo agrega una columna que permite valores NULL y a la que no se han proporcionado valores mediante una definición DEFAULT. Cada fila tendrá un valor NULL en la nueva columna

```
ALTER TABLE PRODUCTOS ADD Clasificación VARCHAR(20) NULL
```

- **Drop.** El siguiente ejemplo borra la tabla de PRODUCTOS y sus datos e índices de la base de datos actual.

```
DROP TABLE PRODUCTOS
```

Define la base de datos para crear base de datos, tablas y tipos de datos definidos por el usuario.

Por default únicamente miembros de los siguientes roles sysadmin, dbcreator, db_owner o db_ddladmin (Agrega, modifica o borra objetos de la base de datos) pueden ejecutar dichas sentencias.

En general, es recomendable que ninguna otra cuenta sea usada para crear objetos de base de datos. Si diferentes usuarios crean sus propios objetos en una base de datos, entonces cada uno de éstos objetos requiere otorgar sus propios permisos para cada usuario. Esto provoca una carga administrativa que podría ser evitada restringiendo permisos a esos roles, ésta restricción también evita problemas con objetos propios que han sido borrados de una base de datos.

Si múltiples cuentas de usuario crean objetos, los roles de sysadmin y db_owner pueden usar el SETUSER para representar a otro usuario o el system store procede sp_changeobjectowner para cambiar el propietario de un objeto.

- *DCL (Data Control Language / Lenguaje de Control de Datos)*

Se encarga de la seguridad de la Base de Datos y determina quien puede usar o modificar los datos. Configura o cambia permisos asociados a roles (Agrupación de permisos) o a usuarios de la base de datos. La siguiente tabla describe las sentencias DCL:

- **Grant.** Crea una entrada en la seguridad del sistema que permite a un usuario trabajar con datos o ejecutar ciertas sentencias Transacciones-SQL. El siguiente ejemplo otorga al rol public permisos para preguntar en la tabla PRODUCTOS:

```
GRANT Select ON PRODUCTOS TO public
```

- **Revoke.** Revoca permisos otorgados o denegados. En el siguiente se revocan varios permisos de instrucción de varios usuarios.

```
REVOKE CREATE TABLE, CREATE DEFAULT  
FROM Mary, John
```

Por default únicamente. los siguientes roles: sysadmin (Realiza cualquier actividad), dbcreator (Crea y modifica base de datos), db_owner (Realiza cualquier actividad de cualquier role de la base de datos) y db_securityadmin (Asigna sentencias y permisos de objetos) pueden ejecutar éstas sentencias.

- *DML (Data Manipulation Language / Lenguaje de Manipulación de Datos)*

Es usado para insertar, modificar, recuperar y/o eliminar los datos con las sentencias:

- **Insert.** El siguiente ejemplo inserta un registro en la tabla PRODUCTOS

```
INSERT INTO PRODUCTOS  
  (ID_Produc, Nom_Produc, Precio, Clasificación)  
VALUES  
  (1, Leche, 8, Lácteos)
```

- **Update.** El siguiente ejemplo actualiza la información de un registro en la tabla PRODUCTOS

```
UPDATE PRODUCTOS  
SET Precio = 10  
WHERE ID_Produc = 1
```

- **Select.** El siguiente ejemplo recupera los siguientes datos de la tabla PRODUCTOS

```
Select ID_Produc,  
       Nom_Produc,  
       Precio  
FROM Productos
```

- **Delete.** El siguiente ejemplo borra el registro del producto 'Leche' en la tabla PRODUCTOS

```
DELETE FROM PRODUCTOS  
WHERE Nom_Produc = 'Leche'
```

Por default únicamente miembros de los roles sysadmin, dbcreator, db_owner o db_datawriter (Agrega, modifica o borra datos de todas las tablas)

V. Administración del Tutorial

5.1 Administración de Capítulos

Capitulos Existentes

Capitulo	Modificar	Eliminar
Bienvenida		
Teoría		
Análisis		
Álgebra Relacional		
Diseño		
Construcción		
Herramientas Case		
Administración		

Esta sección nos permite:

- Alta. Para dar de alta un capitulo es necesario
 - Dar clic en el botón „Nuevo Capitulo’
 - Escribir el Nombre del Capitulo
 - Dar clic en el botón „Enviar’

Administración De Capítulos

Alta de Capitulo

Capitulo:

- Modificación. Modifica el nombre del Capitulo
 - Dar clic en el icono  del capitulo que se desea modificar
 - Escribir el nuevo nombre del capitulo
 - Dar clic en el botón „Enviar’

Administración De Capítulos

Modificación del Nombre del Capítulo

Capítulo:

- Eliminación. Elimina el capítulo.

