

Universidad Autónoma del Estado de México

“Facultad de Contaduría y Administración”

Licenciatura en Informática Administrativa

Administración de Base de Datos

Apuntes de Administración de Base de Datos

Autor:

ROCIO PALMA LOPEZ

Coautor:

María de La Luz Sánchez Paz

Índice

INTRODUCCIÓN.....3

Unidad I CONCEPTOS FUNDAMENTALES DE BASES DE DATOS

- **OBJETIVO:** Conocer los conceptos fundamentales de base de datos y de modelos de bases de datos como antecedentes y definiciones implementaría en las bases de datos.

Antecedentes de Bases de Datos.....	7
Tabla cronológica de los antecedentes de las Bases de Datos.....	7
Diferencia entre Archivos y Bases de Datos: ARCHIVO EXCEL.....	7
Base De Datos Access.....	7
Donde Utilizamos las Bases de Datos en la Actualidad.....	7
Conceptos.....	7
Administrador de Bases de Datos.....	8
Funciones del Administrador de Bases de Datos.....	8
Manejador de Bases de Datos.....	10
Características de la Base de Datos	11
Diferencia entre Base de Datos Transaccional y Datawarehouse.....	11
Ventajas.....	11
Desventajas.....	12
Conceptos de Bases de Datos.....	14
Modelo Conceptual.....	16
Modelos de Ejecución.....	17
Jerárquico.....	17
Red.....	17
Relacional.....	18
Orientado a Objetos.....	19

Unidad 2: DISEÑO DE BASE DE DATOS

OBJETIVO: Diseño de bases de datos conociendo niveles de abstracción y tomando como base el diseño conceptual, principios de normalización.

Arquitectura: Abstracción de Datos.....	20
Otra Opción de Niveles de Abstracción.....	20
Metodología de Base de Datos.....	21
Acoplamiento.....	22
Pasos que sigue una Metodología de Cascada (clásico)	22
Tipos De Datos.....	23
Numéricos.....	23
Lógicos.....	23
Carácter.....	23

Unidad 3: MODELO ENTIDAD-RELACION

- Analizar las características y elementos del modelo entidad relación para realizar el diseño de base de datos.

Modelo Entidad-Relación.....	25
Entidad.....	25
Fuerte.....	25
Débil.....	25
Atributos.....	26
Simple.....	26
Compuesto.....	26
Diseño Modelo Conceptual.....	27
Claves.....	28
Primaria.....	28
Candidata.....	28
Compuesta.....	28
Cardinalidad.....	29
Fuerza de las Entidades.....	30
Grado de Relación.....	30
Normalización.....	32
Reglas De Normalización.....	34
Ejemplo de Normalización.....	36



Unidad 4: DESARROLLO DE BASES DE DATOS

Desarrollar bases de datos a través de los diversos modelos de bases de datos

Uso de SGBD para desarrollar de Base de Datos.....	38
Elaborar diccionario de datos, llaves, relaciones en un gestor de base de datos....	39
Implementación del modelo E-R en un gestor de bases de datos relacional.....	40

Unidad 5: ADMINISTRACION DE BASE DE DATOS

Administración de base datos, tanto de los elementos como los datos

Lenguaje estructurado de consultas empleando el modelo relacional.....	41
Herramientas de administración de base de datos.....	42
Procesamiento de transacciones.....	43
Herramientas de explotación y administración de la información en la base de datos.....	44

Introducción

La administración de bases de datos forma parte de la informática, y se ha transformado con el tiempo tomando como base diversos sistemas electrónicos.

El contenido entero de una base de datos relacional se representa por una y sola una forma, a saber como valores de atributos dentro de relaciones.

Edgar Frank Codd definió las bases del modelo relacional a finales de los 60's. Trabajaba para IBM empresa que tardó un poco en implementar sus bases. Pocos años después el modelo se empezó a implementar cada vez más, hasta ser el modelo de bases de datos más popular.

Este modelo popular ha traído al mundo informático la sustitución de archivos de información, hoy en día lo conocemos como Base de Datos que ha venido con el paso del tiempo a facilitar la vida del ser humano.

El presente trabajo desarrolla íntegramente los contenidos del programa vigente de la asignatura Administración de Base de Datos recogiendo el propósito esencial de la asignatura, que consiste en tener un conocimiento amplio acerca de lo que es una Base de Datos enfatizando algunos de los principales criterios en la creación de la propia Base de Datos.

En las bases de datos se plantean problemas de seguridad como la comparación de datos, acceso a estos, protección contra fallos, contra accesos no permitidos, temas que también se mencionará al final de este trabajo.

UNIDAD I

Conceptos Fundamentales de Bases de Datos

- **OBJETIVO:** Conocer los conceptos fundamentales de base de datos y de modelos de bases de datos como antecedentes y definiciones implementarias en las bases de datos.
1. Antecedentes de las bases de datos
 2. Conceptos básicos de Bases de Datos
 3. Características de los sistemas de bases de datos
 4. Definición de administrador de base de datos y responsabilidades
 5. Modelos de base de datos y sus características como antecedentes
 6. Jerárquico
 7. Red
 8. E-R
 9. Relacional

Antecedentes de Bases de Datos.

- Surge la necesidad de guardar información.
- Antes guardaban la información en Archiveros.
- Después guardaban la información en la Computadora en Archivos Planos (.txt), pero era muy tardado para manejar la información y consultarla.
- A finales de los 70's comienzan las bases de datos que se basaban en Modelos Matemáticos.
- Así se comenzó la Administración de Bases de Datos.

Tabla cronológica de los antecedentes de las Bases de Datos

Década	Características de los archivos	Dispositivo de almacenamiento
50	Secuenciales	Tarjetas perforadas
60	Modo secuencial simple	Cinta magnética
65-70	Modo secuencial y acceso directo	Tambor magnético
70-80	Primeras Bases de Datos direccionales	Disco magnético compartido
80-90	Auge de las Bases de datos	Disco magnético compartido
90	Bases de Datos en red, compartidas o globalizadas	Informix, Fox, SQL, Access, Oracle.

1

¹ David M. Kroenke,(1996).Procesamiento de Bases de Datos, Fundamentos, Diseño e instrumentación (5ta ed.). Mexico: Prentice Hall

Conceptos Básicos de Bases de Datos

Diferencia entre Archivos y Bases de Datos: ARCHIVO EXCEL

- Almacena grandes cantidades de Información
- Es mas lento consultar la información
- No permite una fácil manipulación de los datos.

Base De Datos Access

- Tiene limitantes en cuanto a capacidad
- Es más rápido consultar información
- Permite mas fácil la manipulación de datos

Donde Utilizamos las Bases de Datos en la Actualidad

- ❖ *Bancos.*-registro de clientes
- ❖ *Aerolíneas.*-reservaciones de vuelos
- ❖ *Universidades.*-control Sobre datos de los alumnos
- ❖ *Telecomunicaciones.*-cuentas de los clientes
- ❖ *Producción.*-Inventarios de Materiales
- ❖ *Gobierno.*-estadísticas del INEGI
- ❖ *Hospitales.*-Control de los Pacientes
- ❖ *Finanzas.*-transacciones de Dinero
- ❖ *Ventas.*-Registro de datos de los productos actualizada

Base de Datos: Colección de datos interrelacionados y almacenados en un conjunto sin redundancias y que pueden ser utilizados en cualquier aplicación.(Fundamentos de Datos de Aldo Patini. AlfaOmega)

Una base de datos o banco de datos (en inglés: database) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.²

BASE DE DATOS

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por el sistema de información de una empresa o negocio en particular.

Consiste en una colección de datos interrelacionados que proporcionan un entorno que sea tan práctico como eficiente de usar en la recuperación y el almacenamiento de la información.

COMPONENTES DE LA BASE DE DATOS

- ✓ Los principales componentes de una base de datos son las **tablas o entidades**
- ✓ Una **tabla o entidad** está formada por un conjunto de registros
- ✓ Cada **registro** está formado por campos
- ✓ Cada **campo** es el espacio designado para guardar información y está compuesto por caracteres o bytes
- ✓ Cada carácter consta de 8 bits.

- ✓ La información de las tablas se pueden organizar en secuenciales y directos
- ✓ En las secuenciales los registros están almacenados en secuencia que depende de algún criterio definido.

² Conceptosbasicos.(n.d.).http://es.wikipedia.org/wiki/Diagrama_entidad-relaci%C3%B3n#Claves

- ✓ Los directos permiten acceder directamente un registro de información sin tener que buscar uno por uno. Usan la llave de acceso.
- ✓ Las bases de datos proporcionan la estructura requerida por los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos de las empresas, los cuales explotan la información contenida en las Bases de Datos.
- ✓ Por lo tanto es importante conocer la estructura y el manejo de las bases de datos ya que su uso adecuado proporciona grandes ventajas a la organización.

- Entidades: son unidades básicas usadas para modelar clases de objetos *concretos o abstractos*. Ejemplo: edificio, cuarto, silla, vacaciones, préstamos, conceptos, etc.

- Conjunto de Entidades: grupo de objetos similares de importancia para una organización por lo cual mantiene sus datos. Ejemplos: transacciones, conceptos, posiciones de trabajo, cursos, empleados, inventarios de productos brutos o terminados, estudiantes, clientes, etc.

- Atributos: son las propiedades directas y atómicas que caracterizan un conjunto de entidades. Se les conoce como elementos o campos. Ejemplo: para la entidad Empleado: No. de Seguro, Nombre, Dirección.
 - Simples y compuestos
 - Ejemplo: Nombre cliente, se puede dividir en nombre, 1er apellido, 2do apellido
 - Univalorados y multivalorados
 - Ejemplo: No. Préstamo, es único..... Un empleado puede tener varios Subordinados
 - Nulos
 - Ejemplo: Si el atributo No_Imss no está asignado se considera perdido o nulo
 - Derivados

- Ejemplo: Para obtener antigüedad de un empleado, contando sólo con la fecha de ingreso
- Dominio: se le llama así a el conjunto de posibles valores que puede tener un atributo. Ejemplo: para el No. de Seguro es un número de 9 dígitos positivo.
- Llave (identificadores): atributo único o combinación de dos o más atributos de un conjunto de entidades que se usan para identificar una o más instancias del conjunto
- Llave Primaria: es la llave que nos sirve para identificar de manera única a una instancia de un conjunto de entidades.
- Superllave: llave primaria aumenta con atributos adicionales y que aun identifican una instancia del conjunto de entidades.
- Llave Candidata: son aquellos atributos o conjunto de atributos que pueden funcionar como llave primaria. Se debe elegir una como primaria y las demás son llaves alternas.

