

UNIVERSIDAD MAYOR DE SAN ANDRES  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMATICA



## **TESIS DE GRADO**

“METODOLOGIA PARA LA AUTOMATIZACIÓN DE  
BASE DE DATOS”

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMATICA  
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS

POSTULANTE: OVIDIO SANGA CONTRERAS  
TUTOR: LIC. EFRAIN SILVA SANCHES  
REVISOR: Mg. Sc. CARLOS MULLISACA CHOQUE

LA PAZ - BOLIVIA

2011

## **DEDICATORIA**

A mis queridos padres Mario y Ines quienes  
me apoyaron, me brindaron su cariño y me  
dieron aliento para seguir adelante.

A mis hermanos por brindarme su apoyo en todo

A una persona bien especial Zuli y a la luz de mi  
Esperanza Mey Ling.

## **AGRADECIMIENTOS**

El esfuerzo y la realización de este trabajo estuvieron centrados en varias personas a las que tengo que nombrar:

- Lic. Efraín Silva, por la ayuda brindada,
  - Lic. Carlos Mullisaca por colaborarme en mi trabajo,
  - Ing. Jorge Tancara mi amigo y mentor,
  - Ronald, al que siempre lo tuve en las buenas y en las malas,
  - Teddy, por siempre espíritu de ayuda incondicionalidad,
  - Pedro, por su impulso, a veces necesario, a no claudicar ante las adversidades,
  - Erick, por ayudarme y por su amistad incondicional,
  - Zulema, por su apoyo incondicional de ayer, hoy y siempre,
  - Mey Ling, por darme una luz en la vida
- y muy en especial a mi madre de cariño Ines, que solo gracias a ese su incansable e incomprensible amor en las adversidades pasadas, pude entender el amor de madre.

A todos ellos... MUCHAS GRACIAS

## RESUMEN

El modelado de Bases de Datos, es fundamental para el almacenamiento óptimo de la información que manipularan las aplicaciones denominadas: sistemas de información

Si bien las técnicas mencionadas anteriormente son frecuentemente empleadas, su proceso presenta una serie de dificultades fundamentalmente para diseñadores novatos, es por esa razón que se propone la siguiente metodología denominada: EAS-J, la cual es altamente versátil e *ideal para ser automatizada*

El grado de normalización depende mucho del diseñador, esto implica “controlar la redundancia”, aunque lo ideal sería tener una base de datos que inclusive se encuentre en “quinta forma normal”, obviamente con las posibles “dificultades” en la generación de reportes y la lentitud de las mismas.

Diseño conceptual.- es la construcción de un modelo que represente todos los datos utilizados en una organización independientemente de las consideraciones físicas

Diseño lógico.- construir un modelo de la organización basados en un modelo de datos específicos, relacionar el modelo conceptual con el lógico

Diseño físico.- generar de cómo va ha ser la implementación de la base de datos dependiendo de la el SGBD que se baya ha utilizar también vemos si hay redundancia en un modelo de datos

Primero se debe examinar y reexaminar las relaciones uno a uno en un diseño de base de datos

El proceso debe ser efectuado de manera recursiva a todos los atributos de las tablas que se vayan generando.

## ABSTRACT

Modeling Database is essential for optimal storage of the información referred to manipulate applications: information system

While the above techniques are frequently employed, the process presents a number of difficulties mainly for novice designers, it is for this reason that we propose the following methodology called EAS-J, which is highly versatile and ideal for automated

The degree of normalization depends very much on the designer, this means "controlled redundancy", although the ideal would be an inclusive database is in "Fifth normal form", obviously with the possible "difficulties" in the reporting and the slowness of the same.

Conceptual design is the construction of a model that represents all the data used in an organization regardless of physical considerations

Logical Design .- build a model of organization based on a specific data model linking the conceptual model to the logical

Physical design .- generation is going to be how the implementation of the database depending on the DBMS that Berry has also been used see if there is redundancy in a data model

First you must examine and reexamine the one-to-one in a database design

The process must be done in a recursive manner to all attributes from tables that are generated.

# ÍNDICE DE CONTENIDOS

## CAPITULO I

1.1	.....INTRODUCCIÓN.....	1
1.2	..... ANTECEDENTES .....	2
1.3	.....PLANTEAMIENTO DEL PROBLEMA .....	2
1.3.1	PROBLEMA GENERAL .....	3
1.4	.....HIPÓTESIS .....	4
1.5	.....OBJETIVOS .....	4
1.5.1	.....OBJETIVO GENERAL.....	4
1.5.2	.....OBJETIVO ESPECIFICO.....	4
1.6	.....JUSTIFICACIÓN .....	5
1.6.1	.....SOCIAL .....	5
1.6.2	.....TÉCNICO .....	5
1.6.3	.....ECONÓMICA .....	6
1.6.4	.....METODOLOGÍA .....	6
1.7	.....ALCANCES Y LIMITES .....	7
1.8	..... APORTES .....	7

## CAPITULO II

### MARCO TEORICO

2.1	.....INTRODUCCION.....	9
2.2	..... EVOLUCIÓN E HISTÓRICA DE BASE DE DATOS	9
2.2.1	.....Década de 1950.....	10
2.2.2	.....Década de 1980 .....	10
2.2.3	.....Principios década de los 90 .....	11
2.2.4	.....Finales de la década de los 90 .....	11
2.2.5	.....Siglo XXI .....	12
2.2.6	.....DEFINICIÓN DE BASE DE DATOS .....	12

2.3	CARACTERÍSTICA	12
2.4	TEORIA DE APRENDIZAJE Y ENSEÑANZA	13
2.4.1	LA TEORIA COGNOSCITIVA	13
2.4.2	LA TEORIA CONTRUCTIVA	13
2.4.3	TECNICAS EDUCATIVAS APLICABLES	19
2.4.4	QUE ES LA ENSEÑANZA	22
2.5	EL METODO	23
2.5.1	CLASES DE METODOS	23
2.6	SISTEMA DE GESTIÓN DE B.D. (SGBD)	23
2.7	CONTROL SOBRE LA REDUNDANCIA DE DATOS	24
2.7.1	CONSISTENCIA DE DATOS	24
2.7.2	COMPARTICIÓN DE DATOS	24
2.7.3	MANTENIMIENTO DE ESTÁNDARES	25
2.7.4	MEJORA EN LA INTEGRIDAD DE DATOS	25
2.7.5	MEJORA EN LA SEGURIDAD	25
2.7.6	MEJORA EN LA ACCESIBILIDAD A LOS DATOS	26
2.7.7	MEJORA EN LA PRODUCTIVIDAD	26
2.7.8	MEJORA EN EL MANTENIMIENTO	26
2.7.9	AUMENTO DE LA CONCURRENCIA	26
2.7.10	MEJORA EN LOS SERVICIOS DE COPIAS DE SEGURIDAD	27
2.7.11	DESVENTAJAS DE LAS BASES DE DATOS	27
2.7.12	TIPOS DE CAMPOS	27
2.8	TIPOS DE BASE DE DATOS	29
2.8	Bases de datos estáticas	29
2.8.2	Bases de datos dinámicas	29
2.8.3	Bases de datos bibliográficas	30
2.8.4	Bases de datos de texto completo	30
2.8.5	Bases de datos o "bibliotecas" de información Química Biológica	30
2.8.6	Modelo de Bases de datos	30

2.8.7 Bases de Datos jerárquicas.....	30
2.8.8 Bases de Datos de Red.....	32
2.8.9 Bases de Datos Transaccionales.....	32
2.8.10 Bases de Datos relacionales .....	32
2.8.11 Bases de Datos multidimensionales .....	33
2.8.12 Bases de Datos orientadas a objetos .....	34
2.8.13 Bases de Datos documentales .....	35
2.8.14 Bases de Datos deductivas .....	35
2.9 Álgebra Relacional .....	36
2.9.2 Conversión de Diagramas E/R a tablas. ....	39
2.9.3 Teoría de la normalización. ....	41
2.9.4 Dependencias funcionales. ....	42
2.9.5 Axiomas De Armstrong. ....	42
2.9.6 Clausuras. ....	45
2.9.7 Algoritmo para calcular x .....	46

## **CAPITULO III**

### **MARCO APLICATIVO**

3.1 .....DESCRIPCION DEL METODO.....	52
3.2 .....HERRAMIENTAS DE ESPECIFICACION .....	52
3.2.1 .....METODOLOGÍA DEL DESARROLLO .....	53
3.3 .....ARQUITECTURAS DE SISTEMAS DE B.D .....	53
3.4 .....EL ESTÁNDAR ANSI/SPARC .....	55
3.5 .....CORRESPONDENCIAS, ASOCIACIONES O LIGADURAS (MAPPINGS) .....	55
3.6 .....MODELO DE BASES DE DATOS .....	56
3.7 .....ARQUITECTURA CENTRALIZADA.....	57
3.8 APLICACIÓN DE LA METODOLOGIA .....	58
3.8 .....METODOLOGÍA EAS-J .....	58
3.8.1 .....APLICACIÓN DE LA METODOLOGIA .....	58



3.8.2 .....	CONCEPTOS FUNDAMENTALES.....	59
3.8.2.1	ENTIDAD.....	59
3.8.2.2	ENTIDADES OBJETO .....	60
3.8.2.3	ENTIDADES SUJETO.....	60
3.8.2.4	ENTIDADES EVENTO .....	60
3.8.3 .....	RELACIÒN RECURSIVA.....	62
3.8.4 .....	TIPOS DE ATRIBUTOS.....	62
3.8.4.1 .....	Atributo Llave Primaria .....	62
3.8.4.2 .....	Atributo Llave Foránea.....	62
3.8.4.3 .....	Atributo Multivalorado .....	63
3.8.4.4 .....	Atributo Candidato A Patron De Búsqueda .....	63
3.8.4.5 .....	Atributo Descriptor Puro.....	63
3.8.5 .....	Semántica De Atributos .....	63
3.8.5.1 .....	Atributos Estáticos.....	63
3.8.5.2 .....	Atributos Desencadenadores.....	63
3.8.5.3 .....	Semántica De Un Atributo Multivalorado .....	63
3.8.5.4 .....	Semántica De Un Atributo Candidato A Patrón De Búsqueda .....	64
3.8.6	ETAPAS DE LA METODOLOGÍA .....	64

## **CAPITULO IV**

### **MARCO FINAL**

4.1 .....	CONCLUSIONES Y RECOMENDACIONES .....	71
	BIBLIOGRAFIA .....	72

## **ANEXOS**

## Presentación

### 1.9 Introducción

El incesante crecimiento de la tecnología nos hace pensar e imaginar una forma de aprendizaje más efectiva, puesto que la utilización de herramientas informáticas se hace necesaria para dirigir las al campo de la educación

El modelado de Bases de Datos, es fundamental para el almacenamiento “optimo” de la información que manipularan las aplicaciones denominadas: sistemas de información.

La motivación de este trabajo radica en la dificultad que tienen los estudiantes que cursan la asignatura de base de datos de la carrera de informática de la Universidad Mayor de San Andrés.

Con el presente trabajo, se pretende aportar con esta nueva herramienta de aprendizaje, en este caso especial para aprender una nueva metodología (EAS-J) inspirado en el modelado ER (Entidad Relación).

El mundo competitivo en la actualidad, ha definido un nuevo escenario en los métodos de estudio obligando en muchos casos, al establecimiento de vínculos entre distintas tecnologías donde el intercambio de información resulta de vital importancia, el uso de estas tecnologías es una aproximación de solución a la mejor comprensión del tema de interés, Para la elaboración de esta nueva metodología EAS-J(Entidad Atributo Semántica-J) se hace uso de esta

herramienta que aporta la informática y se verán en estrecha relación con otras metodologías en el área de la informática.

### 1.10 Antecedentes

Las nuevas herramientas para el aprendizaje en la educación, para el modelado de una Base de Datos, es fundamental para el almacenamiento “optimo” de la información que manipularan las aplicaciones denominadas: sistemas de información.

El uso de la computadora se ha tenido muy en cuenta, y una ascendente aplicación ya que en el principio solo se utilizaba como una herramienta de cálculo y editor de textos, como procesador de palabras, archivos programas que hasta el día de hoy combinan texto, sonido e imagen gracias a tecnologías multimedia.

La facilidad que brindan estas tecnologías permite desarrollar diversas aplicaciones, es así que en este campo fascinante de la educación se ha desarrollado sistemas como ser sistemas de instrucción basada en computadoras (Computer based instruction CBI).

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una **base de datos** se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada.

El modelado de base de datos para la metodología EAS-J, donde el grado de normalización depende mucho del diseñador, esto implica “controlar la redundancia” aunque lo ideal sería tener una base de datos que inclusive se encuentre en “quinta forma normal”, obviamente con las posibles “dificultades” en la generación de reportes y la lentitud de las mismas.

En los libros de Alvin Toffler<sup>1</sup>, la historia de la humanidad se entiende como dividida en tres periodos a las que denomino Olas. La primera ola corresponde a la agricultura, la segunda ola se refiere a la sociedad industrial o revolución industrial, y la tercera ola corresponde al desarrollo de las tecnologías de la

Información TIC. Y no en vano Peter Ducker nos dice que la nueva era es la del conocimiento.

## 1.11 Planteamiento Del Problema

### 1.11.1 Problema General

El principal problema que se pretende resolver con la metodología EAS-J es el siguiente:

Disminuir la redundancia y evitar la inconsistencia. Son objetivos básicos de una base de datos eficiente. Disminuir la redundancia consiste en agrupar todos los datos en un mismo objeto sin repetir información. Esto no puede realizarse siempre, con lo cual hay ocasiones en las que se duplica información.

### 1.3.2 Problema Específico

- Si bien las técnicas mencionadas anteriormente son frecuentemente empleadas, su proceso presenta una serie de dificultades fundamentalmente para los diseñadores es por esa razón que se propone la siguiente metodología denominada: EAS-J, la cual es altamente versátil e *ideal para ser automatizada*.
- Normalizar una base de datos y llegar a la quinta forma normal
- Evitar la redundancia
- Inconsistencia

***¿Será posible mejorar el almacenamiento óptimo de una Base de Datos y así evitar la redundancia e inconsistencia, implementando una nueva metodología para su óptimo almacenamiento de la información?***

## 1.12 Hipótesis

**“Aplicando la metodología EAS-J se construye un método óptimo para la automatización de una base de datos automáticamente”.**

## **1.13 Objetivos**

### **1.13.1 Objetivo General**

Desarrollar una metodología capaz de adaptarse a las necesidades, para automatizar una Base de Datos.

### **1.13.2 Objetivo Especifico**

- Desarrollar una estrategia teórica, el cual ayudara de sustento y marco de referencia.
- Hacer el uso de la metodología para el diseño, que posibiliten una solución al problema de acuerdo a las necesidades del área.
- Construcción del método
- Prueba del método
- Construcción del prototipo para la automatización de una base de Datos
- Realizar pruebas con datos reales y así obtener los resultados requeridos por el sistema de automatización de B.D.

## **1.14 Justificación**

Esta metodología permitirá esclarecer algunos interrogantes respecto de cómo enseñar esta nueva metodología y aplicarse para la materia de Base de Datos y para y para todos los estudiantes de la carrera de informática, El modelado de base de datos, es fundamental para el almacenamiento optimo de la información y que no haya redundancia de datos, para las aplicaciones de nominadas “sistemas de información”.

### **1.14.1 Social**

Mediante el crecimiento de la tecnología, se ha definido un nuevo escenario en los medios de educación, obligando, en muchos casos, al establecimiento de vínculos de distintos métodos de enseñanza. Se ha observado que en los cursos universitarios de los sextos semestres de la carrera de informática sea aplicado ya

como una de las metodologías de aprendizaje, donde la relación estudiante/docente es muy alta y existe una gran diferencia en el nivel de conocimientos previos que traen los alumnos se piensa que esta metodología

#### **1.14.2 Técnico**

La informática va revolucionando el mundo, toda la información que necesitamos ya está siendo procesada cada vez más en sistemas basados en conocimientos

El desarrollo de esta metodología para la implementación permitirá apoyar mejor la enseñanza en el área de manejo de modelado de Base de Datos. Siendo el principal apoyo del computador, es evitar la redundancia y la inconsistencia de la información, de acuerdo a normas curriculares y adecuarse a los estudiantes y están destinados a utilizar esta nueva metodología para el modelado de una base de datos.