- Dar clic en el icono  del capítulo que se desea eliminar
- Dar clic en el botón „Aceptar” del mensaje de confirmación de la eliminación

Capítulos Existentes

Capítulo	Modificar	Eliminar
Bienvenida		
Teoría		
Análisis		
Álgebra		
Diseño		
Constru		
Herram		

Microsoft Internet Explorer

¿Esta seguro de eliminar el capítulo?

5.2 Administración de Temas

Temas Existentes		
TEORÍA	Mofificar	Eliminar
Cronología de las bases de datos		
Definición de una base de datos		
Objetivos de una base de datos		
Definición de un sistema de base de datos		
Ventajas y desventajas de un sistema de base de datos		
Sistema Manejador de Base de Datos (DBMS)		
Administrador de la base de datos		
Diccionario de datos		
Estructura del diccionario de datos		
Introducción al análisis, diseño y construcción		
Teoría del modelo de datos Entidad-Relación		
ANÁLISIS	Mofificar	Eliminar
Identificar entidades		
Descripción de entidades		
Establecer las relaciones entre entidades		

Esta sección nos permite:

- Alta. Para dar de alta un tema es necesario
 - Dar clic en el botón „Nuevo Tema’
 - Seleccionar el Capitulo al que pertenecerá
 - Escribir el Nombre del Tema
 - Dar clic en el botón „Enviar’

Administración De Temas

Alta de Tema

Capitulo: 

Tema:

- Modificación. Modifica el nombre del Tema

- Dar clic en el icono  del tema que se desea modificar
- Escribir el nuevo nombre del tema
- Dar clic en el botón „Enviar“

Administración De Temas

Modificación del Nombre del Tema

Capítulo:

Tema:

- Eliminación. Elimina el tema.

- Dar clic en el icono  del tema que se desea eliminar
- Dar clic en el botón „Aceptar“ del mensaje de confirmación de la eliminación

HERRAMIENTAS CASE	Modificar	Eliminar
Metodología Case		
Fase de estrategia		
Fase de análisis		
Fase de diseño		
Construcción		
Documentación		
Transición		
Producción		

Microsoft Internet Explorer

¿Esta seguro de eliminar el tema?

5.3 Administración de Exámenes

Exámenes Existentes

Nº	Versión	Descripción	Modificar	Eliminar	Consultar	Examen Contestado
1	1	Examen de teoría y análisis Sem 2004-2			Consultar	Ver
2	1	Examen de Álgebra Relacional			Consultar	Ver
3	1	Examen de diseño			Consultar	Ver
4	1	Examen de construcción			Consultar	Ver
5	1	Examen de Herramientas Case			Consultar	Ver
7	1	prueba			Consultar	Ver

Nuevo Examen

Esta sección nos permite:

- Alta. Para dar de alta un examen es necesario
 - Dar clic en el botón „Nuevo Examen’
 - Escribir el Número de Examen
 - Escribir el Nombre del Examen
 - Dar clic en el botón „Enviar’

Administración De Exámenes

Alta de Examen	
Nº :	<input type="text"/>
Descripción:	<input type="text"/>
<input type="button" value="Enviar"/>	

- Modificación. Modifica el nombre del Examen
 - Dar clic en el icono  del examen que se desea modificar
 - Escribir el nuevo nombre del examen
 - Dar clic en el botón „Enviar“

Administración De Exámenes

Modificación del Nombre del Examen

Nº :

Descripción:

- Eliminación. Elimina el examen.
 - Dar clic en el icono  del examen que se desea eliminar
 - Dar clic en el botón „Aceptar“ del mensaje de confirmación de la eliminación

Exámenes Existentes

Nº	Versión	Descripción	Modificar	Eliminar	Consultar	Examen Contestado
1	1	Examen de teoría y análisis Sem 2004-2			Consultar	Ver
2	1	Examen de Álgebra Relacional			Consultar	Ver
3	1	Examen de diseño			Consultar	Ver
4	1	Exame			Consultar	Ver
5	1	Exame			Consultar	Ver

Microsoft Internet Explorer X

?