Componentes Principales de un Sistema de Base de Datos

- ✓ Datos: Son la base de datos propiamente dicha, cadena de caracteres sin significado
- ✓ Hardware: Son los dispositivos de almacenamiento en donde reside la base de datos.
- ✓ Software: Conjunto de programas conocidos como sistema manejador de base de datos (DBMS)
- ✓ Usuarios: Tres categorías (programador de aplicaciones, usuario final y el administrador de la base de datos)

Ventajas de las Bases de Datos

- Globalización de la información

- Eliminar información redundante
- Eliminar información incongruente
- Permite compartir información
- Permite mantener la integridad de la información
- Independencia de datos
- Eliminar la dificultad en el acceso de los datos

Administrador de Bases de Datos

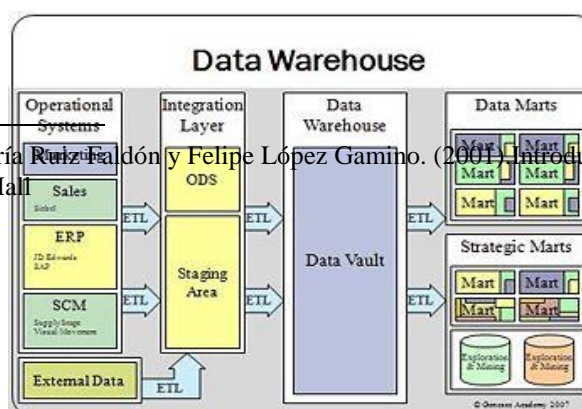
Persona que va a Administrar la Base de Datos. Controla y Administra un Gestor de Base de Datos.

Definir Rutinas de Conversión de los Datos:

Datawarehouse: es una Base de Datos histórica muy grande la tienen por ejemplo los bancos o supermercados, solo guardan la información mas importante.

El Administrador de Base de Datos trabaja con el gerente para saber que datos son relevantes para tomar futuras decisiones y guardarlos en el DAWAREHOUSE.

Datawarehouse: Es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.³



³ C. J. Date, Sergio L. María Ruiz-Faldón y Felipe López Gamino. (2001) Introducción a los sistemas de bases de datos (2da ed.): Prentice Hall

Estrategias de Trabajo:

- ❖ Diseño
- ❖ Homoclaves: Contraseñas de los usuarios de base de datos.

Manejador De Base De Datos O Sistema Gestor De Base De Datos

Software que tiene las funciones y tareas necesarias para administrar la Base de Datos: EJEMPLO: Oracle, DB2, SQL Server, Progress, MySQL, Informix, Visual Fox Pro.⁴



⁴ Basesdedatos.(n.d.).<http://www3.uji.es/~mmarques/f47/apun/node83.html>

Tareas del Sistema Gestor de Base de Datos

- Permite ingresar datos al Sistema
- Almacena datos y permite trabajar con ellos
- Provee de las herramientas para capturar y manipular datos.
- Debe permitir seguridad
- Permite crear informes y Reportes
-

Características de los sistemas de bases de datos

- ❖ Debe presentar un Modelo exacto de los Datos
- ❖ Organizar los datos con sencillez
- ❖ Permite reducir costos
- ❖ Minimizar las redundancias (no debe existir información duplicada)
- ❖ Integridad referencial (se refiere a tener la información bien organizada cada usuario con sus datos, no revolver la información, no modificar la información.
- ❖ Garantiza la Seguridad y Privacidad (los Administradores otorgan contraseñas.
- ❖ Amplía la capacidad de búsqueda (todos los datos que se introdujeron se deben poder consultar)
- ❖ Que permita una interfaz a futuro (posibles migraciones).

Características de la base de datos

- ➡ Permite que sólo se puedan ver, los registros de la base de datos, cuando el usuario haya introducido un determinado número de registros en ella.
- ➡ Permite configurar los derechos de edición. Puede agregar entradas profesores y alumnos, si lo permite en la base de datos.

- Se puede limitar el número máximo de entradas de cada participante.
- Podemos permitir añadir comentarios a las entradas de la base de datos.
- Podemos habilitar la revisión del profesor antes de que las entradas sean públicas.⁵

Diferencias entre Bases de Datos Transaccionales Y Bases de Datos Datawarehouse.

Base de Datos Transaccional.- Es una Base de Datos Actualizada.

Base de Datos Datawarehouse.- No se actualizan, solo se guardan los datos. El Datawarehouse solo es de consulta histórica, no se modifica.

◆ **Ventajas**

- La utilización de Datawarehouse proporciona una serie de ventajas:
- Proporciona un gran poder de procesamiento de información.
- Permite una mayor flexibilidad y rapidez en el acceso a la información.
- Facilita la toma de decisiones en los negocios.
- Las empresas obtienen un aumento de la productividad.
- Mejora las relaciones con los proveedores y los clientes.
- Transforma los datos en información y la información en conocimiento
- Reduce los tiempos de respuesta y los costes de operación.
- Resumiendo, el Datawarehouse proporciona una información de gestión accesible, correcta, uniforme y actualizada. Proporciona un menor coste en la toma de decisiones, una mayor flexibilidad ante el entorno, un mejor servicio al cliente y permite el rediseño de los procesos.⁶

◆ **Desventajas:**

- Requieren una revisión del modelo de datos, objetos, transacciones y además del almacenamiento.
- Tienen un diseño complejo y multidisciplinar.
- Requieren una reestructuración de los sistemas operacionales.

⁵ Basesdedatos.(n.d.).<http://www3.uji.es/~mmarques/f47/apun/node83.html>

⁶ Basesdedatos.(n.d.).<http://www3.uji.es/~mmarques/f47/apun/node83.html>



- ▶ Tienen un alto coste.
- ▶ Requieren sistemas, aplicaciones y almacenamiento específico

Conceptos de Bases de Datos

Dato: unidad mínima de información, conjunto de caracteres que identifican un objeto. Representación simbólica de algún atributo de una entidad.

En programación un dato es la expresión general que describe las características de las entidades sobre las cuales opera un algoritmo.⁷

Campo: lugar o espacio físico donde se almacena un dato.

NOMBRE

Registro.- Es el conjunto de campos

No. Cuenta	Nombre	Apellido	Teléfono	Edad	RFC
------------	--------	----------	----------	------	-----

Diccionario de Datos.- es la base de los datos que almacena las propiedades y características de la Base de Datos es decir guarda meta-datos.

Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.⁸

Estos diccionarios se desarrollan durante el análisis de flujo de datos y ayuda a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño del proyecto.

Meta-datos.- Donde se almacena la información de los datos.

KEY	Nom_Atributo	Tipo_Dato	Tamaño	Precisión	Descripción
0314534	Nombre	caracter	60		Guarda el nombre

⁷ Henry F. Korth y Abraham Silberschatz .(1998).Fundamentos de Bases de Datos (2da ed.):Mc.Graw Hill

⁸ David M. Kroenke,(1996).Procesamiento de Bases de Datos, Fundamentos, Diseño e instrumentación (5ta ed.). Mexico: Prentice Hall

				del alumno
5656656	Apellido	caracter	60	Guarda el apellido del alumno

MODELOS DE DATOS

Modelo.- Son herramientas que permiten alcanzar un objetivo en forma ordenada.

Modelo de Datos.- es la representación de abstracciones que conduce a la creación de una nueva Base de Datos y desempeña una función prioritaria en el modelo de datos. El modelo de Datos es el enfoque utilizado para la representación de las entidades y sus características dentro de una Base de Datos y esta dividida en tres grandes tipos:

Un modelo de datos es un lenguaje orientado a describir una Base de Datos. Típicamente un Modelo de Datos permite describir:

Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.⁹

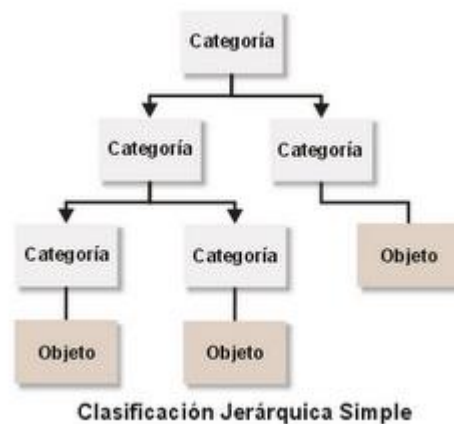
⁹ David M. Kroenke,(1996).Procesamiento de Bases de Datos, Fundamentos, Diseño e instrumentación (5ta ed.). Mexico: Prentice Hall

Modelos de Ejecución

Ejecuta almacenando la información dentro del Gestor de Base de Datos

Modelo de datos jerárquico

Este modelo utiliza árboles para la representación lógica de los datos. Este árbol está compuesto de unos elementos llamados nodos. El nivel más alto del árbol se denomina raíz. Cada nodo representa un registro con sus correspondientes campos.