#### **1.14.3 Económica**

Se pretende alcanzar los objetivos dando a elegir una nueva metodología para el modelado de Base de Datos para los estudiantes, tratando de no invertir gastos económicos, cabe señalar que esta nueva metodología sea aplicable para la materia de Base de Datos.

#### **1.14.4 Metodología**

El desarrollo del presente trabajo de investigación se lo llevo adelante en orden de las siguientes actividades:

- a) En la etapa inicial se centra en la revisión de libros y también lo avanzado en la materia de Base de Datos, también en los estudios de los docentes de la carrera de informática; relevaran las posibles opciones en elegir esta nueva metodología para el modelado de una base de datos. En esta etapa finaliza con la elaboración de un informe que constituye el marco teórico, donde se resumen las investigaciones más relevantes en el tema y el estado actual de las herramientas, métodos y técnicas disponibles aquí se describen también



los fundamentos teóricos que cimentaran la solución elegida a través de la investigación de libros, revistas especializadas, sitios de internet de centros de investigación de universidades nacionales e internacionales y eventos de la temática (Simposios, congresos, etc.).

- b) Consiste en el diseño y elección de esta nueva metodología para su aplicación en el aprendizaje, esto determinara la validez de esta metodología utilizada.
- c) Se definirá esta metodología para la Base de Datos es fundamental para el almacenamiento “optimo” de la información que manipulan las aplicaciones denominadas: sistemas de información.

### **Para la elaboración del modelo del prototipo se hará uso**

Para utilizar esta nueva metodología EAS-J será fundamental el modelo es:

- Evitar la redundancia
- Inconsistencia

### **Los recursos informáticos a utilizar para la implementación serán**

- Un ordenador
- Lenguaje de programación

### **1.15 Límites Y Alcances**

La presente investigación se orienta a una nueva metodología EAS-J. Para el modelado de Base de Datos, más allá de las condiciones que se plantea en la presente tesis cabe hacer notar de sus alcances y limitaciones.

Con esta metodología permitirá dar respuesta de forma sistemática y ordenada a los distintos requerimientos que tengan los estudiantes que deseen aplicar esta nueva metodología para el diseño de base de datos

Por otra parte el presente trabajo, no debe ser considerado como lo último sino más bien una herramienta orientadora limitada por los objetivos que se plantean. consiguientemente se hace una apertura para futuras ampliaciones y así cubrir todo el ámbito informático.

### **1.16 Aportes**

Esta nueva metodología será de gran manera un aporte muy importante para la materia de Base de Datos, ya que contribuirá a mejorar el proceso de aprendizaje de la educación, Donde se utiliza herramientas, metodologías, etc. que proporciona la informática para elaborar esta nueva metodología EAS-J, que ira directo y beneficio a la sociedad y los informáticos que hará el trabajo más fácil.

Esta metodología para su enseñanza proporciona soluciones en un dominio específico del área, mediante la aplicación de esta metodología EAS-J para una Base de Datos. Lo principal es de apoyar a los docentes y ayudantes de la materia





## 2. Introducción

El trabajo de investigación que se encara, necesita de fuentes teóricas y fuentes conceptuales necesarias para el desarrollo del mismo, y de acuerdo a la hipótesis y los objetivos planteados permitirán la generación de una herramienta para el ejercicio del profesional informático.

Es así que en el presente acápite se presentan las bases teóricas y metodológicas que nos permitirán el logro de los objetivos.

El manejo de datos con gestores de base de datos - los sistemas, software y herramientas tecnológicas necesarias para manejar para todo tipo de datos en la aplicación de tecnologías computarizadas para el manejo de datos

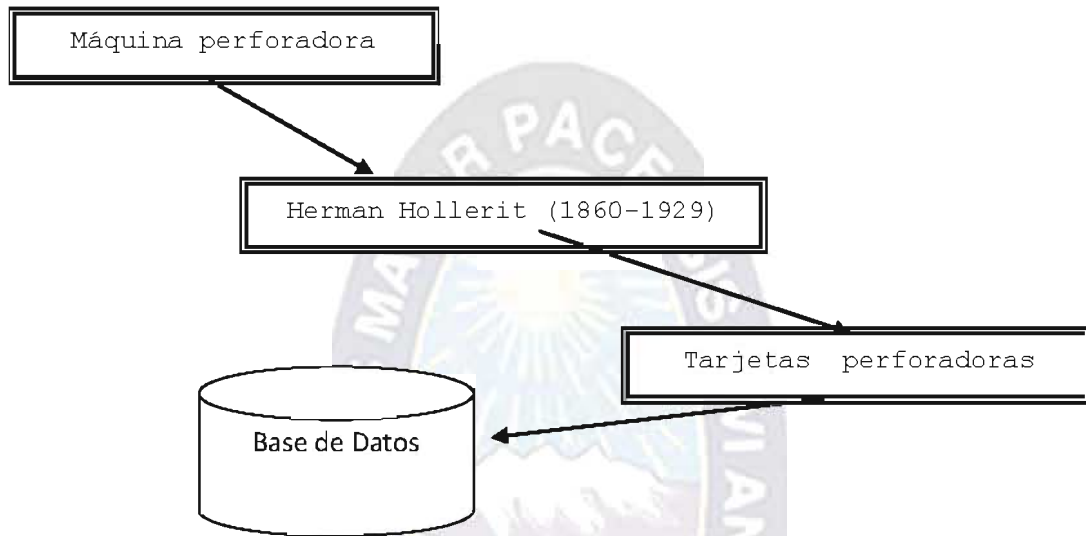
### 2.2 Evolución E Histórica De Base De Datos

El uso de sistemas de bases de datos automatizadas, se desarrollo a partir de la necesidad de almacenar grandes cantidades de datos, para su posterior consulta producidas por las nuevas industrias que creaban gran cantidad de información.

Herman Hollerit (1860-1929) fue denominado el primer ingeniero estadístico de la historia, ya que invento una computadora llamada "Máquina Automática Perforadora de Tarjetas". Para hacer el censo de Estados Unidos en 1880 se tardaron 7 años para obtener resultados, pero Herman Hollerit en 1884 creo la máquina perforadora, con la cual, en el censo de 1890 dio resultados en 2 años y medio, donde se podía obtener datos importantes como número de nacimientos,

población infantil y número de familias. La máquina uso sistemas mecánicos para procesar la información de las tarjetas y para tabular los resultados.

**Figura 2.1 Evolución cronológica de la base de datos**



**Fuente [McCall, 1960]**

A diferencia con la máquina de Babbage, que utilizaba unas tarjetas similares, estas se centraban en dar instrucciones a la máquina. En el invento de Herman Hollerit, cada perforación en las tarjetas representaba un número y cada dos perforaciones una letra, cada tarjeta tenía capacidad para 80 variables. La máquina estaba compuesta por una perforadora automática y una lectora, la cual por medio de un sistema eléctrico leía los orificios de las tarjetas, esta tenía unas agujas que buscaban los orificios y al tocar el plano inferior de mercurio enviaba por medio del contacto eléctrico los datos a la unidad.

Este invento disparó el desarrollo de la tecnología, la industria de los computadores, abriendo así nuevas perspectivas y posibilidades hacia el futuro.

### **1) Década de 1950**

En este lapso de tiempo se da origen a las cintas magnéticas, las cuales sirvieron para suplir las necesidades de información de las nuevas industrias. Por medio de

este mecanismo se empezó a automatizar la información de las nóminas, como por ejemplo el aumento de salario. Consistía en leer una cinta o más y pasar los datos a otra, y también se podían pasar desde las tarjetas perforadas. Simulando un sistema de Backup, que consiste en hacer una copia de seguridad o copia de respaldo, para guardar en un medio extraíble la información importante. La nueva cinta a la que se transfiere la información pasa a ser una cinta maestra. Estas cintas solo se podían leer secuencial y ordenadamente.

## **2) Década de 1960**

El uso de los discos en ese momento fue un adelanto muy efectivo, ya que por medio de este soporte se podía consultar la información directamente, esto ayudo a ahorrar tiempo. No era necesario saber exactamente donde estaban los datos en los discos, ya que en milisegundos era recuperable la información. A diferencia de las cintas magnéticas, ya no era necesaria la secuencialidad, y este tipo de soporte empieza a ser ambiguo.

Los discos dieron inicio a las Bases de Datos, de red y jerárquicas, pues los programadores con su habilidad de manipulación de estructuras junto con las ventajas de los discos era posible guardar estructuras de datos como listas y árboles.

## **3) Década de 1970**

Edgar Frank Codd (23 de agosto de 1923 – 18 de abril de 2003), en un artículo "Un modelo relacional de datos para grandes bancos de datos compartidos" ("A Relational Model of Data for Large Shared Data Banks") en 1970, definió el modelo relacional y publicó una serie de reglas para la evaluación de administradores de sistemas de datos relacionales y así nacieron las bases de datos relacionales.

A partir de los aportes de Codd el multimillonario Larry Ellison desarrollo la base de datos Oracle, el cual es un sistema de administración de base de datos, que se destaca por sus transacciones, estabilidad, escalabilidad y multiplataforma.

Inicialmente no se uso el modelo relacional debido a que tenía inconvenientes por el rendimiento, ya que no podían ser competitivas con las bases de datos jerárquicas y de red. Ésta tendencia cambio por un proyecto de IBM el cual

desarrolló técnicas para la construcción de un sistema de bases de datos relacionales eficientes, llamado System R.

#### 4) **Década de 1980**

Las bases de datos relacionales con su sistema de tablas, filas y columnas, pudieron competir con las bases de datos jerárquicas y de red, ya que su nivel de programación era bajo y su uso muy sencillo.

En esta década el modelo relacional ha conseguido posicionarse del mercado de las bases de datos. Y también en este tiempo se iniciaron grandes investigaciones paralelas y distribuidas, como las bases de datos orientadas a objetos.

#### 5) **Principios década de los 90**

Para la toma de decisiones se crea el lenguaje SQL, que es un lenguaje programado para consultas. El programa de alto nivel SQL es un lenguaje de consulta estructurado que analiza grandes cantidades de información el cual permite especificar diversos tipos de operaciones frente a la misma información, a diferencia de las bases de datos de los 80 que eran diseñadas para las aplicaciones de procesamiento de transacciones. Los grandes distribuidores de bases de datos incursionaron con la venta de bases de datos orientada a objetos.

#### 6) **Finales de la década de los 90**

El boom de esta década fue la aparición de la WWW "Word Wide Web", ya que por éste medio se facilitaba la consulta de las bases de datos. Actualmente tienen una amplia capacidad de almacenamiento de información, también una de las ventajas es el servicio de siete días a la semana las veinticuatro horas del día, sin interrupciones a menos que haya planificaciones de mantenimiento de las plataformas o el software.

#### 7) **Siglo XXI**

En la actualidad existe gran cantidad de alternativas en línea que permiten hacer búsquedas orientadas a necesidades específicas de los usuarios, una de las tendencias más amplias son las bases de datos que cumplan con el protocolo Open Archives Initiative – Protocol for Metadata Harvesting (OAI-PMH) los cuales permiten el almacenamiento de gran cantidad de artículos que permiten una mayor visibilidad y acceso en el ámbito científico y general.

Cabe destacar que Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows que soporta varios lenguajes de programación tales como Visual C++, Visual#, Visual J#, ASP.NET y Visual Basic.NET, aunque se están desarrollando las extensiones necesarias para otros, cuyo objetivo es permitir crear aplicaciones, sitios y aplicaciones web, así como servicios web a cualquier entorno que soporte la plataforma .Net, creando así aplicaciones que intercomuniquen entre estaciones de trabajo, páginas web y dispositivos móviles.

### 2.3 Definición De Base De Datos

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una **base de datos** se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos. Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más **columnas** y **filas**. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

#### 2.3.2 Características

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.



- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

## 2.4. Teorías de Aprendizaje-Enseñanza

Desde el punto de vista de la psicología existen tres enfoques que hacen a la teoría del aprendizaje.

**2.4.1. La teoría conductiva:** Según Torrealva [Torrealva, J., 04]. "Se concentra en el estudio de conductas que se pueden observar y medir". Ve a la mente como una "caja negra" en el sentido de que las respuestas a estímulos se pueden observar cuantitativamente ignorando totalmente la posibilidad de todo proceso que pueda darse en el interior de la mente. Indica que el proceso de aprendizaje es pasivo, en el sentido de que el alumno no lo construye, la figura del profesor tiene mucha relevancia y aquí es quien define lo que debe aprenderse y como aprenderlo.

**2.4.2. La teoría cognoscitiva:** Según Mergel [Mergel, E., 05]. La teoría cognoscitiva reconoce la importancia del reforzamiento, pero enfatiza su papel como elemento retro alimentador para la corrección de respuestas sobre su función como un motivador, el cognoscitivismo postula que el conocimiento es interno al sujeto y que este conocimiento no existe como *producto*, sino que es el sujeto quien lo construye, pero esto no implica que todo el conocimiento sea subjetivo, evitando este último a través del concepto de consensos. Existe el consenso individual, interno al aprendiz donde se ve la consolidación del conocimiento y el consenso externo que se logra via negociación social con los otros aprendices y el profesor por tanto el proceso de aprendizaje es activo.

**2.4.3. La teoría constructiva:** según Torrealva [Torrealva, J., 04]. Esta teoría postula que el aprendizaje es un proceso activo, en el cual los aprendices construyen nuevas ideas o conceptos, basados en sus conocimientos anteriores, así el aprendiz selecciona y transforma la información, construye hipótesis, toma decisiones y en este proceso las estructuras cognitivas proveen significado y organización a las experiencias. Tanto docente como alumno deben comprometerse en un diálogo activo, en el que la tarea del docente consiste en introducir la información, según el estado de la comprensión del aprendiz.

Existen diferentes teorías de que explican el proceso de aprendizaje, que se pueden agrupar en:

- **Instruccional:** El profesor o tutor es el centro y el encargado de diseñar el contenido, relegando al alumno a la función de “aprender” el conocimiento, este proceso de aprendizaje es altamente guiado por el docente y no requiere la iniciativa personal del alumno.
- **Aprendizaje por descubrimiento:** Es donde el profesor tiene una tarea de “explorador”, entregándole al alumno las herramientas necesarias para que este descubra el conocimiento a su propio ritmo. Este proceso de aprendizaje no es guiado y requiere de iniciativa por parte del alumno, ya que es este el encargado de “descubrir” el conocimiento por sus propios medios, guiado por el tutor para que se puedan lograr los objetivos que surgen de la sesión pedagógica, pero dando lugar a nuevos objetivos que surgen de la experiencia del alumno.

Los procesos de aprendizaje fueron tema de estudio desde siempre. Las primeras reflexiones las expone Platón, cuando en su libro La República [Platon, 01], explica el mito de las cavernas y aclarando que: “el mundo que se conoce es la proyección de nuestras vidas innatas”. A eso se lo conoce como pensamiento racionalista, por otro lado Aristóteles, rechaza la teoría de las ideas innatas y establece que el conocimiento procede de los sentidos que dotan a la mente de

imágenes que se asocian entre sí según tres leyes: contigüidad, similitud y contraste, iniciando así el pensamiento empirista. Estas teorías son refutadas, por Santo Tomas de Aquino.

### **Teoría de la instrucción de Gagné**

El aprendizaje se define como un proceso que permite a un organismo vivo modificar sus comportamientos en forma rápida y permanente; el aprendizaje se verifica cuando existe un cambio de comportamiento, relativamente estable. Esto supone cuatro elementos: un aprendiz, una situación o entorno que permite el aprendizaje, un comportamiento explícito del aprendiz y un cambio interno.

Esta teoría establece que existen varios tipos de aprendizaje y que por lo tanto son necesarios diferentes modos de instrucción. Estos tipos son: información verbal, destrezas intelectuales: discriminar, formar conceptos, aprender reglas, estrategias cognitivas, actitudes y destrezas motoras.

Para cada tipo de aprendizaje son necesarias diferentes condiciones internas y externas. Una vez que plantea estos factores que afecta al aprendizaje, Gagné puede plantear al proceso de aprendizaje como un sistema, donde dichas entradas afectan la salida; y por lo tanto, el profesor puede mejorar la salida controlando las entradas. Gagné sugiere agregar el concepto de jerarquía como base para su descripción del funcionamiento del sistema.