¿Esta seguro de eliminar el examen?

- Consultar. Muestra el examen completo y un botón para Imprimirlo.
- Examen Contestado. Muestra el examen completo con las respuestas correctas y un botón para Imprimirlo

5.4 Administración de Alumnos

Alumnos Existentes

Clave	Nombre	Apellido Paterno	Apellido Materno	Usuario	Modificar	Eliminar
1	Olga	Cisneros	Monroy	ocisneros		
2	Donovan	Estrella	Cisneros	destrella		
3	Brandon	Montoya	Cisneros	bmontoya		
4	Cintha	Cisneros	Monroy	ccisneros		
5	Ian	Estrella	Cisneros	icisneros		
6	Felipe	Cisneros	Rangel	fcisneros		
7	Maricela	Monroy	Anaya	mmonroy		

Esta sección nos permite:

- Alta. Para dar de alta un alumno-usuario es necesario
 - Dar clic en el botón „Nuevo Alumno’
 - Escribir el Nombre
 - Escribir el Apellido Paterno
 - Escribir el Apellido Materno
 - Dar clic en el botón „Enviar’

Al dar de alta un alumno se genera en automático un usuario y contraseña

Administración De Alumnos

Alta de Alumno

Nombre:

Apellido Paterno:

Apellido Materno:

- Modificación. Modifica los datos del Alumno

- Dar clic en el icono  del alumno que se desea modificar
- Escribir la información nueva
- Dar clic en el botón „Enviar”

Administración De Alumnos

Modificación de Alumno

Nombre:

Apellido Paterno:

Apellido Materno:

En automático se modifica el usuario

- Eliminación. Elimina al alumno.

- Dar clic en el icono  del alumno que se desea eliminar
- Dar clic en el botón „Aceptar” del mensaje de confirmación de la eliminación

Alumnos Existentes

Clave	Nombre	Apellido Paterno	Apellido Materno	Usuario	Modificar	Eliminar
1	Olga	Cisneros	Monroy	ocisneros		
2	Donovan	Estrella	Cisneros	destrella		
3	Brand					
4	Cinthy					
5	Ian					
6	Felipe					
7	Maric					

Microsoft Internet Explorer

¿Esta seguro de eliminar al alumno?

5.5 Administración de Tipo de Preguntas

Tipos De Preguntas Existentes

Tipo De Pregunta	Ejemplo	Modificar	Eliminar
Relacionar Columnas	Ver		
Opción Múltiple - 1 respuesta	Ver		
Opción Múltiple - 1 ó más respuestas	Ver		
Respuesta en texto			
Generación de diseño (tablas)			
Álgebra Relacional			

Esta sección nos permite:

- Alta. Para dar de alta un Tipo de Pregunta es necesario
 - Dar clic en el botón „Nuevo Tipo de Pregunta’
 - Escribir el Nombre del Tipo de Pregunta
 - Adjuntar la imagen del ejemplo del Tipo de Pregunta
 - Dar clic en el botón „Enviar’

Administración De Tipos De Preguntas

Alta de Tipo de Pregunta

Tipo De Pregunta:

Imagen:

- Modificación. Modifica el nombre del tipo de pregunta
 - Dar clic en el icono  del tipo de pregunta que se desea modificar
 - Escribir el nuevo nombre del tipo de pregunta
 - Dar clic en el botón „Enviar”

Administración De Tipos De Preguntas

Modificación del Nombre del Tipo de Pregunta

Tipo De Pregunta:

- Eliminación. Elimina el tipo de pregunta.
 - Dar clic en el icono  del tipo de pregunta que se desea eliminar
 - Dar clic en el botón „Aceptar” del mensaje de confirmación de la eliminación

Tipos De Preguntas Existentes

Tipo De Pregunta	Ejemplo	Modificar	Eliminar
Relacionar Columnas	Ver		
Opción Múltiple - 1 respuesta	Ver		
Opción Múltiple - 1 ó más respuestas	Ver		
Respuesta en texto			
Generación			
Álgebra R			

Microsoft Internet Explorer

? ¿Esta seguro de eliminar el tipo de pregunta?