Una Base de datos jerárquica es un tipo de Sistema Gestor de Bases de Datos que, como su nombre indica, almacenan la información en una estructura jerárquica que enlaza los registros en forma de estructura de árbol (similar a un árbol visto al revés), en donde un nodo padre de información puede tener varios nodos hijo.¹⁰

¹⁰ Modelosdebasesdedatos.(n.d.).<http://www.es.wikipedia.org/basesdedatos>

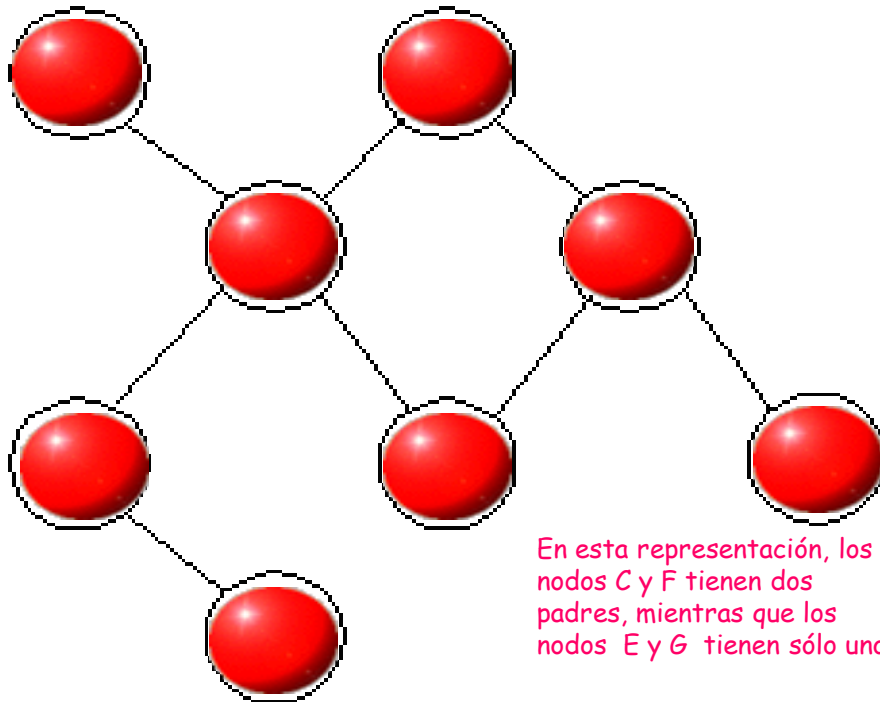
Modelo de datos en red



En este modelo las entidades se representan como nodos y sus relaciones son las líneas que los unen. En esta estructura cualquier componente puede relacionarse con cualquier otro. A diferencia del modelo jerárquico, en este modelo, un hijo puede tener varios padres.

Los conceptos básicos en el modelo en red son:

- ❖ El tipo de registro, que representa un nodo.
- ❖ Elemento, que es un campo de datos.
- ❖ Agregado de datos, que define un conjunto de datos con nombre.
- ❖ Este modelo de datos permite representar relaciones N:M

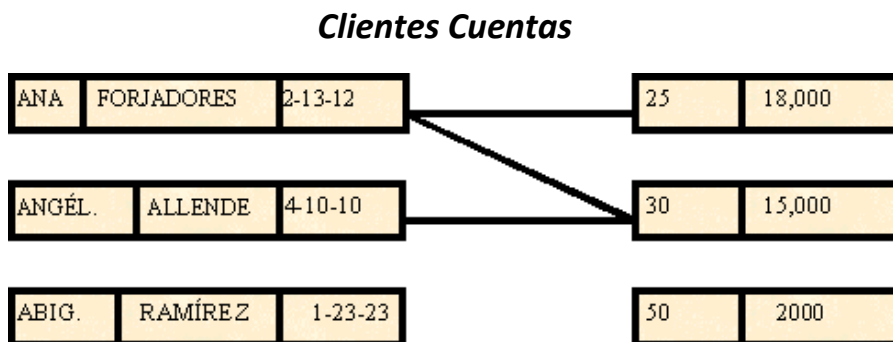


- El concepto básico en el enfoque de red es el *conjunto* ('set'), definido por el comité CODASYL. Un conjunto está constituido por dos tipos de registros que mantienen una relación de muchos a muchos
- Para conseguir representar este tipo de relación es necesario que los dos tipos de registros estén interconectados por medio de un registro conector llamado *conjunto conector*

LOS CONJUNTOS POSEEN LAS SIGUIENTES CARACTERÍSTICAS

- El registro padre se denomina *propietario* del conjunto, mientras que el registro hijo se denomina *miembro*.
- Un conjunto está formado en un solo registro propietario y uno o más registros miembros

Una base de datos de red se compone por una colección de registros que se conectan entre si por medio de ligas. Un registro equivale a una entidad y un campo a un atributo del modelo entidad relación. Los campos contienen exclusivamente valores atómicos. Una liga es una relación que se establece solamente entre dos registros; es decir; debe utilizarse una liga para cada relación entre una pareja de registros.¹¹



¹¹ Modelosdebasesdedatos.(n.d.).<http://www.es.wikipedia.org/basesdedatos>

Modelo de datos relacional

Su creador fue Edgar F. Codd, este modelo es el más utilizado actualmente ya que utiliza tablas bidimensionales para la representación lógica de los datos y sus relaciones.

Algunas de sus principales características son:

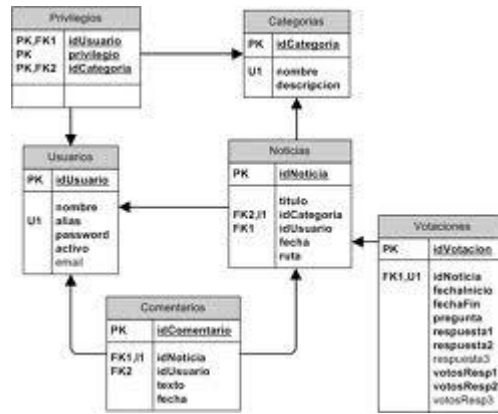
- Se basa en el Modelo Matemático de la Teoría de Conjuntos
- Utiliza Lenguaje Estructurado de Consultas conocido como SQL.
- Relaciones Dinámicas.
- Tuplas (registros)

El modelo relacional es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Tras ser postuladas sus bases en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos.

Su idea fundamental es el uso de «relaciones». Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados «tuplas». Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar, esto es, pensando en cada **relación** como si fuese una **tabla** que está compuestas por *registros* (cada fila de la tabla sería un registro o *tupla*), y *columnas* (también llamadas *campos*).¹²

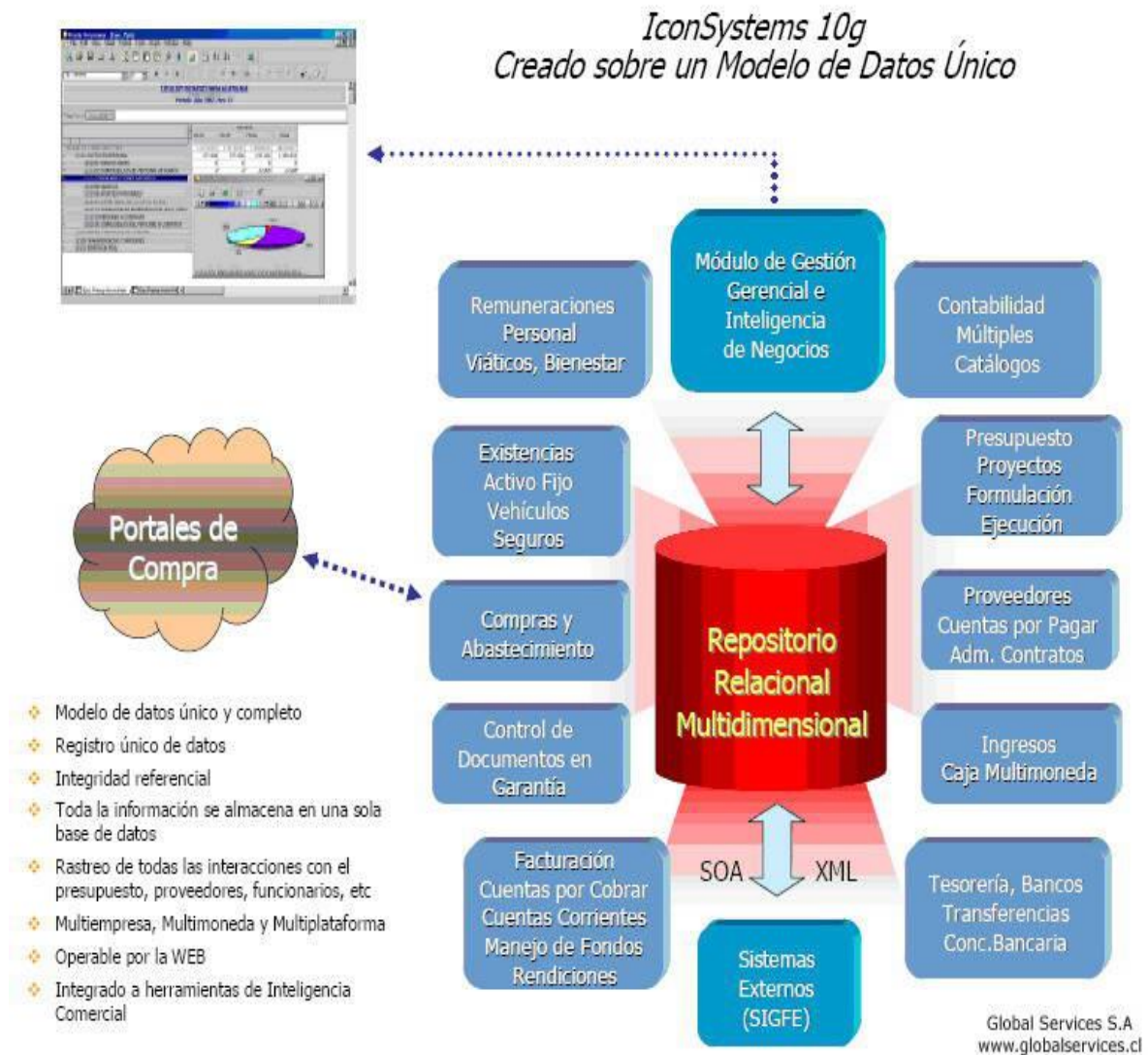
¹² Henry F. Korth y Abraham Silberschatz.(2002). Fundamentos de Bases de Datos (4ta ed.):Mc.Graw Hill



Modelo Orientado a Objetos

El modelo de datos orientado a objetos es una extensión del paradigma de programación orientado a objetos. Los objetos entidad que se utilizan en los programas orientados a objetos son análogos a las entidades que se utilizan en las bases de datos orientadas a objetos puros pero con una gran diferencia: los objetos del programa desaparecen cuando el programa termina su ejecución, mientras que los objetos de la base de datos permanecen. A esto se le denomina persistencia.