### **Teoría de Equilibrio de Piaget**

Para Piaget [Piaget, 89], existen dos tipos de aprendizaje, uno el aprendizaje en sentido estricto, a través del cual se consigue información específica y otro, el aprendizaje en sentido amplio, que consiste en el progreso de las estructuras cognitivas. El aprendizaje se produce cuando se presenta un desequilibrio o conflicto cognitivo que da lugar a dos procesos: la asimilación y la acomodación.



La asimilación es la acomodación de elementos exteriores a estructuras en evolución o ya acabadas en el organismo [Piaget, 89]. El individuo interpreta la información del medio de acuerdo a sus conceptos disponibles. A través del proceso de acomodación, el sujeto adapta sus conocimientos a la realidad en tanto y en cuanto contrasta con ella lo asimilado; por otra parte se produce el cambio de estructuras cognitivas del individuo en dos sentidos, el primero como consecuencia de la suma de nuevos conceptos y el segundo como consecuencia de una reinterpretación de la ya sabido a la luz de los nuevos conocimientos.

La teoría de Piaget reduce todo el aprendizaje a adquisiciones espontaneas y necesarias, esto es un fallo puesto que la mayor parte de los conceptos en realidad no son necesarios y además, no pueden adquirirse sin la intervención de la cultura y la instrucción, es decir no son espontáneos.

### **Teoría sociocultural de Vigotsky**

Vigotsky [Vigotsky, 26], plantea que “estudiar las conductas del hombre sin lo psíquico, como pretender la psicología, es tan imposible como estudiar lo psíquico sin la conducta. Por tanto no hay lugar para ciencias distintas. El estado actual de las dos ramas del saber plantea insistentemente la cuestión de tal necesidad y fecundidad de la completa fusión de ambas ciencias”.

El aprendizaje sucede a partir de la interacción social, puede establecer que: el individuo reconstituye su saber debido a que se entremezclan procesos de construcción personal y procesos en colaboración con los otros que intervinieron. En este el maestro debe ser un agente facilitador y mediador de la cultura. En la tabla 2.1 se puede observar paralelismo entre la concepción de Piaget y la de Vigotsky.

**Figura 2.2 paralelismo entre concepción de Piaget y la de Vigotsky**

<i>Teoría de Piaget</i>	<i>Teoría de Vigotsky</i>
El conocimiento es un proceso de interacción entre el sujeto y el medio entendido físico únicamente	El conocimiento es un proceso de interacción entre el sujeto y el medio social y culturalmente
El ser humano al nacer es un individuo biológico	El ser humano al nacer es un individuo social.
La potencialidad cognoscitiva del sujeto depende de la etapa del desarrollo en la que se encuentre.	La potencialidad cognitiva del sujeto depende de la calidad de interacción social del sujeto.
El ser humano al nacer se encuentra en un estado de desorganización que deberá ir organizando a lo largo de las etapas del desarrollo de su vida.	El ser humano al nacer tiene una percepción organizada puesto que está dotado para dirigirla a estímulos humanos y para establecer relaciones sociales.

Fuente [Teoría del Aprendizaje Mc Grill ]

### **Teoría del Andamiaje de Bruner**

Esta teoría está relacionada con la de Piaget, ya que enfatiza el aprendizaje por descubrimiento. El alumno descubre, pues esto le causa mayor satisfacción y obtiene una mayor retención. Llega un paso más delante de la teoría de Piaget, desarrollando un proceso de instrucción donde el profesor debe crear un ambiente que favorezca lograr el descubrimiento [Bruner, 90]. Propone que los contenidos a enseñar se presenten como un conjunto de problemas y relaciones que el alumno debe resolver a fin de que se interese y resulte significativo el aprendizaje.

Bruner explica como el alumno llega a dominar los contenidos que el docente le propone construyendo el concepto de *andamiaje*, estrechamente ligado al de *zona de desarrollo próximo* de Vigotsky. Según esta idea, en el proceso de

interacción y diálogo y que se basa la enseñanza, el experto tiende un conjunto de andamios o ayudas por medio de las cuales el alumno elabora las construcciones necesarias para aprender los contenidos. Progresivamente, el maestro ajustará el sistema de ayudas y apoyos necesarios conforme se desarrollen las habilidades del alumno, y así poder ir cediendo el control y el manejo de los contenidos y formas de discurso a aprender.

### **Teoría del aprendizaje significativo de Ausubel [Ausubel et al., 83]**

Trata de la adquisición de aprendizaje significativo, por lo tanto se considera el aprendizaje mecánico (de memoria). Para considerar que un aprendizaje es significativo, requiere tener un sentido para ser incorporado al conjunto de conocimientos que el sujeto posee. También considera el aprendizaje como receptivo, ya que es el profesor que establece los contenidos para facilitar la organización del nuevo contenido en la estructura mental del alumno. Un aprendizaje significativo si pueden incorporarse a las estructuras de conocimiento, es decir los nuevos conceptos adquieren significado para el sujeto dentro de sus estructuras cognitivas. En general un aprendizaje significativo es más eficaz que el memorístico puesto que: produce represión más duradera, facilitan nuevos aprendizajes y produce cambios que persisten más allá del olvido de los detalles concretos.

Otro aporte de Ausubel es el concepto de los organizadores previos, sobre los cuales el alumno se apoya, de tal manera que estos hacen las veces de estructura o andamio entre los conocimientos a adquirir y los que ya poseía.

### **Inteligencias múltiples de Howard Gardner**

Esta teoría de la inteligencia humana, desarrollada por el psicólogo Howard Gardner [Gardner, 93], establece que al menos ocho formas en aquellas personas perciben y entienden el mundo. Llama a cada uno de estas formas inteligencias.

Define una inteligencia como “la capacidad de resolver problemas o de crear productos, quisieron valiosos en uno o más ambientes culturales”. Éste autor fundamenta su estructura en pruebas biológicas y antropológicas y más específicamente, en bases neurológicas, evolucionistas y transculturales. Presenta una calcificación de inteligencias, aunque sugiere que no es terminante:

- Verbal-lingüística
- Lógico-matemático
- Visual-espacial
- Kinestésica
- Musical rítmica
- Interpersonal
- Intrapersonal
- Naturalista

El objetivo de la escuela debe ser desarrollar las inteligencias y ayudar a la gente a alcanzar los fines vocacionales y aficiones que se adecuen a su particularidad espectro de inteligencias.

#### **2.4.4. Técnicas Educativas Aplicables**

Una vez presentadas las técnicas educativas más importantes, se debe analizar cuáles de ellas son aplicables a la enseñanza de lenguajes de programación con paradigma lineal estructurado y dentro de ellas, cuáles son implementables a un sistema tutor inteligente. Cualquiera de las teorías de enseñanza explicadas resulta útil para entender el proceso de aprendizaje de lenguajes de programación. La selección de un marco teórico adecuado sólo puede justificarse por medio de razones pedagógicas, correspondientes al tipo de alumnos con el que se va a enfrentar al sistema a las técnicas de programación o características disponibles para el futuro sistema [Perkins, 95].

Como no es el propósito de esta tesis de evaluar los distintos modelos de enseñanza y de aprendizaje, por lo que se selecciono la *Teoría uno* [Perkins, 95], que no es un modelo, ni un método de enseñanza, sino un conjunto de recomendaciones compatibles con cualquier teoría. Básicamente, estipula que *“la gente aprende más cuando tienen una oportunidad razonable y una motivación para hacerlo”*.

Por lo tanto se tomarán la Teoría Uno, como la base teórica del Sistema Tutor Inteligente (STI), en la que se toman las bases de las distintas formas de enseñanzas disponibles. Perkins plantea que si se combinan las condiciones que estipula la Teoría Uno se “personifican” las distintas maneras según el programa del momento. Las cuatro formas principales se enumeran a continuación:

- a) según Perkins es la representación clara y concreta de la información por parte de los maestros y de los textos. Su objetivo se centra especialmente en la explicación donde se exponen *“los que”* y *“los porque”* de un determinado tema. Es fundamental que se aclaren los componentes básicos de una buena explicación, ya que la instrucción didáctica el alumno queda relegado a un papel de sujeto pasivo y el tutor o maestro queda en primer plano de proveedor único de los saberes y conocimientos.
- b) El entrenamiento. Existe un vínculo entre el entrenamiento y la instrucción didáctica y por eso se representa en segundo lugar. Sin una instrucción didáctica que presente cierta base de información sobre los conceptos nuevos, los estudiantes carecerían de los conocimientos básicos para realizar las ejercitaciones prácticas. Se plantea una analogía muy clara con respecto al fútbol *“el entrenador observa desde afuera el desempeño de los deportistas y les da consejos. Elogia los puntos fuertes, detecta los débiles, hace observar ciertos principios, ofrece guía e inspiración y decide que todo tipo de prácticas se deben enfatizar”*. Y es ésta función tan importante en la que se extrapola y se aplica en cualquier dominio, será este fútbol, clase de matemática o literatura.



- c) **La enseñanza socrática.** Ambos métodos vistos con anterioridad poseen un aspecto regulativo, pues sus funciones consisten en modelar y guiar las actividades de los alumnos y cómo se explicó anteriormente se relega al estudiante a un papel secundario, en este marco surge el método socrático, a través del cual se logra que los estudiantes trabajen de manera más flexible, pero no en forma libre sino que continuamente estén recibiendo el apoyo de sus investigaciones por parte de un maestro o tutor, pero sin el esquema más elegido en el que el tutor les dice todo el tiempo lo que tienen que hacer. La enseñanza se realiza cuando el maestro socrático plantea un enigma conceptual e incita a investigar el asunto de manera libre por parte de los estudiantes, haciendo preguntas del tipo: ¿qué piensan al respecto?, ¿Qué posición se podría tomar?, ¿Qué definiciones necesitamos? Luego de la investigación inicial se proponen ideas, criterios y definiciones, donde el maestro actúa como incitador y moderador en la conversación. Uno de los conceptos claves de la teoría uno, es que el maestro socrático no provea abundancia de datos, pero controle la claridad y la calidad de la información suministrada por los alumnos haciéndole preguntas. Cuando los alumnos discuten entre sí sobre una determinada cuestión, el maestro socrático les exige una práctica un clima de reflexión, provee además, realimentación inmediata por medio de estímulos y críticas, e incita aquí todos los participantes de la conversación hagan lo mismo.
- d) **Otros métodos de enseñanza.** Los protocolos pedagógicos puestos en la teoría uno no son los únicos acercamientos que proveen las características requeridas por la teoría uno: "información clara, práctica reflexiva, realimentación informativa y fuerte motivación intrínseca y extrínseca". Existen otras aproximaciones como pueda serlo el aprendizaje cooperativo y la colaboración entre partes, que vale la pena resaltar en el marco de los asesores inteligentes, donde el sistema debe reaccionar como un "estudiante virtual" que interactúan con los estudiantes usuarios del sistema para resolver los distintos tipos de problemas que se presentan y para lograr definiciones básicas de los conceptos que se pretende enseñar en una sesión pedagógica

a través del "andamiaje" [Ausubel et al., 83] , como lo hará un ayudante alumno.

Perkins sostiene que los niños aprenden mucho mejor en grupos cooperativos bien configurados que en soledad. Por lo general, las agrupaciones cooperativas pueden ayudar a lograr determinados fines, sin la planificación cuidadosa que reúne los métodos como la instrucción didáctica y el entrenamiento los resultados obtenidos son satisfactorios porque el aprendizaje cooperativo exige que todos los participantes se hagan responsables del desempeño del grupo.

#### **2.4.5. Que es Enseñanza**

"Si lo que ha de aprenderse evoluciona, y nadie duda de que evoluciona y cada vez a mayor velocidad, la forma en que ha de aprenderse y enseñarse también debería evolucionar" señala Pozo [Pozo, J., 96].

Almeida y colegas [Almeida, S. et al., 97] definen medio de enseñanza como todos los componentes del proceso docente que actúan como soporte material de los métodos (instructivos o educativos) con el propósito de lograr los objetivos planteados.

La enseñanza programada supone que los alumnos pueden aprender cierto material más efectivamente si está programado, o sea si está organizado lógicamente para presentar sólo una idea elemental cada vez, según explica Mateo [Mateo, A., 97]. Cada idea se construye a partir de las previas, cada nuevo elemento de información va seguido por una serie de preguntas y ejercicios con las correspondientes respuestas, de modo que el alumno pueda integrar y revisar las informaciones previas y vigilar por sí mismo la marcha del trabajo. El problema es preparar el material. También, según esta autora, resulta problemático la rigidez del orden prefijado. Se le podría agregar elementos tales que "si resolvió tal problema, realizar tal otro", lo cual es sencillo de hacer con computadora.

## 2.5 El Método

Se entiende por método a: "Reglas ciertas y fáciles, gracias a las cuales quien las observe exactamente no tomará nunca lo falso por verdadero, y llegará, sin gastar inútilmente esfuerzo alguno de su espíritu, sino aumentando siempre gradualmente, su ciencia, al verdadero conocimiento de todo aquello de que sea capaz".

### 2.5.1 Clases De Métodos

Según John Stuart Mill<sup>3</sup> los métodos son cuatro: el de concordancia, el de diferencia, el de variaciones concomitantes y el de residuos:

- **Método De Concordancia**. Si dos o más casos tiene una circunstancia común, ésta es la causa (o efecto) del fenómeno. Se trata de estudiar casos diferentes para ver en qué concuerdan.
- **Método De Diferencia**. Si un caso donde se presenta el fenómeno y otro donde no se presentan tiene todas las circunstancias comunes menos una, esa es la causa (o parte de la causa) del fenómeno. Se trata de buscar casos que se parezcan en todas sus circunstancias y difieren en alguna.
- **Método Conjunto De Concordancia Y Diferencia**. Se trata de la utilización conjunta de los otros dos métodos: una concordancia y una diferencia.
- **Método De Variaciones Concomitantes**. Se trata de establecer relaciones de causa y efecto entre dos fenómenos. Los fenómenos estudiados podrían ser ambos efectos de una misma causa.
- **Método De Residuos**. Se trata de averiguar las causas cuya presencia no puede ser eliminada por experimentación.

La característica común de estos métodos, el verdadero método de la investigación científica, es el de la eliminación. El resto de métodos están, por lo tanto, subordinados al método de la diferencia.

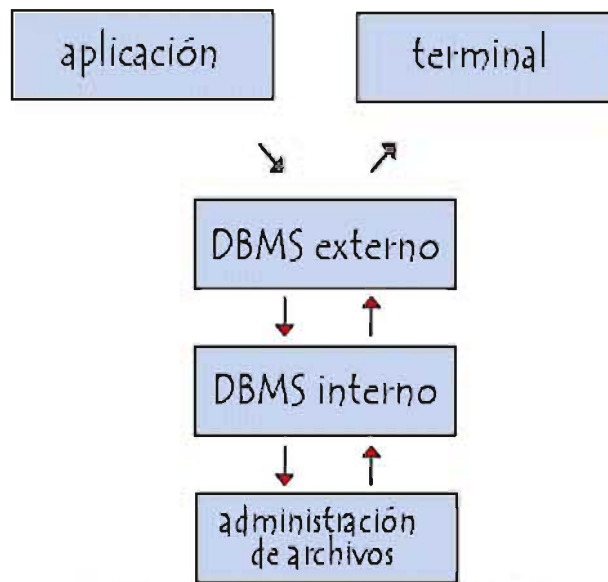
## 2.6 Sistema De Gestión De Base De Datos (SGBD)

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz



entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

**Figura 2.2 Administración de Base de Datos**



Fuente [Base de Datos Wikipedia]

## 2.7 Control Sobre La Redundancia De Datos

Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.

En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

### **2.7.1 Consistencia De Datos**

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes.

### **2.7.2 Compartición De Datos**

En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

### **2.7.3 Mantenimiento De Estándares**

Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

### **2.7.4 Mejora En La Integridad De Datos**

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargarse de mantenerlas.

### **2.7.5 Mejora En La Seguridad**

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de

datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

### **2.7.6 Mejora En La Accesibilidad A Los Datos**

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

### **2.7.7 Mejora En La Productividad**

El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.