5.6 Administración de Preguntas, Subpreguntas y Respuestas

Preguntas Existentes

Exam.	Ver.	Capitulo	Tema	Tipo	No.	Descripción	Puntos	Imagen	Modificar	Eliminar
1	1	Teoría	Cronología de las bases de datos	1	1	Relaciona cada pregunta con su respuesta correcta.	4			
1	1	Teoría	Cronología de las bases de datos	1	2	Relaciona cada pregunta con su respuesta correcta.	2			
1	1	Teoría	Definición de una base de datos	2	3	¿Cuándo se utiliza barra UID?	2			
1	1	Teoría	Teoría del modelo de datos Entidad-Relación	4	4	Escribe las relaciones del siguiente diagrama, separadas por un punto y coma (;)	3			
1	1	Teoría	Atributos	3	5	Determina si todos los atributos de la entidad CURSO son atributos. Si no es así selecciónalo(s).	3			

Nueva Pregunta

Esta sección nos permite:

- Alta. Para dar de alta una Pregunta es necesario
 - Dar clic en el botón „Nuevo Tipo de Pregunta’
 - Seleccionar el Examen al que pertenece la pregunta. Una vez que se haya hecho la selección se muestra el número de preguntas que hasta el momento conforman el examen seleccionado y la puntuación total
 - Seleccionar el capítulo al que pertenece la pregunta
 - Seleccionar el Tema al que pertenece la pregunta
 - Escribir la puntuación de la pregunta
 - Seleccionar el tipo de pregunta
 - De acuerdo al tipo seleccionado escribir la pregunta, subpreguntas y respuestas
 - Dar clic en el botón „Enviar’

Administración De Preguntas

Nº de Preguntas: 5

Puntuación: 10

Alta de Pregunta

Examen:

Capitulo:

Tema:

Puntuación:

Tipo de Pregunta:

----- Seleccione un tipo de pregunta -----

Relacionar Columnas

Opción Múltiple - 1 respuesta

Opción Múltiple - 1 ó más respuestas

Respuesta en texto

Generación de diseño (tablas)

Álgebra Relacional

- Modificación. Modifica las preguntas, subpreguntas y respuestas asociadas

- Dar clic en el icono  de la pregunta que se desea modificar
- Escribir la nueva información
- Dar clic en el botón „Enviar“

Administración De Preguntas

Modifica Pregunta

Examen:

Capítulo:

Tema:

Puntuación:

Tipo de Pregunta:

Pregunta:

Respuesta:

- Eliminación. Elimina la pregunta, subpreguntas y respuestas asociados

- Dar clic en el icono  del tipo de pregunta que se desea eliminar
- Dar clic en el botón „Aceptar“ del mensaje de confirmación de la eliminación

Preguntas Existentes

Exam.	Ver.	Capítulo	Tema	Tipo	No.	Descripción	Puntos	Imagen	Modificar	Eliminar
1	1	Teoría	Cronología de las bases de datos	1	1	Relaciona cada pregunta con su respuesta correcta.	4			
1	1	Teoría	Cronología de las bases de datos	1	2	Relaciona cada pregunta con su respuesta correcta.	2			
1	1	Teoría	Definición de una base de datos				2			
1	1	Teoría	Teoría del modelo de datos Entidad-Relación			ama,	3			
1	1	Teoría	Atributos			dad onalo(s).	3			
1	1	Teoría	Definición de una			de está en	4			

Microsoft Internet Explorer

?

¿Esta seguro de eliminar la pregunta?