El modelo orientado a objetos se basa en encapsular código y datos en una única unidad, llamada objeto.



Un objeto tiene asociado:

- ✚ Un conjunto de variables que contienen los datos del objeto.
- ✚ Un conjunto de mensajes a los que el objeto responde.
- ✚ Un método, que es un trozo de código para implementar cada mensaje. Un método devuelve un valor como respuesta al mensaje.
- ✚ El término mensaje, no implica el uso de un mensaje físico en una red, si no que se refiere al paso de solicitudes entre objetos.
- ✚ La capacidad de modificar la definición de un objeto sin afectar al resto del sistema está considerada como una de las mayores ventajas del modelo de programación orientado a objetos.¹³

¹³ Henry F. Korth y Abraham Silberschatz.(2002). Fundamentos de Bases de Datos (4ta ed.):Mc.Graw Hill

RESUMEN UNIDAD 1

- Los datos son datos sin refinar. La información es el resultado de procesar los datos para dejar ver su significado, Una información precisa, relevante y oportuna es la clave para una buena toma de decisiones, que a su vez es la clave de éxito organizacional en un ambiente global
- Los datos suelen ser almacenados en una base de datos. Para implementar una base de datos y administrar su contenido, se requiere un sistema de administración de base de datos (DBMS). El DBMS sirve como intermediario entre el usuario y la base de datos. La base de datos contiene los datos que se han capturado y los datos acerca de los datos conocidos también como metadatos
- Las bases de datos evolucionaron a partir de sistemas de archivos manuales y luego computarizados. En un sistema de archivos, los datos se guardan en archivos independientes, cada uno de los cuales requiere sus propios programas de administración de datos. Aun cuando este método de administración es ahora anticuado, entender sus características hace que el diseño de la base de datos sea más fácil de entender
- Los sistemas de administración de bases de datos se perfeccionan para manejar las debilidades inherentes del sistema de archivos. Más que

depositar datos en archivos independientes, un DBMS presentan la base de datos al usuario final como un solo depósito. Esta disposición hace posible que los datos se compartan, lo cual elimina el problema potencial de las islas de información. Además el DBMS hace cumplir la integridad de datos , elimina la redundancia y promueve la seguridad de datos.

UNIDAD II

DISEÑO DE BASE DE DATOS

OBJETIVO: Diseño de bases de datos conociendo niveles de abstracción y tomando como base el diseño conceptual, principios de normalización.

- 1.-Conceptos de diseño**
- 2.- Herramientas de diseño**
- 3.-Niveles de Abstracción**
- 4.- Modelo conceptual, elementos y características**
- 5.- Principios de normalización**

CONCEPTOS DE DISEÑO

INDEPENDENCIA DE LOS DATOS

Esta se refiere a la libertad que pueda existir para modificar algunos de los esquemas sin que exista la necesidad de reescribir los programas de aplicación.

Existen básicamente dos tipos de independencia:

- a. INDEPENDENCIA FISICA.- Esta se presenta cuando es posible la modificación del esquema físico sin afectar a los esquemas restantes. Las principales razones para llevar a cabo una modificación del esquema físico serán un ajuste en el hardware de almacenamiento o una redistribución de los datos en él.
- b. INDEPENDENCIA LOGICA.- Ocurre cuando se modifica el esquema conceptual sin afectar al resto de los esquemas. Básicamente se modifica el esquema conceptual cuando cambian las características de los datos a almacenar.

Es relativamente más sencillo y probable lograr la independencia física puesto que una modificación del esquema conceptual, (estructuras, ligas y demás) inevitablemente requerirá de modificaciones el código para su manipulación

El diseño de base de datos en gran medida un arte, no una ciencia. *Existen* (para repetir) algunos principios científicos que pueden ser aplicados al problema; sin embargo, existen muchos aspectos del diseño que esos simplemente no abordan en lo absoluto. Como consecuencia, diversos teóricos y profesionales de las bases de datos han propuesto metodologías de diseño* —algunas de ellas bastante rigurosas, otras no tanto, pero todas ellas adecuadas hasta cierto punto— que pueden ser

utilizadas como un ataque en contra de lo que hasta el momento de la publicación este libro era todavía un problema intratable; es decir, el problema de encontrar el diseño lógico "único" que sea sin lugar a dudas el adecuado. Puesto que estas metodologías son en su mayoría adecuadas hasta cierto punto, existen pocos criterios objetivos para preferir algún enfoque sobre todos los demás. Sin embargo, presentamos un enfoque bien conocido que sí tiene por lo menos el mérito de ser utilizado ampliamente en la práctica.

a. **El diseño de bases de datos** no es sólo una cuestión de obtener las estructuras de dato: correctas, también la integridad de los datos es un ingrediente clave (quizás el ingrediente clave). Repetiremos y ampliaremos esta observación en muchas de las ideas que aparecen en los capítulos siguientes.

b. Nos ocuparemos en gran medida de lo que podríamos denominar **diseño independiente de la aplicación**. En otras palabras, nos ocupamos principalmente de lo que son los datos, en lugar de cómo serán *usados*. En este sentido, la independencia de la aplicación es necesaria por la muy buena razón de que normalmente —o tal vez siempre— se da el caso de que al momento del diseño no son bien conocidos todos los usos que se darán a los datos; por lo tanto, queremos un diseño que sea *robusto*, en el sentido de que no se invalida por el advenimiento de requerimientos de la aplicación que no se previeron al momento del diseño. Para ponerlo de otra forma, lo que tratamos de hacer principalmente, es *obtener el esquema concepto correcto*; es decir, nos interesa producir un diseño lógico abstracto que sea independiente del hardware, del sistema operativo, del DBMS, del lenguaje, del usuario, etcétera.

NIVELES DE ABSTRACCIÓN

Funciones del Administrador de Base de Datos

1.-Definir el esquema lógico de Bases de Datos:

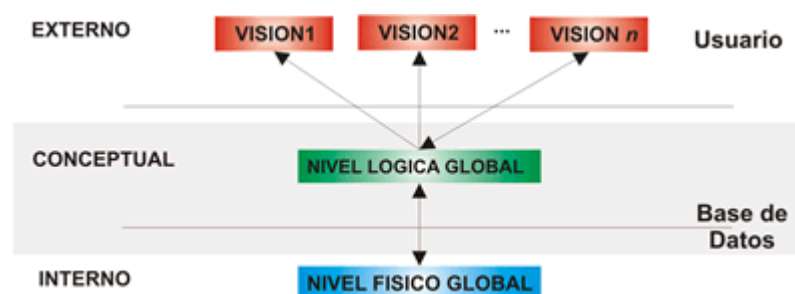
◆ Niveles de Abstracción

Conceptual.- Define el diseño y planeación de la Base de Datos.

Lógico.- Manejador de Base de datos, debe saber lógicamente como funciona la Base de Datos.

Vistas.- lo que el usuario ve en la Base de Datos al realizar una consulta.

- **Nivel físico:** El nivel mas bajo de abstracción describe como se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.¹⁴
- **Nivel lógico:** Describe que datos se almacenan en la base de datos y que relaciones existen entre esos datos. Los administradores de base de datos, que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.³
- **Nivel de Vistas:** Describe solo parte de la base de datos completa. Para que su interacción con el sistema se simplifique, se define la abstracción a nivel de vistas. El sistema puede proporcionar muchas vistas para la misma base de datos.³



¹⁴ Nivelesdeabstraccion.(n.d.).<http://www3.uji.es/~mmarques/f47/apun/node83.html>

2.-Servir de Enlace entre Usuarios de la Base de Datos y el Sistema Manejador de Base de Datos (Oracle, SQL, etc.)

3.-Definir Estrategias de Respaldo y Recuperación de Datos: prever cada cuando se van a hacer los respaldos. Hacer un programa que automáticamente respalde la información. Tener un plan de respaldo y recuperación es muy importante.

4.-Monitorear la Eficiencia y Capacidad de Respuesta de la Base de Datos: la Base de Datos en las organizaciones se encuentran por servidores, si un servidor se satura, se alenta, se pasma, el Administrador de Base de Datos debe monitorear que la Eficiencia y la capacidad de los servidores este bien.

Arquitectura: Abstracción de Datos

Con los Niveles de Abstracción de Datos se diseña la Base de Datos:

Ⓢ Nivel Externo

Ⓢ Nivel Lógico

Ⓢ Nivel Físico

Ⓢ Nivel Conceptual

❖ Otra Opción de Niveles de Abstracción:



Nivel Físico: tiene que ver con el SGBD se refiere a saber qué equipo va a ser un servidor y cliente o si va a ser una red. Se refiere al Hardware en donde va a ser almacenada la Base de Datos.

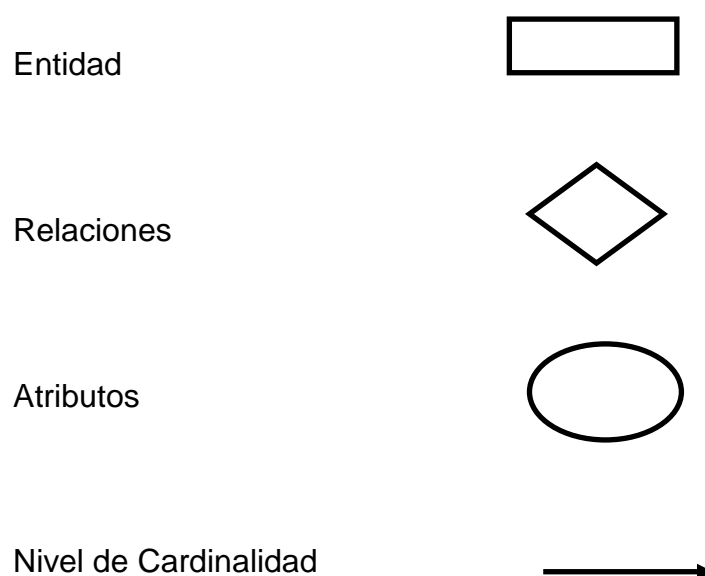
Nivel Lógico: Nivel donde se describen los datos que se almacenan en la Base de Datos y las relaciones que existen entre ellas y las restricciones que deberán cumplir.