El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

### **2.7.8 Mejora En El Mantenimiento**

En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

### **2.7.9 Aumento De La Concurrencia**

En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre

ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

#### **2.7.10 Mejora En Los Servicios De Copias De Seguridad**

Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

#### **2.7.11 Desventajas De Las Bases De Datos**

##### **Complejidad**

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

##### **Coste del equipamiento adicional**

Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

##### **Vulnerable a los fallos**

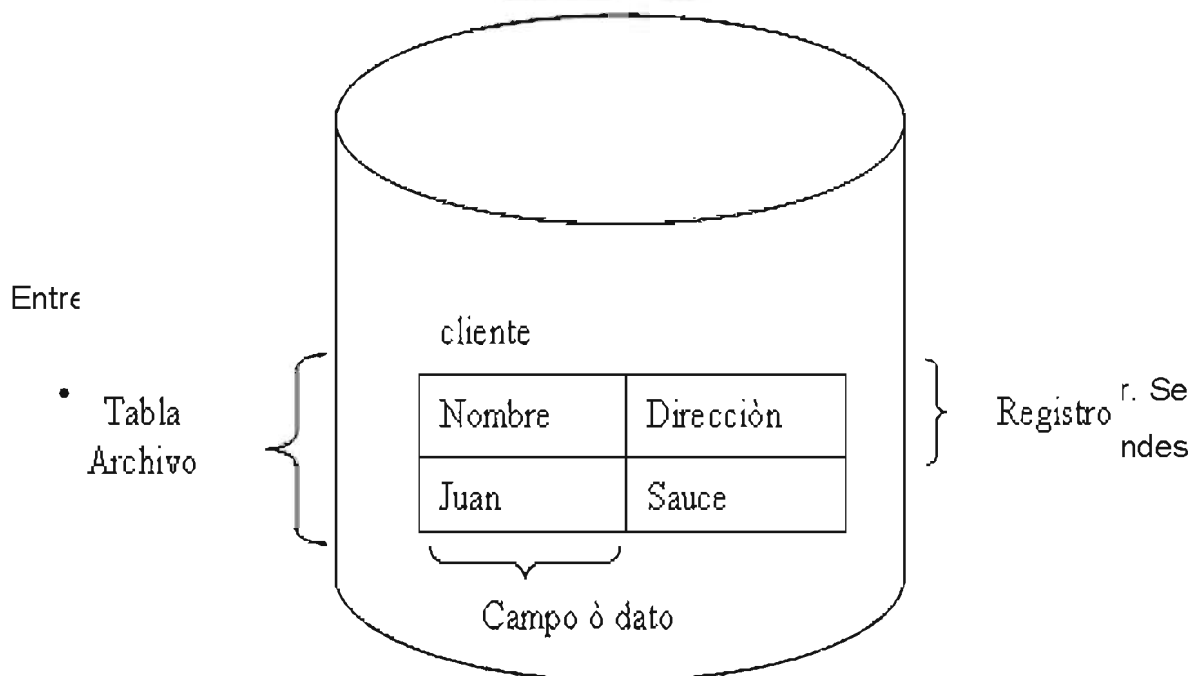
El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backup).

## 2.7.12 Tipos De Campos

Cada Sistema de Base de Datos posee tipos de campos que pueden ser similares o diferentes. Entre los más comunes podemos nombrar:

- **Numérico:** entre los diferentes tipos de campos numéricos podemos encontrar enteros “sin decimales” y reales “decimales”.
- **Booleanos:** poseen dos estados: Verdadero “Si” y Falso “No”.
- **Memos:** son campos alfanuméricos de longitud ilimitada. Presentan el inconveniente de no poder ser indexados.
- **Fechas:** almacenan fechas facilitando posteriormente su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.
- **Alfanuméricos:** contienen cifras y letras. Presentan una longitud limitada (255 caracteres).
- **Autoincrementables:** son campos numéricos enteros que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta: Servir de identificador ya que resultan exclusivos de un registro.

Figura 2.3 Registro de una Base de Datos



- **PostgreSQL y Oracle:** Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.
- **Access:** Es una base de datos desarrollada por **Microsoft**. Esta base de datos, debe ser creada bajo el programa access, el cual crea un archivo .mdb con la estructura ya explicada.
- **Microsoft SQL Server:** es una base de datos más potente que access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones.

## 2.8 Tipos De Base De Datos

### Según la variabilidad de los datos almacenados

#### 2.8.1 Bases de datos estáticas

Son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

#### 2.8.2 Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa.



### **2.8.3 Bases de datos bibliográficas**

Sólo contienen un subrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no, estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias —ver más abajo). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

### **2.8.4 Bases de datos de texto completo**

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

### **2.8.5 Bases de datos o "bibliotecas" de información química o biológica**

Son bases de datos que almacenan diferentes tipos de información proveniente de la química, las ciencias de la vida o médicas. Se pueden considerar en varios subtipos:

- Las que almacenan secuencias de nucleótidos o proteínas.
- Las bases de datos de rutas metabólicas.
- Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas
- Bases de datos clínicas.
- Bases de datos bibliográficas (biológicas, químicas, médicas y de otros campos): PubChem, Medline, EBSCOhost.

### **2.8.6 Modelos de Bases de Datos**

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como *contenedor de datos* (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de *base de datos*; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

### **2.8.7 Bases de Datos jerárquicas**

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se los conoce como *hojas*.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

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

### **2.8.8 Base de Datos de red**

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

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado



que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

### **2.8.9 Bases de Datos transaccionales**

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos, por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

### **2.8.10 Bases de Datos relacionales**

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

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

almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

Durante los años 80 la aparición de dBASE produjo una revolución en los lenguajes de programación y sistemas de administración de datos. Aunque nunca debe olvidarse que dBase no utilizaba SQL como lenguaje base para su gestión.

#### **2.8.11 Bases de Datos multidimensionales**

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de **Cubos OLAP**. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

#### **2.8.12 Bases de Datos orientadas a objetos**

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento).

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

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

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

SQL:2010, es el estándar de SQL ampliado, soporta los conceptos orientados a objetos y mantiene la compatibilidad con SQL.

### **2.8.13 Bases de Datos documentales**

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesauros es un sistema de índices optimizado para este tipo de bases de datos.

### **2.8.14 Bases de Datos deductivas**

Un sistema de base de datos deductiva, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa

principalmente en reglas y hechos que son almacenados en la base de datos. Las bases de datos deductivas son también llamadas bases de datos lógicas, a raíz de que se basa en lógica matemática. Este tipo de base de datos surge debido a las limitaciones de la Base de Datos Relacional de responder a consultas recursivas y de deducir relaciones indirectas de los datos almacenados en la base de datos. Utiliza un subconjunto del lenguaje Prolog llamado Datalog el cual es declarativo y permite al ordenador hacer deducciones para contestar a consultas basándose en los hechos y reglas almacenados.

### **Ventajas**

- Uso de reglas lógicas para expresar las consultas.
- Permite responder consultas recursivas.
- Cuenta con negaciones estratificadas
- Capacidad de obtener nueva información a través de la ya almacenada en la base de datos mediante inferencia.
- Uso de algoritmos de optimización de consultas.
- Soporta objetos y conjuntos complejos.

### **Desventajas**

- Crear procedimientos eficaces de deducción para evitar caer en bucles infinitos.
- Encontrar criterios que decidan la utilización de una ley como regla de deducción.
- Replantear las convenciones habituales de la base de datos.

## **2.9 Álgebra Relacional**

- Es un conjunto de operadores a disposición del usuario para manipular los objetos de la base de datos. Este conjunto es reducido y cerrado (toma como entrada una o dos relaciones y como salida obtiene una nueva relación). Dentro de los operadores del álgebra relacional tenemos:

- Operadores puramente algebraicos. Son los operadores típicos de conjuntos.  $\cup$  (unión),  $\cap$  (intersección) y  $-$  (diferencia) sólo pueden ser aplicados cuando los esquemas de las relaciones operando sean iguales.  $\times$  (producto cartesiano) obtiene una relación cuyos atributos son resultado de la concatenación de los atributos de los operandos ( $R(A,B,C) \times S(B,D,E) = R \times S(A,B,C,B,D,E)$ ).
- Operadores relacionales. Los principales son selección ( $\sigma$ ), proyección ( $\Pi$ ), concatenación o "join" ( $\bowtie$ ) y división ( $\div$ ).
- Selección. Dada la relación  $R$  se define la relación  $\sigma_p$  como el conjunto de tuplas de  $R$  que satisfacen el predicado  $p$ :  $\sigma_p(R) = \{t \in R \mid P(t)\}$ . El esquema de la relación resultante sería  $R$ , es decir, el mismo que el de la relación  $R$  (Obtenemos un subconjunto horizontal). Ejemplo: Militar(DNI#, Nombre, Rango, Edad).  $\sigma_{\text{Rango}="Cabo"}(\text{Militar}) = \{\text{Cabos de la base}\}$
- Proyección. Si  $x \subseteq R$  se define  $\Pi_x(R)$  como el conjunto de las proyecciones de las tuplas de  $R$  sobre el atributo  $x$ :  $x \subseteq R$ ,  $\Pi_x(R) = \{t[x] \mid t \in R\}$ . El esquema obtenido en la relación resultante sería  $x$  (Obtenemos un subconjunto vertical de la base de datos). En el caso de obtenerse duplicaciones en el resultado, estas son eliminadas automáticamente.
- Ejemplo: Militar(DNI#, Nombre, Rango, Edad).  $\Pi_{\text{Rango}}(\text{Militar}) = \{\text{Rangos distintos}\}$

Ejemplo: Militar(DNI#, Nombre, o, Edad).  $\Pi_{\text{Rango,DNI\#}}(\text{Militar}) = \{\text{Rangos y DNI's}\}$

- Concatenación. Existen dos tipos de concatenación: Join natural y  $\theta$ \_join.

➤ *Join natural*: Consideremos  $R(R)$  y  $S(S)$  tales que  $R \cap S \neq \emptyset$  (tienen algún atributo en común). Se define  $(R \bowtie S)$  como una relación cuyo esquema es la concatenación de los dos esquemas eliminando en uno de ellos los atributos repetidos y cuyos datos son el conjunto resultante de la unión de los datos de las relaciones anteriores

Ejemplo:

R	(A, B)	S	(B, C)	$R \bowtie S$	(A, B, C)	
	a1 b1		b1 c1	a1 b1 c1		
	a2 b1		b2 c2	a2 b1 c1		
	a1 b2		b2 c1	a1 b2 c1		
				a1 b2 c2		

Atributo sobre el símbolo  $\bowtie$ . Generalmente, el join natural se usa para conectar relaciones a través de las claves externas.

A=D

R	(A, B)	S	(B, C, D)	$R \bowtie S$	(A, R.B, S.B, C, D)	
	a1 b1		b1 c1 d1	a1 b1 b1 c1 d1		
	a2 b1		b2 c1 d1	a1 b1 b2 c1 d1		
	a1 b2		b2 c1 d2	a2 b1 b2 c1 d2		
				a1 b2 b1 c1 d1		
				a1 b2 b2 c1 d1		



sus relaciones en un atributo común, el join natural la basa en una relación previamente establecida en la que intervienen operadores relacionales  $\{=, <, >, >=, <=, <>\}$ .

División. Sea  $R(A_1, A_2, A_3, \dots, A_n, B_1, B_2, \dots, B_m)$  y  $S(B_1, B_2, \dots, B_m)$ . El resultado de la división es una nueva relación  $(R \div S)(A_1, A_2, A_3, \dots, A_n)$  en la que las tuplas pertenecientes a la misma verifican que  $\forall t' = \langle y \rangle \in S$  tiene que existir una tupla  $t' = \langle xy \rangle \in R$  donde  $\langle y \rangle = (B_1, \dots, B_m)$  y  $\langle x \rangle = (A_1, \dots, A_n)$ .

Ejemplo:  $\text{Piloto}(\text{DNI\#}, \text{Avión}), \text{Aviones}(\text{Avión}), \text{Piloto} \div \text{Aviones}(\text{DNI\#}) = \{\text{Pilotos que pilotan todos los tipos de aviones de Aviones}\}$

Operadores adicionales. Dentro de este tipo de operadores nos encontramos con dos principales: el operador adición ( $\oplus$ ) y el operador agrupamiento ( $\Sigma$ ).

Adición. Sirve para añadir a una relación un determinado atributo en donde el valor que tiene una tupla sobre ese nuevo atributo es el resultado de la expresión que aparece en este formato de la expresión:  $\oplus_{\text{Expresión}} \text{Nuevo\_atributo}(\text{Relación})$ .

Ejemplo:  $\text{Productos}(\text{Nombre}, \text{Precio}, \text{Color}), \oplus_{\text{Precio} * 1,15} \text{Precio\_IVA}(\text{Productos})$

Podemos eliminar el nuevo atributo proyectando la nueva relación sobre el resto de los atributos.

Agrupamiento. Sirve para reordenar las tuplas de una relación formando grupos. El formato que tiene es el siguiente:  $\Sigma_{\text{Atributo}} [\oplus_{\text{Función\_agregada}} \text{Nuevo\_Atributo} \dots](R)$  que produce un ordenamiento de la relación R por el atributo especificado (La parte entre corchetes es opcional). El esquema de la relación resultante es el atributo sobre el que se aplica la misma y los nuevos atributos añadidos si es que los hay.

Ejemplo:  $\Sigma_{\text{DNI\#}}(\text{Alumno})$  crea una lista de DNI's de alumnos ordenados.

Ejemplo:  $\Sigma_{\text{Nombre}} \oplus_{\text{Media}(\text{Precio})} \text{PrecioMedio} \oplus_{\text{Cuenta}(\text{Ciudad})} \text{NumeroCiudades}(\text{Producto})$  con la relación  $\text{Producto}(\underline{\text{Nombre}}, \underline{\text{Ciudad}}, \text{Precio})$  crea una nueva relación cuyo



esquema resultante es (Nombre, PrecioMedio, NumeroCiudades) ordenada por nombres que contiene los precios medios de los productos especificados por el nombre y el número de ciudades que venden dicho producto.

Las funciones agregadas para este tipo de operación son cuenta, suma, media, máximo y mínimo entre otras.

### 2.9.2 Conversión de Diagramas E/R a tablas.

La regla básica consiste en convertir cada una de las entidades del diagrama en una relación, cuya clave primaria será el identificador de la entidad (En el caso de darnos cuenta de que esto no sucede así es seguramente porque el diagrama del que partimos está mal diseñado). Tenemos los siguientes casos:

- Relación 1:1 de A(A#, otros) con B(B#, otros). En una de las dos relaciones se introduce como clave externa el atributo identificativo de la otra: A(A#, otros, B#) y B(B#, otros) con B# clave externa de A para la relación B. En el caso de opcionalidad a uno de los lados, la clave externa se presentará en la relación correspondiente a la entidad obligatoria.
- Relación 1:N de A(A#, otros) con B(B#, otros) donde A es el lado 1 y B el lado muchos. Se introduce en la relación B el atributo identificativo de la clave A como clave externa quedando A(A#, otros) y B(B#, otros, A#).
- Relación N:M de A(A#, otros) con B(B#, otros). En este caso se crea una nueva relación con el nombre de la asociación que contiene los atributos identificadores de A y B además de los suyos propios en el caso de existir. En el caso de que la clave primaria no coincida con la concatenación de las claves primarias de A y B seguramente es porque hemos realizado un diseño erróneo de la base de datos. Las relaciones resultantes por tanto quedarían: (A#, otros), B(B#, otros), Asociación(A#, B#, otros).
- Asociaciones unarias. El funcionamiento es similar al de las relaciones binarias.

- Asociación ternaria 1:1:1. Además de las relaciones ya existentes se crea una nueva con los atributos de la misma y con las claves externas de dos de las relaciones que participan en la asociación.
- Asociación ternaria 1:1:N. Al igual que en las de tipo 1:1:1 se crea una nueva relación con las claves externas de dos de las relaciones que participan en la asociación, teniendo en cuenta que una de las dos ha de ser la clave primaria de la relación del lado N de la asociación.
- Asociación ternaria 1:N:M. Se crea una nueva asociación cuya clave primaria es resultado de la concatenación de las claves externas para las relaciones del lado N y M.
- Asociación ternaria N:M:Ñ. Se crea una nueva asociación cuya clave primaria corresponde con la concatenación de las tres claves primarias de las relaciones integrantes.
- En las especializaciones y generalizaciones se emplean asociaciones diferentes para supertipos y subtipos siempre y cuando cada uno aparezca en una asociación diferente. En el caso de no aparecer en asociaciones no merece la pena la representación de estas.