5.7 Administración de Asignación de Respuestas

Administración de Asignación de Respuestas

Nº	Pregunta	Modificar Asignación	Eliminar Asignación
1	Relaciona cada pregunta con su respuesta correcta.		
2	Relaciona cada pregunta con su respuesta correcta.		
3	¿Cuándo se utiliza barra UID?		
5	Determina si todos los atributos de la entidad CURSO son atributos. Si no es así selecciónalo(s).		
6	Para el siguiente MER evaluar las reglas de normalización. Identificar si cada atributo está en el lugar correcto; si no lo está, identificar qué regla de normalización se está violando.		
7	¿Qué tipo de relaciones se utilizan para modelar jerarquías?		
29	¿Qué etapas componen el ciclo de vida de un sistema?		

- Modificación. Modifica la asignación de las respuestas para las preguntas de opción múltiple

- Dar clic en el icono  de la asignación que se desea modificar
- Seleccionar para la pregunta la respuesta nueva
- Dar clic en el botón „Enviar“

Administración de Preguntas - Respuestas

Modificación de la Asignación de Respuestas

¿Qué etapas componen el ciclo de vida de un sistema?

b) Análisis, Estrategia y Diseño

-- Seleccione la respuesta correcta --

a) Construcción, Transición y Producción

b) Análisis, Estrategia y Diseño

c) Análisis, Diseño y Construcción

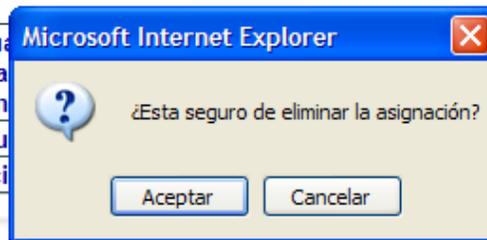
Enviar

- Eliminación. Elimina la asignación.

- Dar clic en el icono  de la asignación que se desea eliminar
- Dar clic en el botón „Aceptar” del mensaje de confirmación de la eliminación

Administración de Asignación de Respuestas

N°	Pregunta	Modificar Asignación	Eliminar Asignación
1	Relaciona cada pregunta con su respuesta correcta.		
2	Relaciona cada pregunta con su respuesta correcta.		
3	¿Cuándo se utiliza barra UID?		
5	Determina si todos los atributos de la entidad CURSO son atributos. Si no es así selecciónalo(s).		
6	Para el siguiente MER evalúa si cada atributo está en el lugar si la normalización se está violando la regla de		
7	¿Qué tipo de relaciones se u		
29	¿Qué etapas componen el ci		



5.8 Consulta de Historial Académico

Consulta de Historial Académico

Clave

Nombre

Apellido Paterno

Apellido Materno

Esta sección nos permite:

- Realizar la búsqueda del historial académico de un alumno ya sea por clave, nombre, apellido paterno o materno. Dar clic en el botón „Consultar”
- Consultar el historial académico de todos los alumnos. Dar clic en el botón „Ver Todos”