Nivel Conceptual: que es el siguiente nivel más alto de abstracción, se describe cuáles son los datos reales que están almacenados en la base de datos y qué relaciones existen entre los datos.

Modelo Conceptual

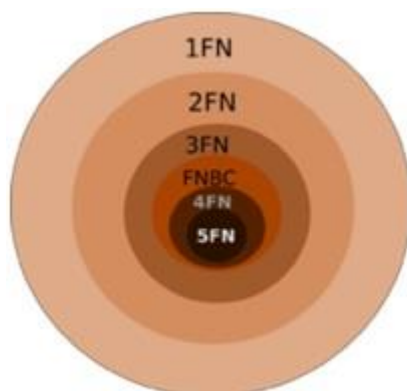
Se encuentra en el Nivel Lógico Modelado, es la representación grafica de datos dentro de este modelo se encuentra el Modelo Entidad Relación (permite representar gráficamente entidades, relaciones , atributos y Nivel de Cardinalidad).

Modelo conceptual: El modelo conceptual del lenguaje UML contiene tres elementos principales: los bloques básicos de construcción, las reglas de combinación de los bloques básicos y algunos mecanismos comunes.¹⁵



¹⁵ Modelosdebasesdedatos.(n.d.).<http://www.es.wikipedia.org/basesdedatos>

NORMALIZACION



Es un proceso que clasifica relaciones, objetos, formas de relación y demás elementos en grupos, en base a las características que cada uno posee. Si se identifican ciertas reglas, se aplica una categoría; si se definen otras reglas, se aplicará otra categoría.

Estamos interesados en particular en la clasificación de las relaciones BDR. La forma de efectuar esto es a través de los tipos de dependencias que podemos determinar dentro de la relación. Cuando las reglas de clasificación sean más y más restrictivas, diremos que la relación está en una forma normal más elevada. La relación que está en la forma normal más elevada posible es que mejor se adapta a nuestras necesidades debido a que optimiza las condiciones que son de importancia para nosotros:

- La cantidad de espacio requerido para almacenar los datos es la menor posible;
- La facilidad para actualizar la relación es la mayor posible;
- La explicación de la base de datos es la más sencilla posible.

Primera forma normal

Para que una relación esté en primera forma normal (1 FN), debe ser solamente una relación propia, una matriz m por n, donde:

- Ninguna celda de la matriz está vacía;
- El valor n cualquier columna está definido por el dominio para dicho atributo.
- Cada tupla tiene una clave que la identifica en forma unívoca, pero dicha clave no significa orden.

Segunda forma

Una relación está en segunda forma normal (2FN) solamente si todos los atributos son dependientes en forma completa de la clave.

Descripción de la Segunda Forma Normal (2Fn)

Su nombre ya nos indica el hecho de que la segunda forma normal es por lo general el próximo paso de normalización y descomposición. Para ser accesible a la normalización, y poder ser puesta en segunda forma normal, la relación debe poseer las siguientes propiedades:

- Debe estar en primera forma normal
- Debe tener una clave compuesta.

Tercer forma normal

Una relación se encuentra en tercera forma normal (EFN) si no existen transitividades entre sus atributos y si ya se encuentra en 2 FN.

Descripción

Una relación R a poner en tercera forma normal debe estar en la segunda forma normal. Es muy común que R sea una sub-relación; la relación original estaba en primera forma normal (para ponerla en segunda forma normal fue descompuesta en varias sub-relaciones). Estas son ahora candidatas a una descomposición adicional.

Recordamos que las propiedades de la segunda forma normal (2Fn) son:

- Tenemos una matriz $m \times n$ con un valor determinado para cada componente de cada tupla.
- Cada valor es obtenido a partir de un dominio propiamente definimos
- Cada valor contiene una clave, ya sea simple o compuesta
- Cada componente no clave es dependiente en forma completa de su clave.

En consecuencia es evidente que tenemos, o bien una clave simple, o una clave compuesta de la cual todos los componentes no clave son dependientes en forma completa.

Cuarta forma normal

Dependencias multievaluadas

La tercera forma normal toma en cuenta la dependencia transitiva y provee una reducción óptima universal, excepto para los casos infrecuentes de dependencia multievaluadas. Ha quedado claro en épocas recientes que es posible una reducción adicional en este caso, y esto es lo que se lleva a cabo mediante la cuarta forma normal.

Existe una dependencia multievaluada cuando un valor de una variable está siempre asociado con varios valores de otra u otras variables dependientes que son siempre las mismas y están siempre presentes.

RESUMEN UNIDAD 2

El modelo entidad-relación (ERM) utiliza los diagramas entidad relación para representar la base de datos conceptual como es vista por el usuario final. Los principales componentes del Modelo Entidad-relación son entidades, relaciones y atributos, también incluye notaciones de conectividad y cardinalidad

La conectividad describe la clasificación (1:1, 1:M o M:N). La cardinalidad expresa el número específico de casos de entidad asociados con un caso de una entidad relacionada. Conectividad y cardinalidades suelen estar basadas en reglas de negocios.

Los niveles de abstracción que son considerados para realizar un diseño de base de Datos son los Niveles de Abstracción de Datos se diseña la Base de Datos:

- ④ Nivel Externo(el que el usuario ve)
- ④ Nivel Físico(la base de las bases de datos servidores)
- ④ Nivel Lógico(tiene relación con el lenguaje)
- ④ Nivel Conceptual(el diseño propiamente)

Otro elemento que se debe considerar es la normalización para que el diseño sea funcional, este cuenta con las siguientes reglas:

Primera forma normal

Para que una relación esté en primera forma normal (1 FN), debe ser solamente una relación propia

2da forma normal: Una relación está en segunda forma normal (2FN) solamente si todos los atributos son dependientes en forma completa de la clave.

Tercer forma normal: Una relación se encuentra en tercera forma normal (3FN) si no existen transitividades entre sus atributos y si ya se encuentra en 2 FN.

Un modelo de datos es una abstracción de un complejo ambiente de datos reales. Los diseñadores de bases de datos usan modelos de datos para comunicarse con programadores de aplicaciones y usuarios finales. Los componentes básicos de modelado de datos son entidades, atributos, relaciones y restricciones. Se usan reglas de negocio para identificar y definir los componentes básicos de modelado dentro de un ambiente específico real.



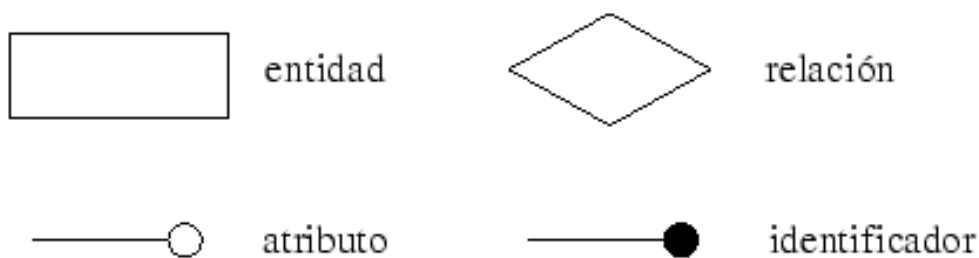
UNIDAD 3

MODELO ENTIDAD-RELACION

- Analizar las características y elementos del modelo entidad relación para realizar el diseño de base de datos.
-
1. Uso del modelo Entidad Relación como base del Diseño
 - a. Concepto
 - b. Etapas de diseño
 - c. Técnicas de Diseño
 2. Conceptos
 - a. llaves
 - b. integridad
 - c. diccionario de datos
 - d. relaciones y tipos
 - e. cardinalidad
 3. Tipos de campos
 4. Graficación del modelo relacional
 5. todos los conceptos aplicados a el modelo entidad relación

Modelo Entidad-Relación

Es el modelo conceptual más utilizado para el diseño de bases de datos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades. Fue introducido por Peter Chen en 1976. Está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.¹⁶



Entidad.- es un objeto que se encuentra en la vida real y que tiene características. Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc.

Hay dos tipos de entidades:

- **Una entidad débil** es una entidad cuya existencia depende de la existencia de otra entidad.
- **Una entidad fuerte** es una entidad que no es débil.¹⁷

¹⁶ Basesdedatos.(n.d.).<http://www3.uji.es/~mmarques/f47/apun/node83.html>

¹⁷ C. J. Date, Sergio L. María Ruiz Faldón y Felipe López Gamino. (2001).Introducción a los sistemas de bases de datos (2da ed.): Prentice Hall

Atributos.- características de una entidad

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos.¹⁸

Atributo Compuesto: está compuesto por más de un dato, se puede subdividir. Es un atributo con varios componentes, cada uno con un significado por sí mismo. Ejemplo:

Calle	Numero	Colonia
Francisco I. Madero	Núm. 102	Col San Pedro

Atributo Simple: Es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Ejemplo:

No. Cuenta
02323

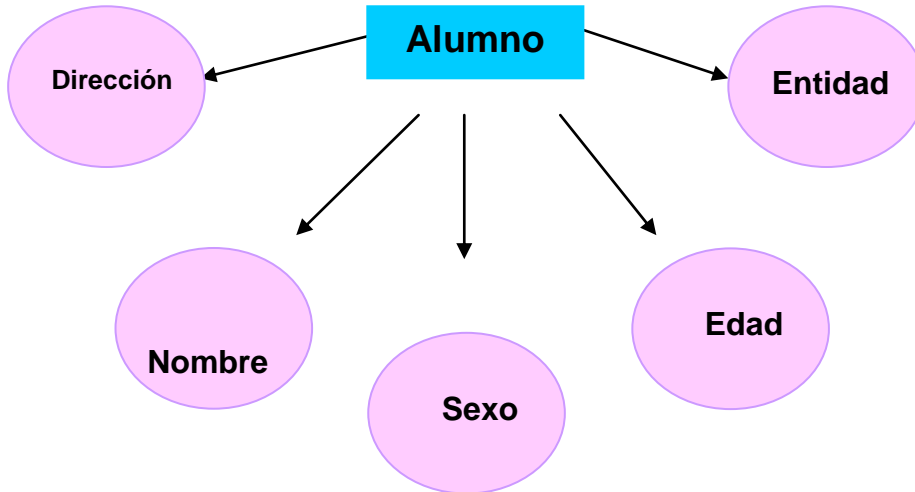
Atributo con un solo valor.- es un atributo único de identificación

Atributo con varios valores.- Contiene varios valores de datos.