### 2.9.3 Teoría de la normalización

La teoría de la normalización es una teoría sobre el diseño de relaciones que nos proporciona una herramienta que permite definir el nivel de diseño que tiene una relación y herramientas para descomponer una relación y obtener relaciones que estén en el nivel de diseño adecuado.

Existen dos tipos fundamentales de defectos a la hora de diseñar una relación: redundancia de datos y anomalías de inserción/borrado.

Supongamos que tenemos la siguiente relación:

ESTUDIANTE\_INFO (Nombre,Cod\_Curso,Tfno,Asigantura,Profesor,Nota)

En este ejemplo hay redundancia de datos, pues un alumno puede estar matriculado en varios cursos. También el nombre de la asignatura o del profesor se repite en todas las tuplas de los alumnos, etc. Estas redundancias pueden producir inconsistencias y poca flexibilidad (p.ej. si un profesor se da de baja hay que cambiar todas las tuplas de los alumnos).

Las anomalías de inserción se refieren a que no se puede almacenar una información determinada. Por ejemplo, si se crease una nueva asignatura en la que aún no se ha matriculado ningún alumno, cómo representaríamos esa asignatura. No podemos poner (null,null,null,asignatura,profesor,null) pues tendríamos claves primarias con valor null.

Las anomalías de borrado se refieren a perder información relevante al intentar borrar una información no relevante. Por ejemplo si sólo hay un alumno matriculado en una asignatura y borramos el alumno, tal y como está construida la relación borraríamos también la asignatura.

La teoría de normalización propone la posibilidad de definir el nivel de diseño y propone herramientas para obtener, a partir de una relación otras relaciones tales que todas esas nuevas relaciones se encuentran en un nivel de diseño mayor que la relación original.

La descomposición de una relación debe verificar que:

sea reversible por concatenación: el join de las relaciones resultantes de la descomposición debe dar como resultado la relación original. Toda la información semántica de la relación original debe permanecer en las relaciones resultado, esto es, deben mantenerse las dependencias funcionales.

#### **2.9.4 Dependencias funcionales**

Son herramientas sencillas que nos permiten formalizar parte de la semántica de la relación.

Sea  $R$  una relación con el esquema  $R(X,Y,Z)$ , se dice que  $R$  satisface la dependencia funcional  $X \rightarrow Y$  ( $X$  determina funcionalmente a  $Y$ , ó  $Y$  depende funcionalmente de  $X$ ) si cualquiera que sea  $x$  valor del dominio de  $X$ , la relación  $\Pi_y$  ( $\sigma_{X=x}(R)$ ) contiene a la sumo UNA tupla (en otras palabras, asociado a un único valor de  $x$  hay un único valor de  $y$ ). A  $X$  se le conoce como *antecedente* y a  $Y$  como *consecuente*.

Ejemplo:

$R(A B C D)$

$a1\ b2\ c1\ d1$   $B \rightarrow D$  cierto, todas las tuplas con un valor de  $B$  tienen

$a1\ b3\ c1\ d2$  asociado un único valor de  $D$  ( $b2 \rightarrow d1, b3 \rightarrow d2$ )

$a2\ b2\ c2\ d1$   $A \rightarrow D$  falso, pues  $a1 \rightarrow d1$  y  $a1 \rightarrow d2$

$a1\ b2\ c2\ d1$   $CD \rightarrow B$  cierto:  $c1d1 \rightarrow b2$   $c1d2 \rightarrow b3$   $c2d1 \rightarrow b2$

La dependencia funcional es algo virtual, pues si añadimos una nueva tupla las dependencias se pueden romper. Por eso nos fijaremos en las dependencias funcionales asociadas al esquema de la relación, a la parte invariable, para así imponer que cualquier extensión válida de la relación satisfaga esa relación válida.

Sea  $\mathfrak{R}$  un esquema de relación y sean  $X, Y \subseteq \mathfrak{R}$ , decimos que  $X \rightarrow Y$  está asociado a  $\mathfrak{R}$  si  $\forall R(\mathfrak{R})$ ,  $R$  satisface necesariamente la dependencia funcional  $X \rightarrow Y$ .

A partir de ahora, a cada relación le asociaremos un conjunto  $\mathbb{F}$  de dependencias funcionales.

## 2.9.5 AXIOMAS DE AMSTRONG

Son axiomas que aplicaremos para desarrollar las reglas funcionales. Consideraremos que tenemos una relación  $\mathfrak{R}$  y un conjunto de atributos  $X, Y, Z, W \subset \mathfrak{R}$  que pueden ser simples o compuestos.

Axioma 1- Reflexividad: Cualquier relación  $R(\mathfrak{R})$  satisface que  $X \rightarrow X$ .

Axioma 2- Aumentación: Si  $R$  satisface  $X \rightarrow Y$  entonces  $R$  satisface  $XZ \rightarrow Y$

Demostración:

$$t_1, t_2 \in R / t_1(XZ) = t_2(XZ) \implies (\text{en particular}) \implies t_1(X) = t_2(X)$$

$$\text{como por hipótesis } R \text{ satisface } X \rightarrow Y \implies t_1(Y) = t_2(Y)$$

Axioma 3- Aditividad: Si  $R$  satisface  $X \rightarrow Y$  y  $X \rightarrow Z$  entonces  $R$  satisface  $X \rightarrow YZ$ .

Demostración:

$$t_1, t_2 \in R / t_1(X) = t_2(X),$$

$$\text{por hipótesis } X \rightarrow Y \implies t_1(Y) = t_2(Y) \mid$$

$$X \rightarrow Z \implies t_1(Z) = t_2(Z) \mid \implies t_1(YZ) = t_2(YZ)$$

Axioma 4- Proyectividad: Si  $R$  satisface  $X \rightarrow YZ \implies R$  satisface  $X \rightarrow Y$

Demostración:

$$t_1, t_2 \in R / t_1(X) = t_2(X)$$

$$\text{por hipótesis } X \rightarrow YZ \implies t_1(YZ) = t_2(YZ) \implies t_1(Y) = t_2(Y) \text{ y } t_1(Z) = t_2(Z)$$

Axioma 5- Transitividad:

Si  $R$  satisface  $X \rightarrow Y$  y  $Y \rightarrow Z$  entonces  $R$  satisface que  $X \rightarrow Z$ .

Demostración:

$$t_1, t_2 \in R / t_1(X) = t_2(X) \text{ y como } X \rightarrow Y \implies t_1(Y) = t_2(Y)$$

Como R por hipótesis  $Y \rightarrow Z$   $t_1(Y)=t_2(Y) \Rightarrow t_1(Z)=t_2(Z) \Rightarrow X \rightarrow Z$

Axioma 6- Pseudotransitividad: Si R satisface  $X \rightarrow Y$  y  $YW \rightarrow Z \Rightarrow XW \rightarrow Z$ .

Demostración:

$$t_1, t_2 \in R / t_1(XW)=t_2(XW) \Rightarrow t_1(X)=t_2(X) \text{ (1) y } t_1(W)=t_2(W) \text{ (2)}$$

$$\text{por hipótesis } X \rightarrow Y \Rightarrow (1) \Rightarrow t_1(Y)=t_2(Y) \Rightarrow (2) \Rightarrow t_1(YW)=t_2(YW)$$

$$\Rightarrow YW \rightarrow Z$$

Con estas reglas de Armstrong podemos obtener nuevas dependencias funcionales. Un conjunto de dependencias funcionales  $\mathbb{F}$  infiere una dependencia funcional  $X \rightarrow Y$  (y se representa:  $\mathbb{F} \vdash X \rightarrow Y$ ) si la dependencia funcional  $X \rightarrow Y$  se puede obtener a partir de  $\mathbb{F}$  por aplicación de los axiomas de Armstrong. Por definición,  $\mathbb{F}$  infiere a todas las dependencias funcionales de  $\mathbb{F}$ , esto es,

$$X \rightarrow Y \vdash X \rightarrow Y.$$

### 2.9.6 Clausuras

Al conjunto de dependencias funcionales que se infieren a partir de  $\mathbb{F}$  se lo denomina clausura de  $\mathbb{F}$  y se denota con  $\mathbb{F}^+$ .

$$\mathbb{F}^+ = \{X \rightarrow Y / \mathbb{F} \vdash X \rightarrow Y\}$$

$\mathbb{F}^+$  es un conjunto tal que incluye a  $\mathbb{F}$  y tal que si aplicamos los axiomas de Armstrong a sus dependencias funcionales no se obtienen nuevas dependencias funcionales, pues ya incluye a todas las posibles. Por ello  $(\mathbb{F}^+)^+ = (\mathbb{F}^+)$ .

Ejemplo:  $R=(A,B,C)$   $\mathbb{F}=\{A \rightarrow B, B \rightarrow C\}$

$$\mathbb{F}^+ = \{A \rightarrow B, B \rightarrow C, A \rightarrow A, B \rightarrow B, C \rightarrow C, AB \rightarrow A, ABC \rightarrow A, AB \rightarrow C, BC \rightarrow C, \dots\}$$



Como  $\mathbb{F}^+$  es un conjunto muy grande, necesitaremos buscar una herramienta que nos permita saber si una dependencia funcional pertenece a  $\mathbb{F}^+$  sin tener que calcular  $\mathbb{F}^+$ .

Definición: Sea  $\mathfrak{R}$  un esquema de relación y  $X$  un atributo del esquema  $\mathfrak{R}$  ( $X \in \mathfrak{R}$ ), y sea  $\mathbb{F}$  un conjunto de dependencias funcionales sobre  $\mathfrak{R}$ , se llama clausura de  $X$  respecto de  $\mathbb{F}$ , y se denota con  $X_r^+$  al atributo más grande de  $\mathfrak{R}$  tal que  $X \rightarrow X_r^+ \in \mathbb{F}^+$ .

Ejemplo:  $R(A,B,C) \quad \mathbb{F}=\{A \rightarrow B, B \rightarrow C\}$

$A \rightarrow A$  y  $A \rightarrow B \Rightarrow A \rightarrow AB$  |

$B \rightarrow C$  y  $A \rightarrow C \Rightarrow A \rightarrow AC$  |  $\Rightarrow A \rightarrow ABC \quad A_r^+ = ABC$

### Proposición

Sea  $\mathfrak{R}$  un esquema de relación, y  $X, Y \subset \mathfrak{R}$ ,  $\mathbb{F}$  conjunto DFs sobre  $\mathfrak{R}$ , entonces  $X \rightarrow Y \in \mathbb{F}^+$  si y solo si  $Y \in X_r^+$

Demostración:

$\Rightarrow$ ) supongamos que  $X \rightarrow Y \in \mathbb{F}^+$ , como  $X \rightarrow X_r^+ \in \mathbb{F}^+$  y  $X_r^+$  es el más grande, entonces  $Y \in X_r^+$  pues de no ser así  $X_r^+$  no sería el más grande al faltarle  $Y$ .

$\Leftarrow$ )  $X \rightarrow X_r^+ \in \mathbb{F}^+$  como por hipótesis  $Y \in X_r^+$  entonces por el axioma de proyectividad  $X \rightarrow Y \in \mathbb{F}^+$

### 2.9.7 Algoritmo para calcular $x$

Sea  $R(\mathfrak{R})$  y  $\mathbb{F}$  conjunto de DFs sobre  $\mathfrak{R}$ ,  $X \subset \mathfrak{R}$

Inicialmente  $X_r^+ = X$

Cambio = TRUE



```

while (cambio)
    cambio=false
    for (cada DF W->Z ∈ F)
        if w ⊂ Xr+ then Xr+ = Xr+ ∪ Z; cambio=true;
    endfor
endwhile
END.

```

Ejemplo:  $F = \{A \rightarrow G, CD \rightarrow E, E \rightarrow C, D \rightarrow AH, AB \rightarrow D, DH \rightarrow BC\}$

inicio cd->e d->ah

$(BCD)_r^+ = BCD$  E AH = BCDEAH (primera pasada)

a->g

$(BCD)_r^+ = BCDEAH$  G Resultado = BCDEAHG

Supongamos que tenemos  $R(\mathbb{R})$   $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$  y  $G = \{A \rightarrow B, B \rightarrow C\}$ . Si observamos  $G$  y  $F$  notamos que ambos proporcionan la misma información: la dependencia  $A \rightarrow C$ , la podemos conocer con  $G$  y los axiomas de Armstrong.  $G$  y  $F$  recogen en esencia la misma información, y nos interesa más  $G$  por ser más manejable.

Definición: Dos conjuntos de dependencias funcionales  $F$  y  $G$  se dicen equivalentes y se de nota con  $F \equiv G$  cuando sus clausuras son iguales:  $F^+ = G^+$ . Interesa eliminar las dependencias redundantes (aquellas que pueden obtenerse con el resto de las dependencias) y reducir en lo que podamos el conjunto de Dfs.

Definición: Sea  $R(\mathcal{R})$  y  $\mathbb{F}$  un conjunto de Dfs sobre  $\mathcal{R}$ ,  $\mathbb{F}$  se dice redundante si existe un  $\mathbb{F}' \subset \mathbb{F}$  tal que  $\mathbb{F}' \equiv \mathbb{F}$ .

Ejemplo:  $\mathbb{F} = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, A \rightarrow C\}$  ¿es redundante?. Observemos cada una de las dependencias:

$\mathbb{F} - \{A \rightarrow B\} \not\models A \rightarrow B?$

$A^+_{\mathbb{F} - \{A \rightarrow B\}} = AC$  (no está  $B \Rightarrow$  esta dependencia no es redundante)

$\mathbb{F} - \{B \rightarrow A\} \not\models B \rightarrow A?$   $B^+_{\mathbb{F} - \{B \rightarrow A\}} = BC$  (no está  $A \Rightarrow$  esta dependencia no es redundante)

$\mathbb{F} - \{B \rightarrow C\} \not\models B \rightarrow C?$

$B^+_{\mathbb{F} - \{B \rightarrow C\}} = BAC$  (está  $C \Rightarrow$  esta dependencia es redundante):

Elimina y trabajamos con el resto:

$\mathbb{F}' = \{A \rightarrow B, B \rightarrow A, A \rightarrow C\}$

$\mathbb{F}' - \{A \rightarrow C\} \not\models A \rightarrow C?$   $A^+_{\mathbb{F}' - \{A \rightarrow C\}} = AB$

Supongamos ahora que  $\mathbb{F} = \{A \rightarrow B, B \rightarrow C, A \rightarrow CD\}$ , la dependencia funcional

$A \rightarrow CD$  no es redundante, pero  $A \rightarrow CD \not\models A \rightarrow D$  que no es redundante y  $A \rightarrow CD \not\models C$  que sí es redundante. Se dice entonces que  $C$  es un atributo extraño a la derecha.

Una dependencia funcional  $X \rightarrow Y$  se dice simple si  $Y$  es atributo simple. Todas aquellas dependencias funcionales que no sean simples se pueden descomponer en dependencias simples (por axioma de proyectividad). Así, por ejemplo, si tenemos  $X \rightarrow A_1 A_2 A_3$  lo descompondremos en  $X \rightarrow A_1$ ,  $X \rightarrow A_2$  y  $X \rightarrow A_3$ .

Observemos ahora  $\mathbb{F} = \{A \rightarrow C, BC \rightarrow D, B \rightarrow C\}$ , en donde  $\mathbb{F}$  no redundante pero aquí sus atributos son simples y ocurre que  $B \rightarrow C$  y  $BC \rightarrow D \not\models B \rightarrow D$  (axioma pseudo-transitividad) y  $B \rightarrow D \not\models BC \rightarrow D$  (pero no  $BC \rightarrow D \not\models B \rightarrow D$ ). La dependencia  $B \rightarrow D$

aporta más información que  $BC \rightarrow D$ . Se dice que  $C$  es un atributo extraño a la izquierda. Nos interesa más trabajar con  $\mathbb{F}' = \{A \rightarrow C, B \rightarrow D, B \rightarrow C\}$

Sea  $R(\mathcal{R})$  y  $\mathbb{F}$  conjunto de DFs sobre  $\mathcal{R}$ . Sea  $X \rightarrow Y \in \mathbb{F}$  con  $A \in C$  ( $X$  es atributo compuesto). Se dice que  $A$  es extraño a la dependencia funcional  $X \rightarrow Y$  respecto de  $\mathbb{F}$  si  $\mathbb{F} - \{X \rightarrow Y\} \cup \{X - A \rightarrow Y\} \equiv \mathbb{F}$ .