Consulta de Historial Académico

Clave: 1

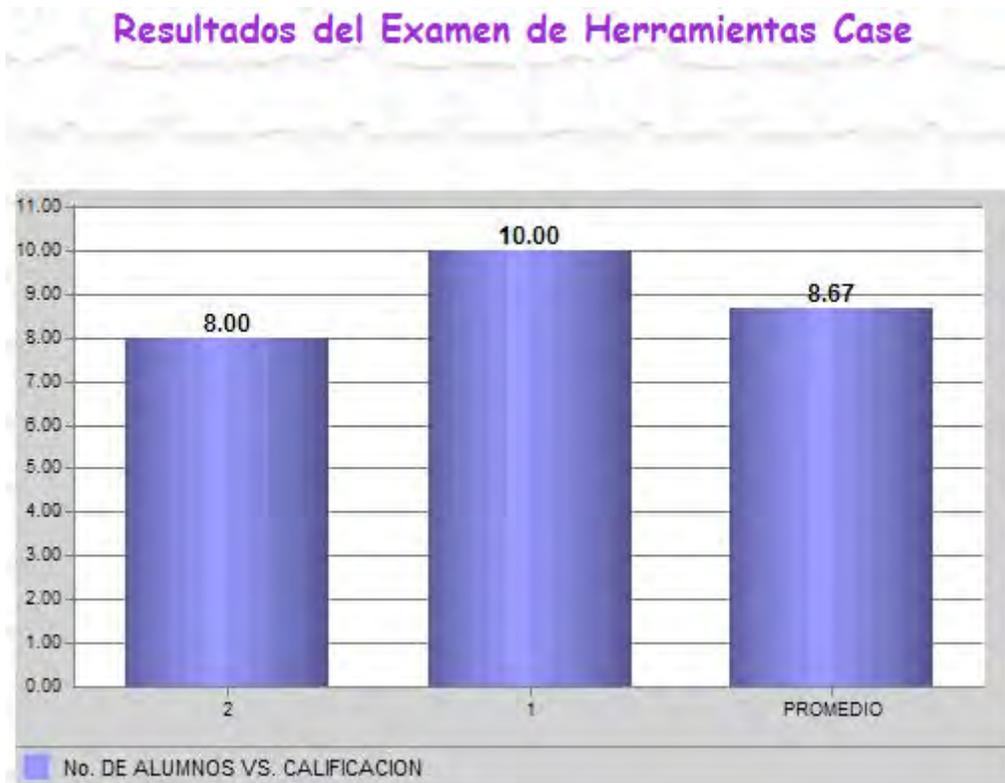
Alumno: Olga Cisneros Monroy

Examen	Versión	Descripción	Fecha	Calificación
1	1	Examen de teoría y análisis Sem 2004-2	05-Julio-2005	10
1	1	Examen de teoría y análisis Sem 2004-2	12-Julio-2005	4.55
1	1	Examen de teoría y análisis Sem 2004-2	28-Febrero-2008	0.77
4	1	Examen de construcción	10-Noviembre-2005	10
5	1	Examen de Herramientas Case	24-Noviembre-2008	8

5.9 Gráfica de Resultados de Exámenes

Esta sección nos permite visualizar la gráfica de las calificaciones obtenidas contra el número de alumnos.

- Se debe seleccionar el examen que se quiere consultar.



VI. Conclusiones

Concluyó que:

El modelo educativo virtual está plenamente soportado tanto desde su perspectiva tecnológica como didáctica, y su uso se irá incrementando. Concretamente, en el caso de la enseñanza superior en nuestro país, prácticamente todas las Universidades públicas y privadas o bien imparten cursos virtuales en la actualidad o bien han iniciado o realizado ya distintos proyectos en el área de la teleformación.

La enseñanza virtual nos brinda diferentes rubros: ejercitación, adiestramiento, simulación, evaluación y tutorías.

El acceso universal a la educación puede estar al alcance de más personas sin restricciones de género, condición social, religión, política, etc.

La enseñanza virtual permite una educación continua; ya que los tiempos en que vivimos los factores: tiempo y distancia son obstáculos para la preparación constante.

La enseñanza virtual es multicultural

La enseñanza virtual puede contribuir a la alfabetización.

El entorno virtual por su carácter multimedia abre nuevas posibilidades para el desarrollo de procesos perceptivos y sensoriales.

Las principales ventajas de la educación virtual son:

- El alumno puede adaptar el estudio a su horario personal
- El alumno tiene un papel activo que no limita recibir información sino que forma parte de su propia formación.
- Desarrolla la creatividad del estudiante
- El alumno puede medir su avance a través de sus auto - evaluaciones
- El profesor puede apoyarse en esta herramienta como complemento a su docencia.