¹⁸ Basesdedatos.(n.d.).<http://www3.uji.es/~mmarques/f47/apun/node83.html>

Diseño Modelo Conceptual

❖ **Entidad - Relación**



❖ **Campos:** Almacena información, lugar donde se almacenan datos.

Nombre	Edad	Sexo	Dirección
Andrés	22	m	Morelos nº 1
Fernando	21	m	Juárez 18
Luís	16	m	Hidalgo 207

Dominio.-conjunto de valores que contienen un campo son los posibles valores de un atributo.

Nombre (char)	Juan Pérez
---------------	------------

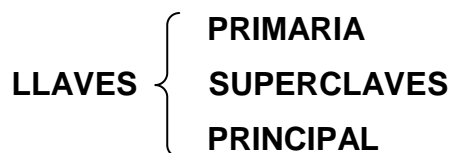


Relación.- Asociación de dos Entidades. Se representa mediante un rombo etiquetado en su interior con un verbo. Este rombo se debe unir mediante líneas con las entidades (rectángulos) que relaciona.¹⁹

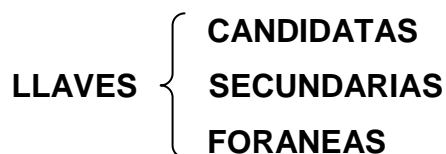
Claves

Es un subconjunto del conjunto de atributos comunes en una colección de entidades, que permite identificar unívocamente cada una de las entidades pertenecientes a dicha colección. Asimismo, permiten distinguir entre sí las relaciones de un conjunto de relaciones.²⁰

Clave primaria: Es una clave candidata, elegida por el diseñador de la base de datos, para identificar unívocamente las entidades en un conjunto de entidades.²¹



Clave candidata: Permiten ordenar la información que contenga varios valores en un mismo atributo. Dada una superclave, si ésta deja de serlo removiendo únicamente uno de los atributos que la componen, entonces ésta es una clave candidata.²²



²⁰ Fundamentosbasesdedatos.(n.d.).http://es.wikipedia.org/wiki/Diagrama_entidad-relaci%C3%B3n

²¹ David M. Kroenke,(1996).Procesamiento de Bases de Datos, Fundamentos, Diseño e instrumentación (5ta ed.). Mexico: Prentice Hall

²² Fundamentosbasesdedatos.(n.d.).http://es.wikipedia.org/wiki/Diagrama_entidad-relaci%C3%B3n

Llaves Compuestas.- cuando no puede existir una llave principal que permita hacer únicos los registros.

Cardinalidad

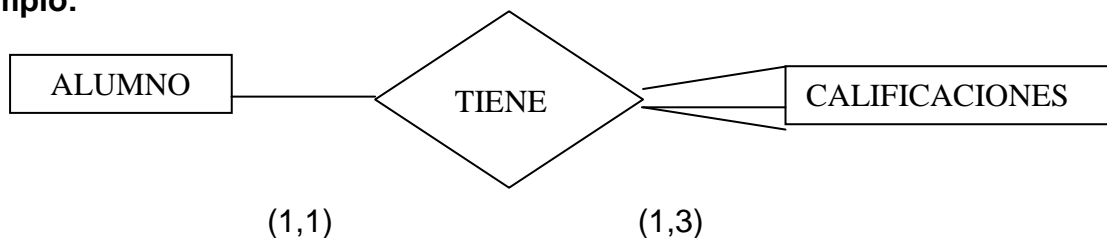
Expresa el número específico de ocurrencias en una entidad asociada con una ocurrencia de la entidad relacionada.

Ocurrencia.-cuantas veces se van a registrar los valores en una Entidad.

La Cardinalidad se Expresa:

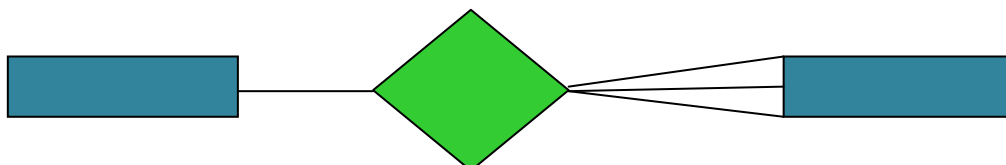
- DE UNO A UNO
- DE UNO A MUCHOS
- DE MUCHOS A UNO
- MUCHOS A MUCHOS

Ejemplo:



Cardinalidad de Uno a Muchos

Cardinalidad Modelo Pata De Gallo





Fuerza de las Entidades

- *Dependencia Existencial.*- el dato no depende de ningún otro dato.
- *Relación Dependiente.*-cuando existe una relación Dependiente se considera una **entidad fuerte**, no tiene llave primaria.

❖ Entidad Fuerte



- **Relación Independiente.**- cuando existe una relación independiente se considera una relación débil y tiene una llave primaria.

❖ Entidad Débil

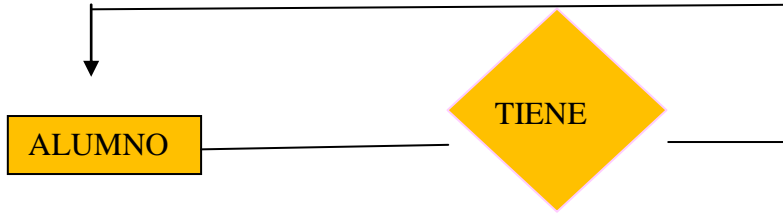


Grado de Relación

Se refiere a Relaciones o Asociaciones que tienen las Entidades. Son:

- **RELACION UNARIA**
- **RELACION BINARIA**
- **RELACION TERNARIA**
- **MAS DE TRES... GRADO 4, GRADO 5, ETC.**

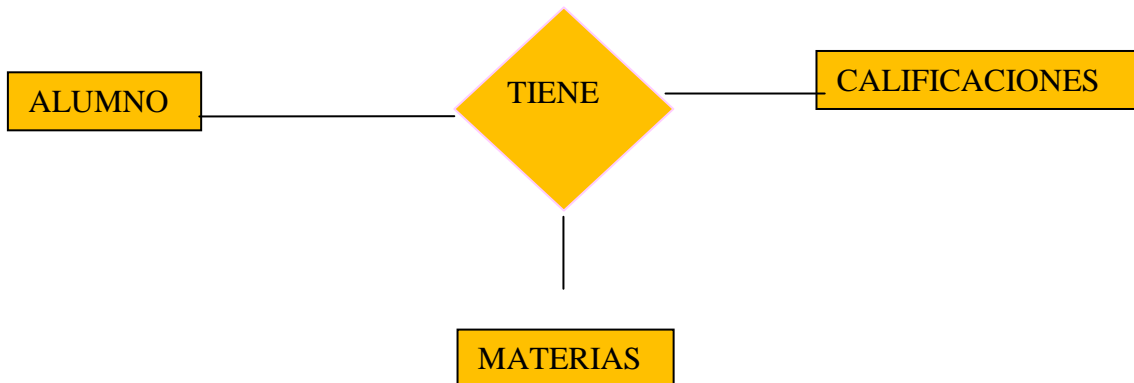
❖ RELACION UNARIA



❖ RELACION BINARIA



❖ RELACION TERNARIA





RESUMEN UNIDAD 3

El modelo Entidad-relación: Es el modelo conceptual más utilizado para el diseño de bases de datos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades. Fue introducido por Peter Chen en 1976

Entidad.- es un objeto que se encuentra en la vida real y que tiene características

Claves: Es un subconjunto del conjunto de atributos comunes en una colección de entidades, que permite identificar unívocamente cada una de las entidades pertenecientes a dicha colección. Asimismo, permiten distinguir entre sí las relaciones de un conjunto de relaciones

La cardinalidad: Expresa el número específico de ocurrencias en una entidad asociada con una ocurrencia de la entidad relacionada.

Grado de Relación: Se refiere a Relaciones o Asociaciones que tienen las Entidades. Son: RELACION UNARIA, RELACIÓN BINARIA, RELACIÓN TERNARIA MÁS DE TRES... GRADO 4, GRADO 5, ETC.