Consideremos  $BC \rightarrow D$ . Queremos ver si  $C$  es extraño en  $BC \rightarrow D$ . Esto sucedería si  $\mathbb{F}' = \{A \rightarrow C, B \rightarrow D, B \rightarrow C\} \equiv \mathbb{F}$ . Siempre se cumple que  $\mathbb{F}' \equiv \mathbb{F}$  por el axioma de aumentación, pero lo que puede que no sea cierto es que  $\mathbb{F} \not\equiv \mathbb{F}'$ . Eso último sucederá cuando  $\mathbb{F} \not\vdash B \rightarrow D$ , pero eso es equivalente a que  $D \in B^+_{\mathbb{F}}$ . Por consiguiente,  $C$  será extraño en  $BC \rightarrow D$  respecto de  $\mathbb{F}$  si  $D \in B^+_{\mathbb{F}}$ .

### Cobertura Canónica

Sea  $R(\mathcal{R})$  y  $\mathbb{F}$  conjunto de DFs sobre  $\mathcal{R}$ . Se llama cobertura canónica de  $\mathbb{F}$  a un conjunto  $\mathbb{F}_c$  tal que  $\mathbb{F}_c$  es no redundante, todas sus dependencias funcionales son simples y no incluye atributos extraños a la izquierda. La cobertura canónica contiene la esencia de la información que maneja  $\mathbb{F}$ .

Método para conseguir una  $\mathbb{F}_c$ :

- 1- desdoblamiento de las dependencias funcionales: si encontramos una dependencia funcional cuyo consecuente sea un atributo doble se transforma en Dfs con consecuentes simples.
- 2- eliminación de atributos extraños a la izquierda.
- 3- eliminación de redundancias.
- 4- agrupamiento de dependencias funcionales

Ejemplo:

Sea  $R(A,B,C,D,E,F,G,H,I,J)$  y

$\mathbb{F} = \{A \rightarrow B, A \rightarrow C, B \rightarrow CD, D \rightarrow B, ABE \rightarrow F, E \rightarrow J, EG \rightarrow H, H \rightarrow G\}$

Conseguir  $\mathbb{F}_c$ :

1) Desdoblamiento. Resulta

$A \rightarrow B, A \rightarrow C, B \rightarrow C, B \rightarrow D, D \rightarrow B, ABE \rightarrow F, E \rightarrow J, EG \rightarrow H, H \rightarrow G\}$

2) Eliminación atributos extraños a la izquierda.

$ABE \rightarrow F$

A extraño?  $(BE)^+_r = BECD$  Como A no pertenece  $\Rightarrow$  A no es extraño

B extraño?  $(AE)^+_r = AEBCD\dots$  B pertenece  $\Rightarrow$  B es extraño

Quitamos  $ABE \rightarrow F$  y ponemos  $AE \rightarrow F$

$AE \rightarrow F$

¿A extraño?  $(A)^+_r = ABCD$  Como F no pertenece  $\Rightarrow$  A no es extraño

¿E extraño?  $(E)^+_r = EI$  F no pertenece  $\Rightarrow$  E no es extraño

$EG \rightarrow H$

¿E extraño?  $(E)^+_r = EJ$  Como H no pertenece  $\Rightarrow$  E no es extraño

¿G extraño?  $(G)^+_r = G$  H no pertenece  $\Rightarrow$  G no es extraño

3) eliminación redundancias

$A \rightarrow B$ ?  $A^+_{r\{A \rightarrow B\}} = AC \rightarrow$  no es red.

$B \rightarrow C$ ?  $B^+_{r\{B \rightarrow C\}} = BD \rightarrow$  no es red.

$D \rightarrow B$ ?  $D^+_{r\{D \rightarrow B\}} = D \rightarrow$  no es red.

$E \rightarrow J$  y  $H \rightarrow G$  no son redundantes

$A \rightarrow C$ ?  $A^+_{r\{A \rightarrow C\}} = ABCD \rightarrow$  es redundante. Eliminamos  $A \rightarrow C$

$B \rightarrow D$ ?  $B^+_{\mathcal{F}-\{B \rightarrow D\}} = BC \rightarrow$  no es red.

$AE \rightarrow F$ ?  $B^+_{\mathcal{F}-\{B \rightarrow D\}} = BC \rightarrow$  no es red.

$EG \rightarrow H$ ? no es redundante (no hay ningún consecuente H)

Resulta  $\mathcal{F} = \{ A \rightarrow B, B \rightarrow C, B \rightarrow D, D \rightarrow B, AE \rightarrow F, E \rightarrow J, EG \rightarrow H, H \rightarrow G \}$

4) Agrupamientos de Dfs:

$\mathcal{F}_C = \{ A \rightarrow B, B \rightarrow CD, D \rightarrow B, AE \rightarrow F, E \rightarrow J, EG \rightarrow H, H \rightarrow G \}$

## Relación Entre Dependencias Funcionales Y Claves

Sea  $R(\mathcal{R})$  y  $\mathcal{F}$  un conjunto de Dfs asociado a  $\mathcal{R}$ , y  $X \subset \mathcal{R}$ . Si encontramos un atributo tal que  $\mathcal{F} \models X \rightarrow \mathcal{R}$  ( $X^+_{\mathcal{F}} = \mathcal{R}$ ) entonces  $X$  es una superclave.

Las dependencias funcionales permiten automatizar el proceso de búsqueda de superclaves.

Una clave candidata es aquel atributo  $X \subset \mathcal{R}$  tal que no existe  $X' \subset X$  tal que  $(X')^+_{\mathcal{F}} = \mathcal{R}$ , o sea que  $X \rightarrow \mathcal{R}$  no contenga atributos extraños a la izquierda.

Sea  $R(\mathcal{R})$  y  $\mathcal{F}_C$ , veremos un método que nos permita obtener las claves candidatas de  $R$ :

1- obtener  $K \subset \mathcal{R}$  que contenga aquellos atributos de  $\mathcal{R}$  que no aparecen como consecuente en  $\mathcal{F}_C$ . Toda clave candidata es un superconjunto de  $K$ .

2- Si  $K^+_{\mathcal{F}_C} = \mathcal{R} \Rightarrow K$  es la única clave candidata

3- Si  $K^+_{\mathcal{F}_C} \neq \mathcal{R} \Rightarrow$  las claves candidatas de  $R$  serán de la forma  $KW$ , donde  $W$  es un atributo simple o compuesto de  $\mathcal{R}$  que no está en  $\mathcal{R} - K^+_{\mathcal{F}_C}$  (con  $W \subset \mathcal{R} - K^+_{\mathcal{F}_C}$ ).

### 3.1 Descripción del método

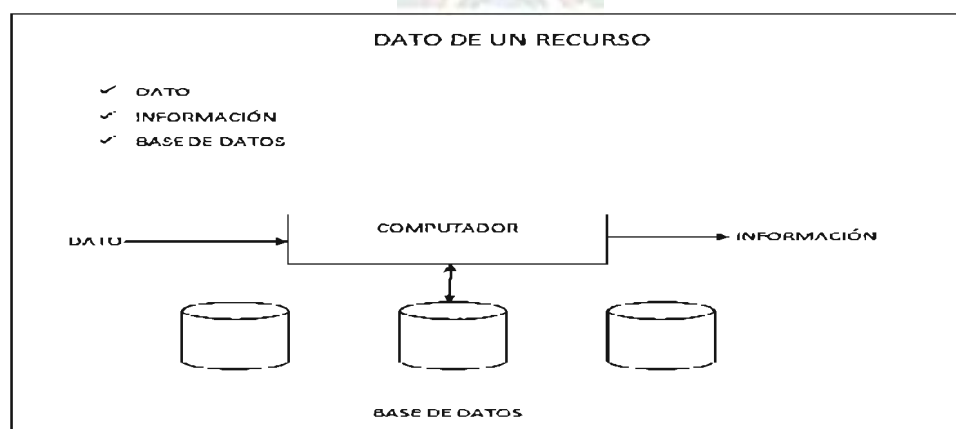
En esta fase de investigación se plantean las consideraciones que son base para realizar la demostración de la hipótesis planteada partiendo de un análisis de involucrados y para luego llegar a un diseño de la base de datos sin utilizar la normalización. Esta labor nos llevara a concluir en una identificación de las principales variables que intervienen para que finalmente sea planteada esta metodología EAS-J para la verificación de la tesis planteada.

Para la realización del presente trabajo se utilizara el método propio

### 3.2 Herramientas de especificación

Durante el proceso de la información, se han de modelar tanto los datos empleados por el sistema como los procesos que realizan tareas sobre esos datos:

**Figura 3.1 Recurso de la información**



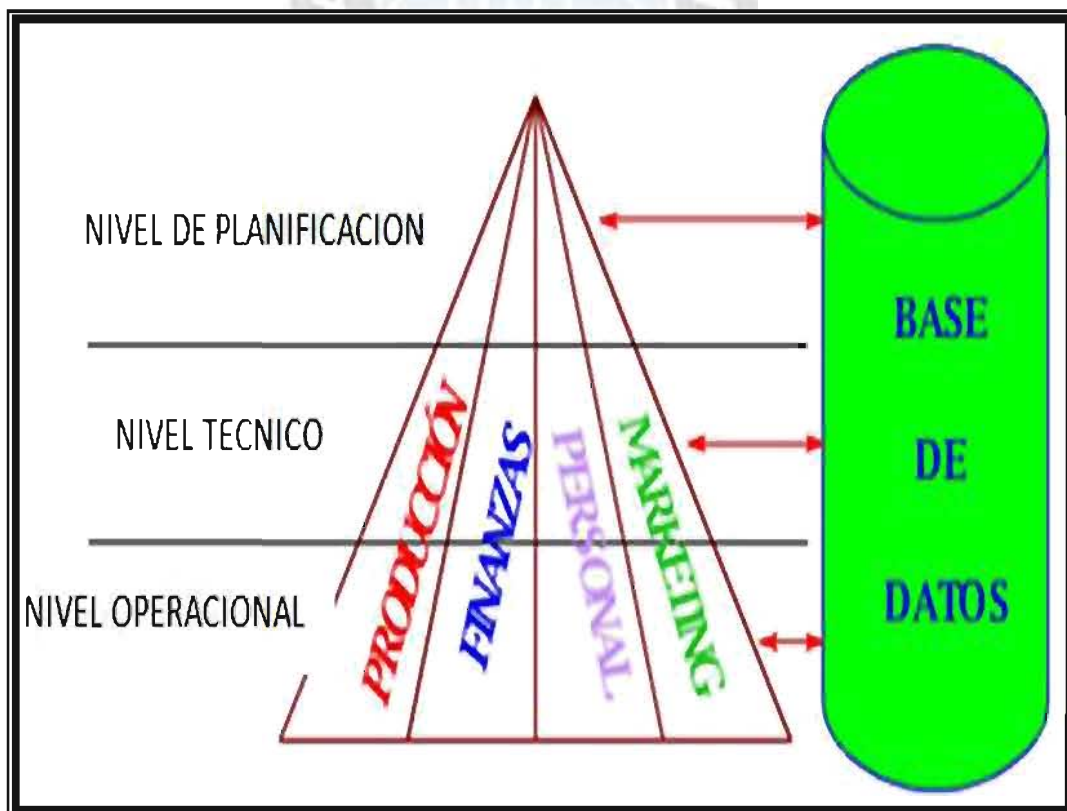
Fuente[[www.hipertexto.info/documentos/oohdm.htm](http://www.hipertexto.info/documentos/oohdm.htm) ]

### 3.2.1 Metodología del desarrollo

- ✓ Modelado de datos
- ✓ Representación gráfica del modelo de datos
- ✓ Modelado de procesos
- ✓ Diagramas de flujo de datos

Técnica de análisis basada en la identificación de las entidades y de las relaciones que se dan en una parte de realidad que pretendemos modelar. Existen notaciones alternativas para la representación grafica del diseño conseguido mediante la técnica de análisis que propone el modelo E/R:

Figura 3.2 Niveles de una Base de Datos



Fuente [Rules of Data Normalization / [www.datamodel.org/NormalizationRules.html](http://www.datamodel.org/NormalizationRules.html))]



### **3.3 Arquitecturas de sistemas de bases de datos**

La evolución de la tecnología y los sistemas de bases de datos. Al comienzo del proceso de datos, durante los cincuenta y el comienzo de los sesenta, la regla era el tratamiento de archivos secuenciales. Todos los datos se almacenaban en archivos secuenciales, que exigían el tratamiento de archivos completos por los programas de aplicación. Durante los sesenta, debido a que el almacenamiento en disco utilizando el acceso directo llegó a estar ampliamente disponible, el procesamiento de archivos de acceso aleatorio llegó a ser factible y popular.

Este método permitió el acceso directo a datos específicos en un archivo en la medida en que los sistemas computacionales de procesamiento de datos se hicieron más importantes, las empresas comenzaron a reconocer que la información era un recurso corporativo de valor considerable. Estas percibieron más y más que los datos necesarios para contestar numerosas preguntas estaban disponibles en sus archivos de procesamiento de datos. Como consecuencia, comenzaron a presionar a los sistemas de información para la gestión en cuanto a la utilización de la potencia del computador para producir información a partir de los datos corporativos. Esto inició la demanda de los sistemas de bases de datos, los que garantizarían más efectivamente el acceso a los datos y su manipulación.

A mediados de los sesenta se introdujeron los primeros sistemas de bases de datos, cuyo fundamento era una estructura jerárquica de los datos estos sistemas permitieron la recuperación de múltiples registros asociados con un registro único de otro archivo. Inmediatamente después, se desarrollaron los sistemas de base de datos en redes que soportaron interrelaciones entre registros de archivos diferentes mucho más complejas. Ambos modelos de base de datos, el jerárquico y el de red, requirieron el uso de punteros físicos predefinidos para enlazar los registros relacionados.

En 1970, Codd revolucionó el pensamiento en la industria de las bases de datos, el enfoque de Codd proponía el acceso y la manipulación de los datos únicamente desde el punto de vista de sus características lógicas. Durante los años setenta y

ochenta se desarrollaron numerosos sistemas de bases de datos relacionales y, en la actualidad, éstos dominan el mercado comercial.

En años recientes han proliferado los computadores personales en los puestos de trabajo, por lo que se han desarrollado las redes de computadores, permitiendo a los usuarios de estos computadores compartir recursos. Un computador, que funciona como servidor de una red, garantiza el acceso a la base de datos desde las estaciones de trabajo en estos puestos, permitiendo una división poderosa y eficiente de la tarea: El servidor recupera los datos, los que la máquina cliente solicitante procesa y presenta en pantalla para su manipulación por parte del usuario final. Las redes de computadores en ambiente cliente/servidor han desarrollado un grado alto de sofisticación y se encuentran cada vez con más frecuencia en las empresas comerciales.