Apéndice I. Ejemplo código de página ASP

```

<form name="frmAlumnos" method="post">
<%If request("send")= "1" Then
    strSQL = "usp_AdmonAlumnos 2, ", "&trim(request("txtNombre"))&"', "&trim(request("txtApPat"))&"',
"&trim(request("txtApMat"))&""
    rsNom.Open strSQL, objConn, 3, 3

    If Not rsNom.EOF Then 'Verifica si ya existe el nombre de alumno
        Mensaje = "Ya existe el alumno"
    Elseif request("Opc") = "1" Then 'Inserta un nuevo alumno
        strSQL = "usp_AdmonAlumnos 3, ", "&trim(request("txtNombre"))&"',
"&trim(request("txtApPat"))&"', "&trim(request("txtApMat"))&""
        rs.Open strSQL, objConn, 3, 3
        If Not rs Is Nothing Then Mensaje = "El Alumno ha sido dado de alta.<br><br><br>Usuario:
"&rs("username")&"<br><br>Contraseña:" &rs("password") Else Mensaje = "Hubo un error"
        Elseif request("Opc") = 2 Then 'Modifica Alumno
            strSQL = "usp_AdmonAlumnos 5, "&request("Clave")&"', "&trim(request("txtNombre"))&"',
"&trim(request("txtApPat"))&"', "&trim(request("txtApMat"))&""
            rs.Open strSQL, objConn, 3, 3
            If Not rs Is Nothing Then Mensaje = "El Alumno ha sido modificado" Else Mensaje = "Hubo un
error"
        Elseif request("Opc") = 3 Then 'Elimina Alumno
            strSQL = "usp_AdmonAlumnos 6, "&request("Clave")&"', ", ", ", ""
            rs.Open strSQL, objConn, 3, 3
            If Not rs Is Nothing Then Mensaje = "El Alumno ha sido eliminado" Else Mensaje = "Hubo un
error"
        End If%>
<br><center><span class="TituloPurple">Administración De Alumnos</span></center>
<br><br><br>
<table align="center" width="40%" border="0" class="BorderTabla">
    <tr class="FondoTabla">
        <td align="center" width="100%">
class="TextoTabla"><br><%=Mensaje%><br><br></td>
        </tr>
    </table>

    <br><br><br><center><input type="button" value="Regresar" id="Regresar" name="Regresar"
border="0" class="TextolInput" onClick="javascript:window.history.back()"></center>

<%
Response.End
Elseif request("Opc") = "" Then 'Despliega los alumnos existentes%>
<center><span class="TituloPurple">Alumnos Existentes</span></center><br>
<%If Not rsAlum.EOF Then%>
    <table width="50%" border="0" cellspacing="1" cellpadding="1" align="center"
class="BorderTabla">
        <tr class="FondoTituloTabla">
            <td class="TituloTabla">Clave</td>
            <td class="TituloTabla">Nombre</td>
            <td class="TituloTabla">Apellido Paterno</td>
            <td class="TituloTabla">Apellido Materno</td>
            <td class="TituloTabla">Usuario</td>
            <td class="TituloTabla" width="10%">Modificar</td>
            <td class="TituloTabla" width="10%">Eliminar</td>
        </tr>
        <%Do Until rsAlum.EOF%>
        <tr class="FondoTabla">

```

Apéndice II. Ejemplo de código store procedure

```
CREATE PROCEDURE usp_AdmonAlumnos
@opc as int,
@Clave as int,
@Nombre as varchar(25),
@ApPat as varchar(25),
@ApMat as varchar(25)

AS

DECLARE    @Numero as int,
           @Username as varchar(30),
           @UsernameOld as varchar(30),
           @Pasword as varchar(8),
           @Consecutivo as int,
           @IniNum as int,
           @FinNum      as int,
           @IniLetas int,
           @FinLet as int,
           @strQuery as varchar(500)

SET @IniNum = 0
SET @FinNum = 9
SET @IniLet = 97 --a
SET @FinLet = 122 --z

If @opc = 1 --Trae los Alumnos dados de alta
Begin
    SELECT A.Clave, A.Nombre, A.ApPat, A.ApMat, U.Username, U.Pasword, U.Perfil
    FROM tblAlumnos A, tblUsuarios U
    WHERE A.Activo = 1 AND A.Clave = U.ClaveAlumno
    ORDER BY A.Clave
End

If @opc = 2 --Verifica si existe tblAlumnos para traer su clave
Begin
    SELECT Clave FROM tblAlumnos WHERE Nombre = @Nombre AND ApPat = @ApPat AND ApMat =
@ApMat
End

If @opc = 3 --Inserta Alumno
Begin
    --Traigo la siguiente clave de alumno
    SELECT @Numero =
    CASE
        WHEN MAX(Clave) IS NULL THEN 1
        ELSE MAX(Clave)+1
    END
    FROM tblAlumnos

    BEGIN TRAN
        --SE INSERTA ALUMNO
        INSERT INTO tblAlumnos (Clave, Nombre, ApPat, ApMat, Activo) VALUES (@Numero,
@Nombre, @ApPat, @ApMat, 1)
```