UNIDAD 4

ADMINISTRACIÓN DE BASE DE DATOS

- Desarrollo de una base de datos a través de los diversos modelos de bases de datos.
 1. Uso de un Sistema Gestor de Base de Datos
 2. Elaborar diccionario de datos, llaves, relaciones en un gestor de base de datos
 3. Implementación del modelo entidad relación en un gestor de base de datos relacional



Metodología de Base de Datos

(Pasos para Desarrollar el Software y la SGBD)

❖ Algunos Modelos para el Desarrollo de Software:

- *Prototipo*: es un modelo que facilita al programador la creación del modelo de Software, ya que se basa en la creación de prototipos según vayan siendo las necesidades de los usuarios.
- *Espiral*: es un modelo de desarrollo evolutivo que acompaña a la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistematizados del modelo lineal secuencial.
- *Evolutivo*: este es un modelo iterativo, el cual permite el desarrollo de versiones cada vez más complejas de software
- *RAD*: desarrollo rápido de aplicaciones es lineal y secuencial, se enfatiza en un ciclo de desarrollo extremadamente rápido; utilizando un enfoque de construcción basado en componentes
- *Clásico o de Cascada*: Es un refinamiento del modelo de etapas. Existe un reconocimiento de los ciclos de retroalimentación entre etapas, y una guía para

confinar las retroalimentaciones a las etapas sucesivas con el objetivo de minimizar el costo del trabajo involucrado en retroalimentaciones a través de muchas etapas.²³

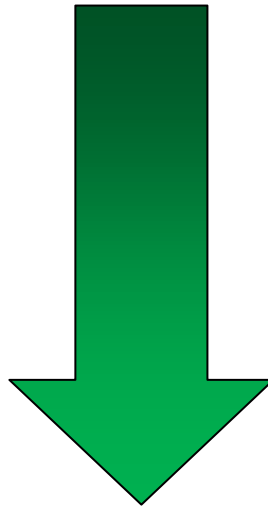
DESARROLLO DE SOFTWARE

Acoplamiento

Se refiere a seguir el Desarrollo del Software al mismo tiempo que el Desarrollo de la Base de Datos.

Software

Análisis	Requerimientos
Diseño	Diagramas Esquemas
Desarrollo	Software
Pruebas	Pruebas del Software
Implementación	
Mantenimiento	

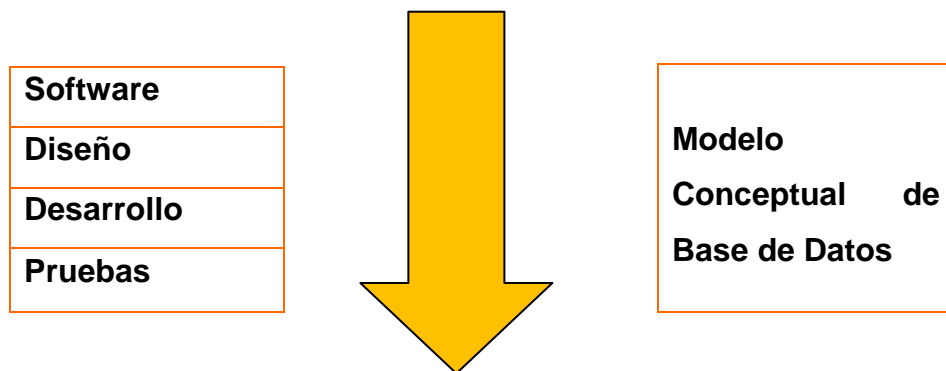


Base de Datos

Analizar qué información es útil para almacenar en la Base de Datos
Diseñar el Modelo Conceptual mediante el Modelo Entidad Relación
Desarrollo del SGBD
Pruebas que almacenan la información que queremos
Implementar el SGBD
Mantenimiento

²³ Ingenieriadelssoftware.(n.d.).<http://html.rincondelvago.com/modelos-de-desarrollo-de-software.html>

Pasos que sigue una Metodología de Cascada (clásico)



DICCIONARIO DE DATOS

Un diccionario de datos es un conjunto de [metadatos](#) que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Diccionario de datos

El DBMS debe proporcionar una función de **diccionario de datos**. Este diccionario puede ser visto como una base de datos por derecho propio (aunque una base de datos del sistema más que como una base de datos del usuario). El diccionario contiene "datos acerca de los datos" (en ocasiones llamados *metadatos* o *descriptores*); es decir, *definiciones* de otros objetos del sistema, en lugar de simples "datos en bruto". En particular, todos los diversos esquemas y transformaciones (externos, conceptuales,



etcétera) y todas las diversas restricciones de seguridad y de integridad, serán almacenadas en el diccionario, tanto en forma fuente como objeto.

Un diccionario extenso incluirá además mucha información adicional; mostrará por ejemplo qué programas utilizan qué partes de la base de datos, qué usuarios necesitan qué informes, etcétera. El diccionario podría incluso —y de hecho, debería— estar integrado dentro de la base de datos que define, e incluir por lo tanto su propia definición. En realidad, debe ser posible consultar el diccionario del mismo modo que cualquier otra base de datos, de manera que, por ejemplo, sea posible saber qué programas o usuarios se podrían ver afectados por un cambio propuesto al sistema

Tipos De Datos

CARACTER	NUMERICO	LOGICO
Texto	Float	F/V
String	Int	S/N
Varchar	Long	
char	double	

◆ ***Numericos:***

- ❖ ***Integer:*** es un tipo simple, y dentro de estos, es ordinal. Es una variable numérica que puede tomar valores positivos o negativos.
- ❖ ***Long:*** corresponde a tipos de datos simples (ordinales), teniendo la capacidad de almacenar el doble de lo que puede almacenar un dato integer,
- ❖ ***Real (real):*** corresponde con los números reales, son importantes para los cálculos y se caracterizan por tratar fundamentalmente con valores decimales.
- ❖ ***Double:*** corresponde a los números reales, con la única diferencia de que este tipo de datos almacena el doble de lo que puede almacenar un real.



◆ **Lógico:**

❖ Boolean (lógico): es el que te permite usar variables que disponen sólo de dos posibles valores: cierto (1) o falso (0).

◆ **Carácter:**

❖ Char y string (carácter y cadena): puedes usar variables o constantes que representen un valor alfanumérico. Las cadenas de caracteres (strings) puedes contener en una sola variable más de un carácter.²⁴

Ejemplo de diccionario datos

Nombre de Archivo: BDPlantilla		Fecha de creación: 27/04/2008.	
Descripción: Base de datos que contendrá la plantilla de personal del instituto.			
Campo	Tamaño	Tipo de Dato	Descripción
CURP	18	Caracter	Clave Unica de Registro de Población
cPaterno	30	Caracter	Apellido paterno del Empleado
cMaterno	30	Caracter	Apellido materno del empleado
cNombre	45	Caracter	Nombre del Empleado.
cDomicilio	60	Caracter	Domicilio actual donde reside el empleado
cColonia	45	Carácter	Colonia del domicilio donde reside el empleado
cCiudad	45	Carácter	Ciudad donde reside el empleado
cEstado	45	Carácter	Entidad federativa de residencia del empleado
cTelefono	12	Carácter	Número telefónico del empleado
nPostal	6	Numérico	Código postal del domicilio del empleado
cFamiliar	65	Carácter	Nombre de un familiar directo del empleado
FDomicilio	65	Carácter	Domicilio de familiar directo del empleado
FTelefono	12	Carácter	Teléfono de familiar directo del empleado
Relaciones: CURP con BDNomina		Campos Clave: CURP, cPaterno, cMaterno	

²⁴ C. J. Date, Sergio L. María Ruiz Faldón y Felipe López Gamino. (2001). Introducción a los sistemas de bases de datos (2da ed.): Prentice Hall



CLAVES

El modelo relacional siempre ha puesto énfasis en el concepto de las claves; aunque como hemos visto, en realidad sólo son un caso especial (muy importante) de un fenómeno más general.

En esta sección, centraremos nuestra atención específicamente en las claves.

Nota: Aunque aquí las ideas básicas son bastante sencillas, por desgracia existe un factor importante de complicación: los valores *nulos*. La posibilidad de que (por ejemplo) una clave externa determinada pudiera permitir nulos, empaña la imagen de manera considerable. Sin embargo, los nulos constituyen por sí mismos un tema extenso, que sería inadecuado abordar en detalle en este momento. Por lo tanto, por razones pedagógicas, en esta sección ignoraremos casi en su totalidad a los nulos; en el capítulo 18, cuando tratemos los nulos en general, volveremos a explicar el impacto de los nulos sobre las claves. (De hecho, nosotros creemos firmemente que los nulos son un error y que nunca debieron ser presentados, aunque estaría mal ignorarlos por completo en un libro de esta naturaleza.)

Claves candidatas Sea R una varrel. Por definición, el conjunto de todos los atributos de R tiene la propiedad de **unicidad**, lo que significa que ningún par de tupias que estén dentro del valor de R en un momento dado, pueden estar duplicadas entre sí. En la práctica, a menudo se da el caso de que algún subconjunto propio del conjunto de atributos de R tiene también la propiedad de unicidad; por ejemplo, en el caso de la varrel de proveedores, V , el subconjunto que contiene únicamente al atributo $V\#$ tiene esa propiedad. Estos hechos constituyen la intuición que está detrás de la definición de la *clave candidata*:

■ Sea K un conjunto de atributos de la varrel R . Entonces K es una **clave candidata** de R si, y solamente si, posee las dos propiedades siguientes:*

a. **Unicidad:** Jamás, ningún valor válido de R contiene dos tupias distintas con el mismo valor de K .

b. **Irreductibilidad:** Ningún subconjunto propio de K tiene la propiedad de unicidad.



Observe que toda varrel tiene por lo menos una clave candidata. La propiedad de unicidad de dichas claves es muy clara. En cuanto a la propiedad de irreductibilidad, la idea es que si especificáramos una "clave candidata" que *no* fuera irreducible, el sistema no estaría al tanto del estado de las cosas; y por lo tanto, no podría hacer cumplir adecuadamente la restricción de integridad asociada. Por ejemplo, suponga que definiéramos la combinación {V#,CIUDAD} —en vez de sólo {V#}— como la clave candidata para proveedores. Entonces el sistema no haría cumplir la restricción de que los números de proveedor son "globalmente" únicos; en su lugar, sólo haría cumplir la restricción más débil de que los números de proveedor son "localmente" únicos dentro de una ciudad. Por esta razón, entre otras, requerimos que las claves candidatas no incluyan atributo alguno que sea irrelevante para los fines de identificación única

La teoría formal subyacente a los sistemas relacionales se denomina **modelo relacional de datos**. Este modelo se ocupa sólo de aspectos lógicos, no de aspectos físicos. Aborda los tres principales aspectos de los datos: la **estructura** de los datos, la **integridad** de los datos y la **manipulación** de los datos. El aspecto *estructural* tiene que ver con las relaciones *como tales*; el aspecto de *integridad* tiene que ver (entre otras cosas) con las **claves primaria y externa**; y el aspecto de *manipulación* tiene que ver con los operadores (restringir, proyectar, juntar, etcétera).