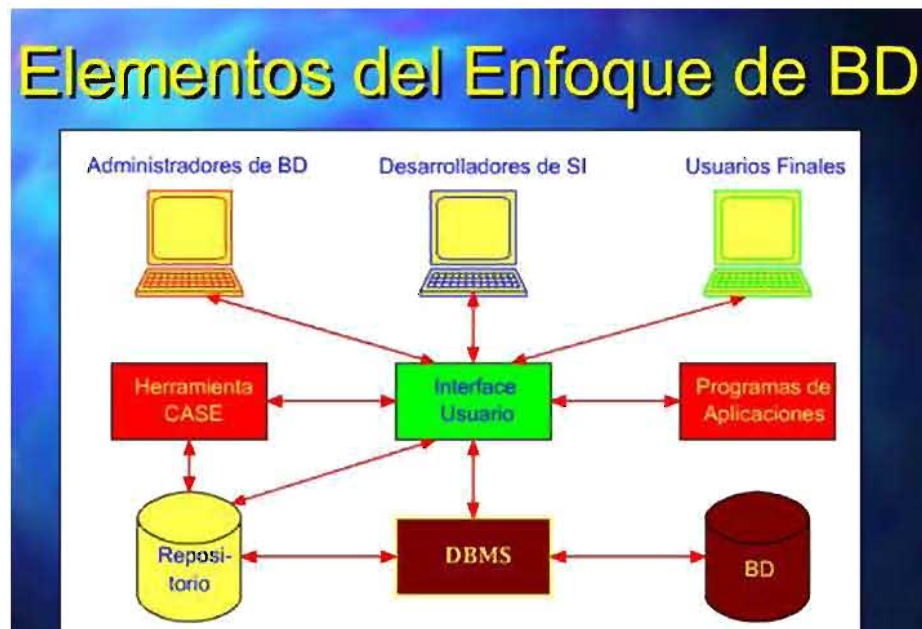
Desde el punto de vista conceptual, un sistema de base de datos en una organización grande está formado por el hardware, el software, los datos y las personas. La configuración del hardware comprende uno o más computadores, unidades de disco, terminales, impresoras, unidades de cinta magnética, conexiones de red y otros dispositivos físicos. El software incluye un sistema de gestión de bases de datos

(SGBD) y los programas de aplicación utilizan el SGBD para tener acceso y manipular la base de datos. Los datos, que representan los hechos importantes para la organización, radican físicamente en el disco, pero se estructuran lógicamente de forma que se logre un acceso fácil y eficiente.

### **3.4 El Estándar ansi/sparc.**

El objetivo principal de la arquitectura ANSI/SPARC es definir un SGBD con el máximo grado de independencia, separando las aplicaciones de usuario y la base de datos física. Para ello se utilizan tres niveles de abstracción conocidos como interno, conceptual y externo.

Figura 3.3 Elementos del enfoque de Base de Datos



Fuente [Database Basics (<http://databases.about.com>)]

### 3.5 Correspondencias, asociaciones o ligaduras (mappings)

Para describir un mismo grupo de datos, un sistema puede gestionar varios niveles de esquemas, para lo cual el DBMS debe poder garantizar la transferencia de los datos desde el formato correspondiente de un nivel al formato correspondiente a otro nivel; este proceso se denomina transformación de datos o mapping.

### 3.6 Modelos de bases de datos

Anteriormente se ha analizado con cierto detalle la arquitectura ANSI/SPARC para sistemas de bases de datos. En esta sección vamos a examinar los sistemas de bases de datos desde un punto de vista diferente.

La arquitectura de un sistema de base de datos está influenciada en gran medida por el sistema informático subyacente en el que se ejecuta el sistema de base de datos. En la arquitectura de un sistema de base de datos se reflejan aspectos como la conexión en red, el paralelismo y la distribución:

- ✓ La arquitectura centralizada es la más clásica. En ella, el SGBD está implantado en una sola plataforma u ordenador desde donde se gestiona directamente, de modo centralizado, la totalidad de los recursos. Es la arquitectura de los centros de proceso de datos tradicionales. Se basa en tecnologías sencillas, muy experimentadas y de gran robustez.

### 3.7 Arquitectura centralizada

Los sistemas de bases de datos centralizados son aquellos que se ejecutan en un único sistema informático sin interaccionar con ninguna otra computadora. Tales sistemas comprenden el rango desde los sistemas de bases de datos monousuario ejecutándose en computadoras personales hasta los sistemas de bases de datos de alto rendimiento ejecutándose en grandes sistemas.

Una computadora moderna de propósito general consiste en una o unas pocas CPU's y un número determinado de controladores para los dispositivos que se encuentren conectados a través de un bus común, el cual proporciona acceso a la memoria compartida. Las CPU's poseen memorias caché locales donde se almacenan copias de ciertas partes de la memoria para acelerar el acceso a los datos. Cada controlador de dispositivo se encarga de un tipo específico de dispositivos (por ejemplo, una unidad de disco, una tarjeta de sonido o un monitor). La CPU y los controladores de dispositivo pueden ejecutarse concurrentemente, compitiendo así por el acceso a la memoria. La memoria caché reduce la disputa por el acceso a la memoria, ya que la CPU necesita acceder a la memoria compartida un número de veces menor.

Se distinguen dos formas de utilizar las computadoras: como *sistemas monousuario* o como *sistemas multiusuario*. En la primera categoría están las computadoras personales y las estaciones de trabajo. Un sistema monousuario típico es una unidad de sobremesa utilizada por una única persona que dispone de una sola CPU, de uno o dos discos fijos y que trabaja con un sistema operativo que sólo permite un único

Las bases de datos diseñadas para las máquinas monoprocesador ya disponen de multitarea, permitiendo que varios procesos se ejecuten a la vez en el mismo

procesador, usando tiempo compartido, mientras que de cara al usuario parece que los procesos se están ejecutando en paralelo. De esta manera, desde un punto de vista lógico, las máquinas paralelas de grano grueso parecen ser idénticas a las máquinas monoprocesador, y pueden adaptarse fácilmente los sistemas de bases de datos diseñados para máquinas de tiempo compartido para que puedan ejecutarse sobre máquinas paralelas.

### 3.8 Aplicación de la Metodología

#### 3.8.1 Metodología EAS-J

Está inspirado en el modelado ER y presenta un ciclo recursivo basado en los conceptos de:

Figura 3.4 Principios de la Metodología EAS-J



Fuente[Elaboración propia]

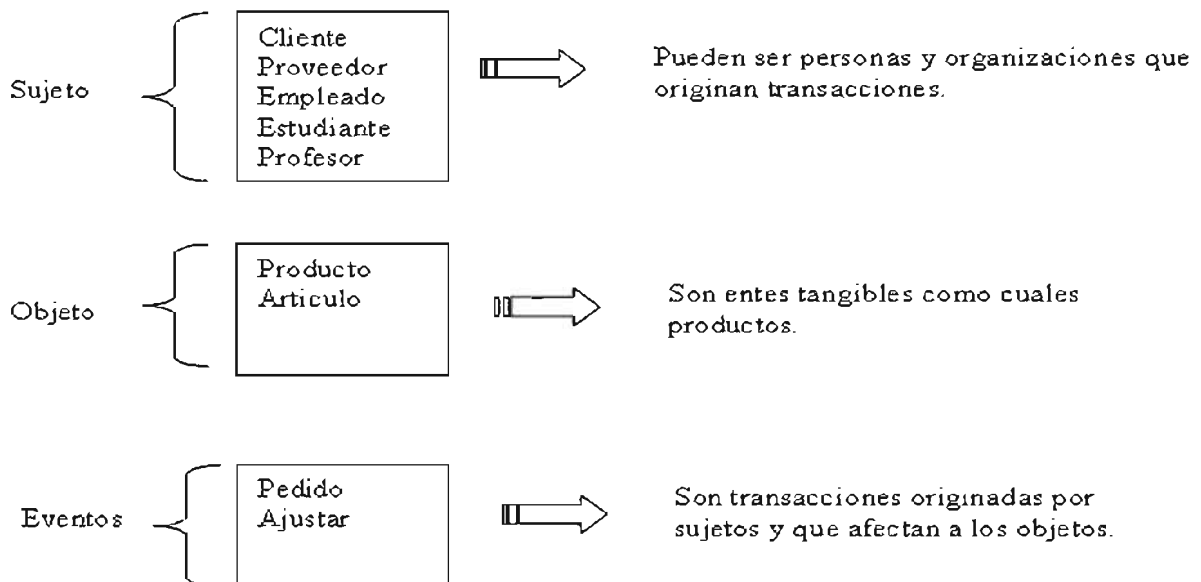
Entidad y semántica de atributos, generando de manera sencilla bases de datos altamente normalizadas.

### 3.8.2 Conceptos fundamentales

**3.8.2.1 Entidad.**- Es cualquier Objeto, Sujeto o Evento del cual se desea guardar información.



**Entidades:** se puede considerar entidades a los sujetos, objetos, a los eventos, a los lugares y a los abstracciones.



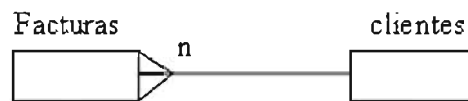


**3.8.2.2 Entidades Objeto** - Son entidades tangibles, visibles, que siempre llevan un atributo llave primaria.

**3.8.2.3 Entidades Sujeto** - Son entidades vivientes (personas, animales, plantas), que siempre llevan un atributo llave primaria.

**3.8.2.4 Entidades Evento** - Son entidades que demuestran ciertas actividades, opcionalmente pueden poseer llave primaria, siempre están compuesta por llaves foráneas (llaves primarias de las entidades objeto y/o sujeto), además de otros atributos complementarios.

**Relación:** es la asociación significativa y estable entre dos entidades



**Relaciones:** las relaciones tiene tres propiedades ò características:

➤ **Grado ò Cardinalidad:** que se clasifica en:

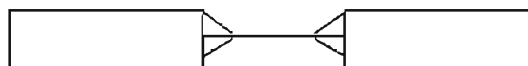
1 a 1



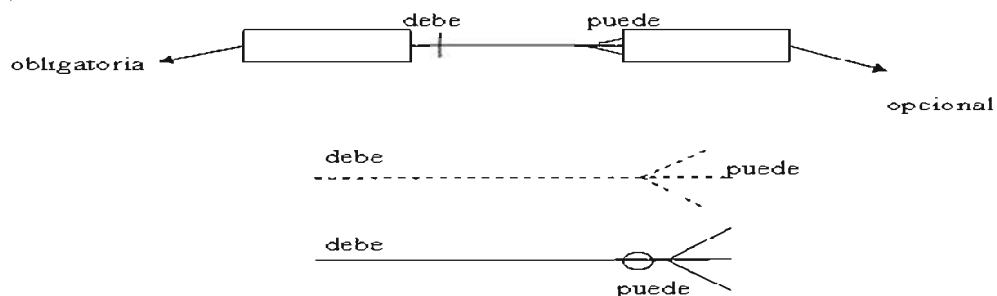
1 a N



N a N

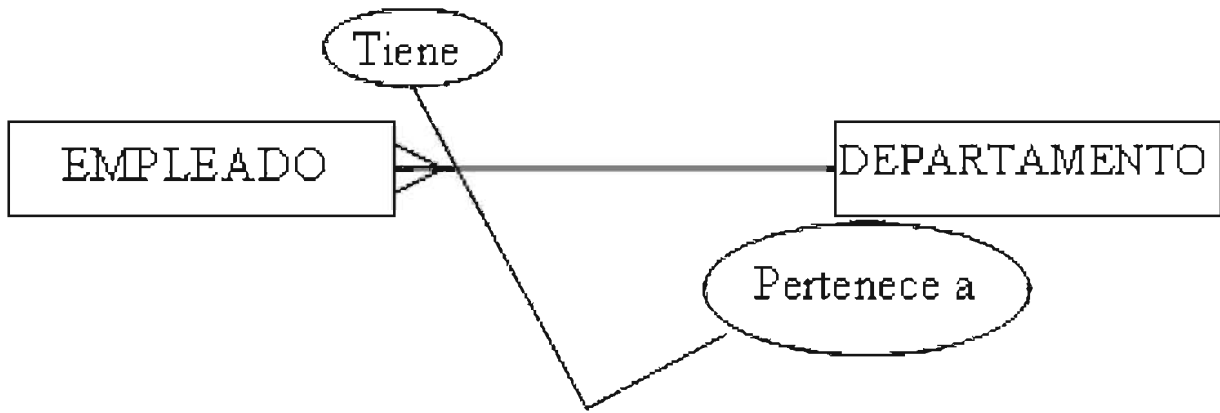


➤ **Opcionalidad:** es la participación obligatoria u opcional en la entidad de la relación.





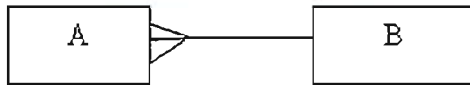
➤ *Leyenda:* es una expresión que escribe el rol de cada entidad en la relación.



### Leyendas

#### Como se lee el Grado ò Cardinalidad

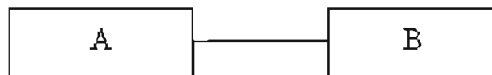
➤ *Uno a muchos:* una instancia de la entidad A se relaciona con una ò màs instancias de la entidad B.



➤ *Muchos a muchos:* una instancia de la entidad A se relaciona con una ò màs instancias de la entidad B y una instancia de la entidad B se relaciona con uno ò màs instancias de le entidad B.

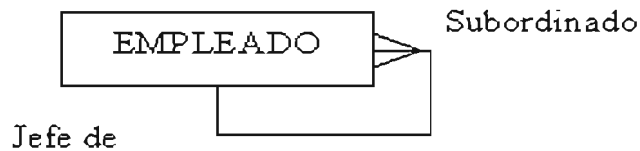


➤ *Uno a uno:* una instancia de la entidad A se relaciona con uno y sólo una instancia de la entidad B.



### 3.8.3 Relación Recursiva

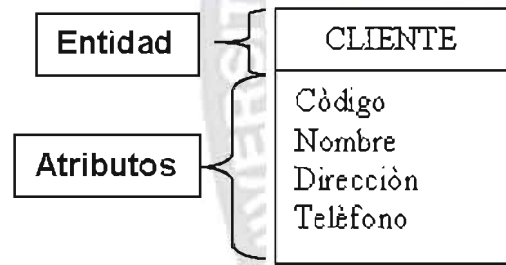
Una instancia de una entidad se asocia con instancia de si misma, es opcional en los dos extremos, es decir, no hay el carácter de obligatorio. Ej:



Un Empleado puede ser jefe de uno o más empleados y un Empleado puede ser subordinado de un y sólo un Empleado

**Atributo**.- Es una característica que describe de alguna manera a una entidad.

Entidad cliente(nombre, apellido, dirección, edad, sexo)



### 3.8.4 Tipos de atributos

En esta metodología, lo clasificamos en cinco tipos: llave primaria, llave foránea, multivalorado, candidato a patrón de búsqueda y descriptor puro.

- 1) **Atributo Llave Primaria**.- Es aquel campo que identifica de manera unívoca a alguna fila de datos de las entidades OBJETO y SUJETO.
- 2) **Atributo Llave Foránea**.- Es aquel campo que en otra tabla es llave primaria. Son frecuentemente utilizados en las entidades EVENTO aunque no exclusivamente.

- 3) **Atributo Multivalorado.**- Es aquel atributo que en algún momento puede presentar más de un caso, esto obligaría a repetir todos los demás atributos en la misma tabla por cada caso, produciendo gran redundancia.
- 4) **Atributo Candidato A Patron De Búsqueda.**- Es aquel atributo que en una aplicación puede ser patrón de búsqueda durante las consultas, esto produciría redundancia del atributo y riesgo de inconsistencia producida por los procesos conocidos como: Altas, Bajas y Cambios.
- 5) **Atributo Descriptor Puro.**- Es aquel atributo que identifica de manera estricta y precisa a una entidad y no se asemeja a ninguno de los cuatro tipos anteriores.

### **3.8.5 Semántica De Atributos**

Por su semántica podemos clasificarlos en: atributos estáticos y atributos desencadenadores.

#### **1) Atributos Estaticos**

Son aquellos que se mantienen en sus tablas y no generan nuevas tablas, entre ellos tenemos:

- a) Llave primaria
- b) Llave extranjera
- c) Descriptor puro.

#### **2) Atributos Desencadenadores**

Son aquellos que generan nuevas tablas, entre ellos tenemos: atributos multivalorados y atributos candidatos a patrones de búsqueda.

### **3) Semántica De Un Atributo Multivalorado**

En la tabla origen, el campo desaparece y genera una nueva tabla denominada: tabla-campo, cuyos atributos mínimamente son dos: Llave extranjera (llave primaria de la tabla origen) y Descripción del atributo “multivalorado”, adicionalmente se puede agregar otros atributos dependiendo del diseñador, a estos se debe aplicar nuevamente el mismo criterio

### **4) Semántica De Un Atributo Candidato A Patrón De Búsqueda**

En la tabla origen, el atributo se convierte en llave foránea (llave primaria de la nueva tabla), generándose una nueva tabla cuyo nombre será el nombre del atributo, formado por al menos dos campos: llave primaria y descripción, según los requerimientos se pueden agregar otros atributos, a los cuales se les debe aplicar el mismo criterio.