Bibliografía

- * Introducción a los Sistemas de Bases de Datos
C.J.DATE
Addison-Wesley Iberoamericana
- * Organización de las Bases de Datos
JAMES MARTIN
Prentice Hall
- * Fundamentos de Bases de Datos
FORTH / SILBERSCHARTZ
Mc. Graw Hill
- * Understanding Database Management Systems
JOSEPH A. VASTA
Wadsworth Publishing Company
Grupo Editorial Iberoamericana
- * Design Analysis and Performance Evaluation
Methodologies for D.B. Computers.
Steven A. Demurjiad, David K. Hsiao, Roger G. Marshall
Prentice Hall
- * Introducción a las Bases de Datos
Mark L. Gillenson
Mc. Graw Hill
- * Database Analysis and Design
Y. T. Hawryszkiewicz
Maxwell Macmillan
- * Técnicas de Diseño de Bases de Datos
Secundala Atre, G. Gordorin
Parainfo
- * ORACLE Manual de Referencia
George Koch
Osborne Mc. Graw Hill
- * Database Management and Design (Diseño y Administración de B.D.)
Gary W. Hasen y James V.
Prentice Hall
- * Mastering Oracle 7 & Client/Server Computing
Steven M. Bobrowski
SYBEX
- * ORACLE, Guía de aprendizaje. Una completa introducción para principiantes
Michael Abbel, J. Corey
Oracle Press - Mc Graw Hill
- * Principles of Database Systems
Ulman, Jeffrey
Computer Science Press
- * Data Processing Fundamentals, Design, Implementation
David M. Kroenke
Mc Millan
- * ORACLE DBA Handbook
Kevin Loney
Every DBA can learn to manage a networked oracle database efficiently and effectively with this comprehensive guide.
Price \$34.95
ISBN 0-07-881182-1
- * UNIX for Oracle DBAs Pocket Reference
Donald K. Buleson
O'Reilly

- * TUNING ORACLE
Michael Abbey, Daniel J. Dechichlo
Learn to customize oracle for optimal performance and productivity with this focused guide
Price 29,95
- * Oracle8 & UNIX
Performance Ahmed / Alomari
- * An introduction to Database Systems, Vol I
Date, C.J.
5ª. Edición, Addison-Wesley
- * An Introduction to Database Systems, Vol II
Date, C. J.
Addison-Wesley
- * Relational Database; Selected Writings
Date, C.J.
Addison-Wesley
- * A Relational Model for Large Shared Data Banks
Codd, E.F.
Communications of the ACM, Vol 13 Number 6, Junio 1970
- * Recommended Diagramming Standards for Analyst & Programmers
Martin, J
Prentice-Hall, Englewood Cliffs
- * Structured Techniques for Computing
Martin, J., an McClure, C.
Prentice-Hall, Englewood Cliffs
- * Entity Modeling; Techniques and Application
Ross, R.G.
Database Research Group
- * Modern Structured Analysis
Yourdon, E.
Yourdon Press Computing Series
- * Handbook of Relational Database Design
Flemming, C.C. and von Halle, B.
Addison-Wesley, Reading
- * A simple Guide to five Normal Forms in Relational Database Theory
Kent, W
Computing Practices, Volume 26, Number 2, Feb 1983
- * Computer aided software Engineering - the methodologies, the Products, and the future
Prentice Hall
- * CASE Tools : A comparison of IEW and IEF. InfoDB, Vol 3 Number 2
Winsberg, P.
- * CASE : Getting the Big Picture Database Programming and Design, Vol 1, Number 3 , March 1988
Winsberg, P.
- * AD/Cycle Overview and Analysis
Winsberg, P.
InfoDB vol 4 number 4, Winter 1989/1990
- * Open and distance learning and the use of new technology for the self development of manager
Binstead,D.
Centre for the Study of Management Learning, University of Lancaster
- * Interactive Learning International Vol.5
Topham,P.

Referencias

- * <http://www.monografias.com/trabajos24/educacion-virtual/educacion-virtual.shtml>
- * <http://www.educar.org/articulos/educacionvirtual.asp>
- * <http://contexto-educativo.com.ar/2001/5/nota-10.htm>