### **3.8.6 Etapas De La Metodología**

La metodología debe ser aplicada luego del análisis del entorno de trabajo, en principio se deben identificar todas las entidades organizándolos por tipo, de la siguiente manera:

- a) IDENTIFICACION DE ENTIDADES SUJETO Y SUS ATRIBUTOS
- b) IDENTIFICACION DE ENTIDADES OBJETO Y SUS ATRIBUTOS
- c) IDENTIFICACION DE ENTIDADES EVENTO Y SUS ATRIBUTOS

Es importante hacer notar que, a cada atributo se le debe aplicar la “semántica de atributos” en el siguiente orden:

1º) DESCARTAR QUE SEAN: “LLAVES PRIMARIAS”, “LLAVES FORÁNEAS” Y “DESCRIPTOR PURO”.

2º) VERIFICAR SI ES ATRIBUTO “MULTIVALORADO”, ENTONCES APLICAR SU SEMÁNTICA.

3º) VERIFICAR SI ES ATRIBUTO “CANDIDATO A PATRÓN DE BÚSQUEDA”.

Este proceso debe ser efectuado de manera recursiva a todos los atributos de las tablas que se vayan generando



Bienvenid@ - admin admin

**CREAR BASE DE DATOS**

**NUEVO**

**MODELAR DATOS**

Aquí podemos crear la base de datos

The image shows a web application interface for a database system. At the top, there is a banner with the text "SISTEMA AUTOMATIZADO DE BASE DE DATOS" in large, 3D-style letters. Below the banner, the user is logged in as "admin admin". The main area displays a table with the following data:

idLibro	Autor	Especialidad	Edicion	Nro_pagi	Editorial	Agregar nuevo campo
1200	Miguel de Cervantes	Novela	2011	320	Navarrete	

At the bottom of the interface, there are two blue buttons: "EDITAR" and "MOSTRAR".



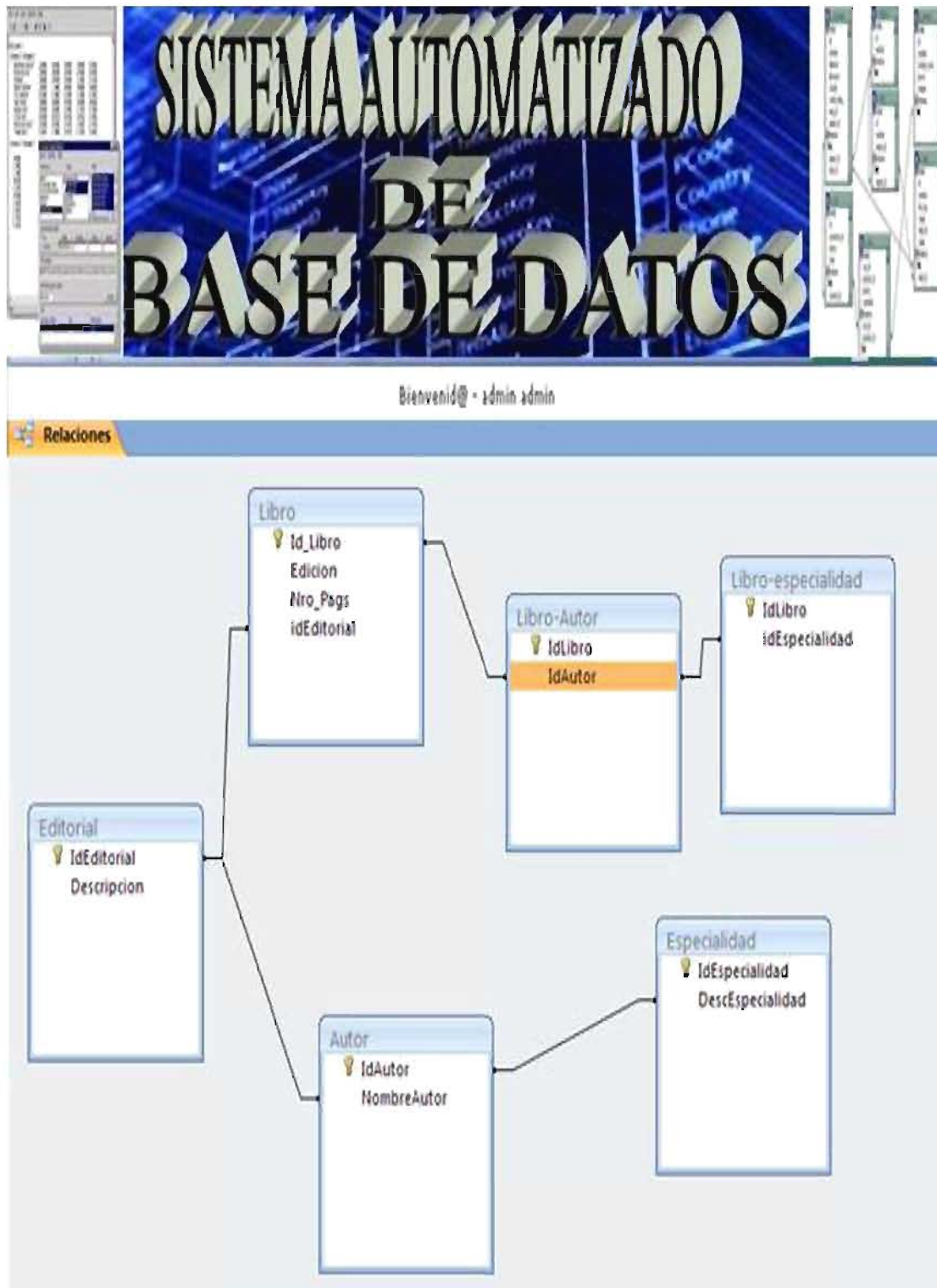
Se crea la clase



Modelamos los datos y nos generara el siguiente



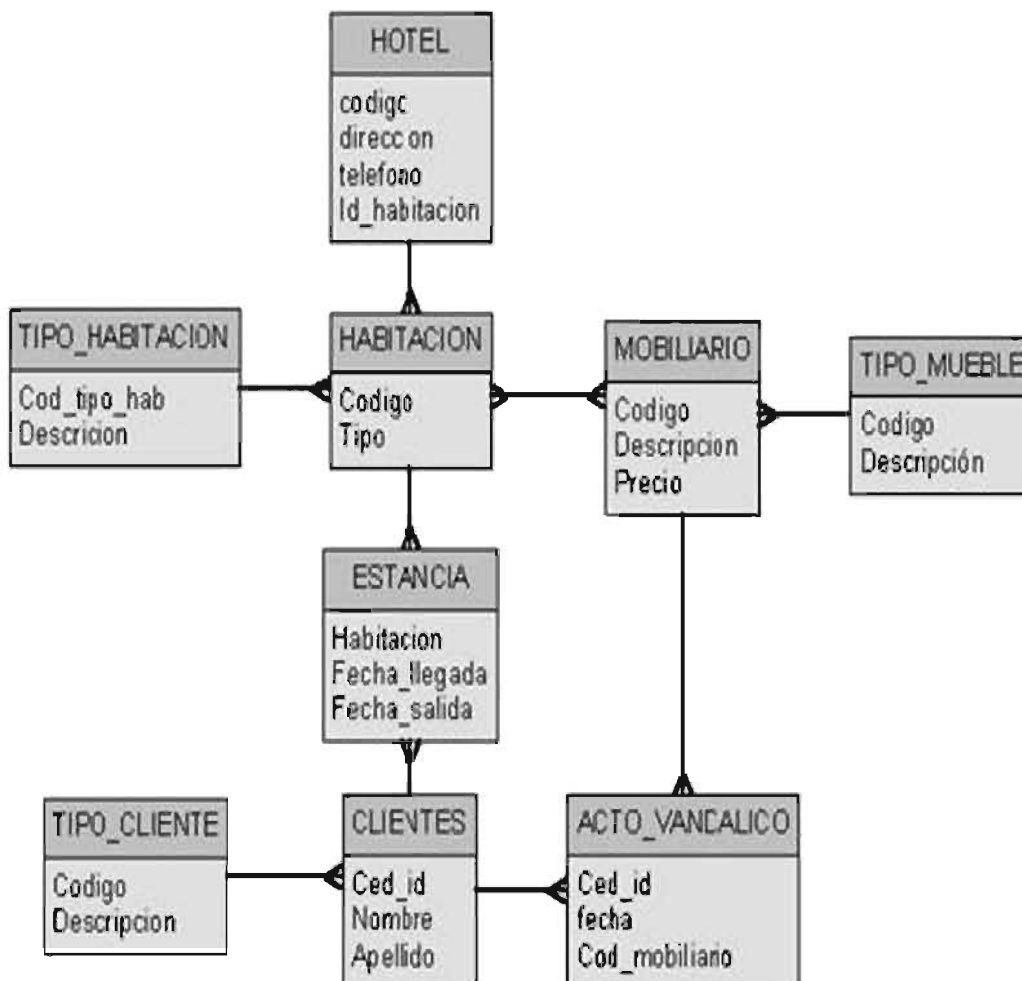
El resultado es



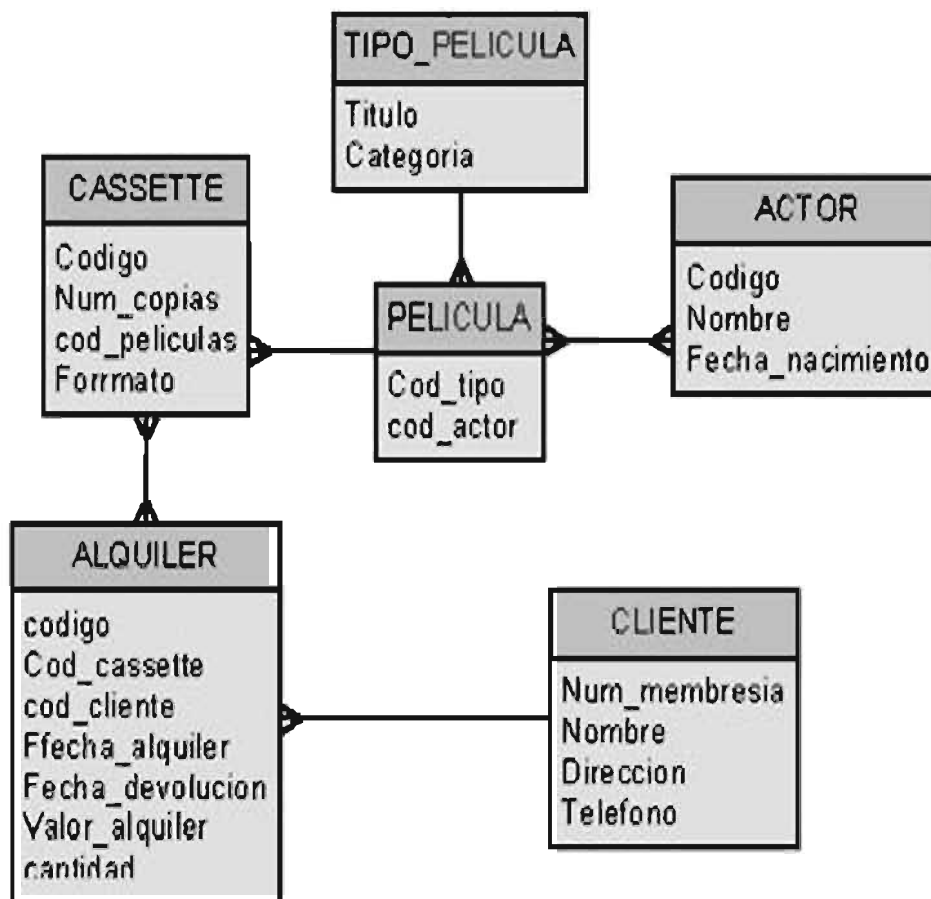
Otro ejemplo modelando datos y su relación



Bienvenido@ - admin admin



## Modelamos más Datos



## CAPITULO IV

### CONCLUSIONES Y RECOMENDACIONES

La aplicación de la metodología para automatizar una base de datos consigue una disminución de las anomalías de actualización y de las redundancias, evitando muchos de los problemas que se pueden plantear en las actualizaciones de la información

Sin embargo al mismo tiempo, penaliza las consultas, al disminuir la eficiencia de las mismas, las necesidades de información de los usuarios cambian constantemente, no es posible anticiparlas en su totalidad.

El modelo relacional de bases de datos con sus relaciones es una solución simple y elegante para satisfacer las más diversas condiciones de consulta y extracción de datos e información.

Se recomienda que la metodología EAS-J es aplicable en Base de Datos para dar a conocer a los docentes y estudiantes de la carrera de informática y los diseñadores de bases de datos para su aplicación de esta metodología.



## BIBLIOGRAFIA

[Ausubel, et al, 90] David Ausubel, Joseph Novak, Helen Hanesian: psicología educativa: un punto de vista cognoscitivo. México: Editorial Trillas, SA de CV, 2da, edición, 1990

[Arratia-Sarvia, 03] Arratia D. Saravia R. 2003 Tesis de Grado de Informática Tutores Inteligentes para mejorar el entendimiento de las matemáticas La Paz Bolivia Tesis de grado UMSA.

[Booch, 94] Booch, G. "Object Oriented Analysis and Design with Applications", the Benjamin Cummings Publishing Company, 1994.

[Brj, 97] Booch, G., Rumbaugh, J., Jacobson, I. Unified Modeling Language. Addison Wesley longman, Inc. Massachusetts, USA, 1997.

[Brooks,97] Brooks, R. A.; "A robust layered control system. For a mobile robot", IEEE Journal of Robotics and Automation, Vol. 2, N° 1, pp. 14-23, 1986.

[Buretta,97] Buretta, M.; "Data Replication hools und Techniques, for magazine Distributed information", Editorial Wiley, 19 97.

[Bruner 90] Bruner J. Actos de significado. Más allá de la revolución cognitiva. Alianza. Madrid. 2002.

[Caeiro, M. et al., 2003] Manuel Caeiro, Fernando Mikic, Luis Anido, Martín Llamas: Análisis de componentes para un Modelo de Descripción de Unidades de Aprendizaje Heterogéneas, En: Tercer Congreso Iberoamericano de Telemática-CITA 2003, Uruguay, octubre 2003.



[Stuart, 95] Stuart J.R. Norvig, Canny, J. Artificial Intelligent a Modern Approach. Prentice Hall, Englewood Cliffs, New Jersey 07632.

[Woold, 95] Wooldridge, M. and Jennings N. R.; "Agent theories, Architectures, and Languages: a Szrrvey", in Wooldridge and Jennings Eds., Intelligence Agents, Berlin: Springer-Verlag, Vol. 1, N° 22, 1995.

[Kim, 89] Kim, J. H. (1989). CIRCSIM-Tutor: An Intelligent Tutoring System for Circulatory Physiology. Ph.D. tesis, Illinois Institute of Technology.

[Van Lehn, 88] Student Modelling. M. Polson. Foundations of intelligent Tutoring System. Hillsdale. N. J. Lawrence Erlbaum Associates, 55-78

[Vernet, 05]. Vernet, F.M.T, Dunin-Keplicz B.M. Treur J. and Verbrugge, "lirfodelling Internal Dynamic Behczviour of 13 Agents". Proceedings of the Third International Workshop on Formal models of Notes Springer Verlag, 1997.

[Vigotsky, 78] vigotsky L. Mind in society. The development on higher psychological process Harvard University Press Cambridge M. A.

[Villareal, 03] Villareal Goulart R. R.; Giraffa Arquitectura de Sistemas tutores Inteligentes.

[Wang, 98] Wang K. y Liu B. Concurrent discretization of multiple attributes. Rim Internacional Conference on AI, pp. 250, 259.

[Wooldrigde, 99] Artificial Intelligent and tutoring systems 1999.

[Woo Woo, 91] Instructional planning in an Intelligent Tutoring System: Combining global lesson plan whit local discourse control. Degree of Doctor of Philosophy in computer Science in the Graduate School of the Illinois Institute of Technology. Chicago, Illinois December, 1991.