El principio de información establece que todo el contenido de información de una base de datos relacional está representado en una y sólo una forma; es decir, como valores explícitos en posiciones de columnas dentro de filas en relaciones.

Cada relación tiene un **encabezado** y un **cuerpo**; el encabezado es un conjunto de parejas nombre-de-columna:nombre-de-tipo, mientras que el cuerpo es un conjunto de filas que se apegan al encabezado. El encabezado de una relación dada puede ser visto como un **predicado** y cada fila en el cuerpo denota una cierta **proposición verdadera**, la cual se obtiene sustituyendo ciertos valores de **argumento** del tipo apropiado por los *indicadores de posición* o **parámetros** del predicado. En otras palabras, los *tipos* son (conjuntos de) cosas de las cuales podemos hablar y las



relaciones son (conjuntos de) cosas que podemos decir acerca de las cosas de las que podemos hablar. Juntos, los tipos y las relaciones son **necesarios y suficientes** para representar cualquier dato que queramos (esto es, al nivel lógico).

El **optimizador** es el componente del sistema que determina cómo implementar las peticiones del usuario (las cuales se ocupan del *qué*, no del *cómo*). Por lo tanto, ya que los sistemas relacionales asumen la responsabilidad de navegar alrededor de la base de datos almacenada para localizar los datos deseados, en ocasiones se les describe como sistemas de **navegación automática**.

La optimización y la navegación automática son prerrequisitos para la **independencia física de los datos**.

El **catálogo** es un conjunto de varrels del sistema que contienen **descriptores** de los diversos elementos que son de interés para el sistema (varrels base, vistas, índices, usuarios, etcétera).

Los usuarios pueden consultar el catálogo exactamente del mismo modo en que consultan sus propios datos.

Implementar la Normalización en la base de datos

Siempre que un analista de BD arma una base de datos, queda a su cargo descomponer dicha base en grupos y segmentos de registros. Este proceso es la descomposición; el mismo es necesario independientemente de la arquitectura de la base de datos. Sin embargo, para la bd relacional, la acción puede dividirse y expresarse en términos formales es denominada normalización²⁵. Tiene doble propósito:

- 1.-Permite Eliminar la información redundante innecesaria de los diferentes esquemas de relación o tipos de registros lógicos de la Base de Datos.
- 2.-Evitar la perdida injustificada de información al remover n-adas de la base de datos.

²⁵ Normalización.(n.d.).<http://www.lawebdelprogramador.com/temas/tecdisenio.php>



La Normalización debe hacerse en Base a análisis del Significado de la Información tomando en cuenta las dependencias funcionales existentes entre los atributos que participan en la relación y verificando que no se hayan omitido esquemas de relación.

Normalización son las Reglas para la Unión entre atributos y Entidades.

Es una teoría de descomposición de Entidades lógicas dentro de un conjunto de entidades que contienen mas propiedades físicas con esto se desean obtener los siguientes beneficios:

- 1.-Mayor Flexibilidad
- 2.-Asegurar que los atributos estén en las entidades apropiadas.
- 3.-Disminuir la redundancia de los Datos
- 4.-Maximizar cambios estructurados al desarrollar nuevas aplicaciones.
- 5.-Mantenimiento sencillo de la Información.

Normalización es un conjunto de reglas que sirven para ayudar a los diseñadores a desarrollar un esquema que minimice los problemas de lógica. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica. Las guías que la normalización provee crean el marco de referencia para simplificar la estructura.²⁶

Entidad Compuesta: cuando existe una relación de muchos a muchos (m:n) se crea una entidad compuesta o entidad puente. Esta tabla o entidad compuesta se utiliza para vincular las entidades o tablas que originalmente se relacionaron en una relación m:n.

²⁶ David M. Kroenke,(1996).Procesamiento de Bases de Datos, Fundamentos, Diseño e instrumentación (5ta ed.). Mexico: Prentice Hall



La **estructura de la Entidad Compuesta** incluye claves ajenas o foráneas y por lo menos las claves primarias de las Entidades o tablas que se han de vincular. Para llevar a cabo este proceso se debe definir una clave primaria compuesta utilizando la combinación de claves foráneas y crear una nueva clave foránea.

Clave del Empleado	Proyecto	Nombre
1	A	LUIS
2	B	RAUL
3	C	PEDRO
4	D	ANGEL

PROYECTO	Nom_Proj	Empl_Asign
1	A	LUIS
2	B	RAUL
3	C	PEDRO
4	D	ANGEL

La Entidad Compuesta:

A1 LUIS

2B RAUL

3C PEDRO

4A ANGEL



Reglas De Normalización

PRIMERA REGLA.- Los datos deben ser atómicos en un campo solo debe tener un solo atributo, una sola propiedad, un solo numero.

*Establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas. Ésta es una regla muy fácil de seguir. Observe el esquema de la tabla Clientes de la base de datos.*²⁷

Ejemplo: El id de un Empleado debe ser único.

SEGUNDA REGLA.- Deben existir llaves claves primarias Dependientes y estas no deben ser Nulos. Menciona la Integridad Referencial así como la Consistencia de Datos.

Establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un

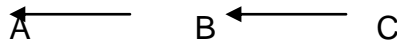
²⁷ Henry F. Korth y Abraham Silberschatz .(1998).Fundamentos de Bases de Datos (2da ed.):Mc.Graw Hill



*término que describe a aquellos datos que no dependen de la clave de la tabla para identificarlos.*²⁸

TERCERA REGLA.- Dependencia Transitiva

*Señala que hay que eliminar y separar cualquier dato que no sea clave. El valor de esta columna debe depender de la clave. Todos los valores deben identificarse únicamente por la clave.*²⁹



Primero saber que hay en C sin pasar por B pero nunca Borrar B.

Cuarta Forma Normal (4FN): Una tabla se encuentra en 4FN si, y sólo si, para cada una de sus dependencias múltiples no funcionales $X \twoheadrightarrow Y$, siendo X una super-clave que, X es o una clave candidata o un conjunto de claves primarias.³⁰

Quinta Forma Normal (5FN) Una tabla se encuentra en 5FN si:

- La tabla esta en 4FN
- No existen relaciones de dependencias no triviales que no siguen los criterios de las claves. Una tabla que se encuentra en la 4FN se dice que esta en la 5FN si, y sólo si, cada relación de dependencia se encuentra definida por las claves candidatas.³¹

²⁸ Normalización.(n.d.).http://www.wikilearning.com/tutorial/disenio_de_bases_de_datos_en_sql-normalizacion/21129-4

²⁹ Henry F. Korth y Abraham Silberschatz .(1998).Fundamentos de Bases de Datos (2da ed.):Mc.Graw Hill

³⁰ Basesdedatos.(n.d.).<http://www.monografias.com/trabajos5/norbad/norbad.shtml>

³¹ Basesdedatos.(n.d.).<http://www.monografias.com/trabajos5/norbad/norbad.shtml>

Ejemplo de las reglas de Normalización ³²

Formalización CERO

Usuarios				
nombre	empresa	direccion_empresa	url1	url2
Joe	ABC	1 Work Lane	abc.com	xyz.com
Jill	XYZ	1 Job Street	abc.com	xyz.com

Primer nivel de (F/N)

1. Eliminar los grupos repetitivos de las tablas individuales.
2. Crear una tabla separada por cada grupo de datos relacionados.
3. Identificar cada grupo de datos relacionados con una clave primaria.

Usuarios				
userid	nombre	empresa	direccion_empresa	url
1	Joe	ABC	1 Work Lane	abc.com
1	Joe	ABC	1 Work Lane	xyz.com
2	Jill	XYZ	1 Job Street	abc.com
2	Jill	XYZ	1 Job Street	xyz.com

³² Ejemplosdebasesdedatos.(n.d.).http://www.wikilearning.com/tutorial/disenio_de_bases_de_datos_en_sql-normalizacion/21129-4



Segundo nivel de F/N

1. Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
2. Relacionar estas tablas mediante una clave externa.

Usuarios			
userId	nombre	empresa	direccion_empresa
1	Joe	ABC	1 Work Lane
2	Jill	XYZ	1 Job Street

urls		
urlId	relUserId	url
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com



Tercer nivel de F/N.

3. Eliminar aquellos campos que no dependan de la clave.

usuarios		
userId	nombre	relEmpresald
1	Joe	1
2	Jill	2

empresas		
emprld	empresa	direccion_empresa
1	ABC	1 Work Lane
2	XYZ	1 Job Street

Urls		
urlld	RelUserId	url
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

RESUMEN UNIDAD 4

Durante la etapa de implementación se instala el hardware, el software del DBMS, los programas de aplicación y se implementa el diseño de base de datos. Durante las etapas iniciales de esta fase, el sistema entra en un ciclo de codificación, prueba y depuración hasta estar listo para su entrega. Se crea una base de datos real y el



sistema se personaliza mediante la creación de tablas, vistas y autorizaciones del usuario, etc.

El sistema debe considerar un ciclo de vida para su creación como se menciono anteriormente, se deben establecer las reglas de normalización considerando las reglas de negocio dicho en otras palabras el sistema se codifica de acuerdo al sistema gestor de base de datos elegido, el cual en la actualidad todos tienen como estándar el Lenguaje estructurado de consultas.



UNIDAD 5

ADMINISTRACIÓN DE BASE DE DATOS

- Administración de base de datos, tanto de los elementos como de los datos.
 - Lenguaje estructurado de consultas empleado en el modelo relacional
 - Herramientas de administración de base de datos
 - Procesamiento de transacciones
 - Herramientas de explotación y administración de la información en la base de datos



Lenguaje estructurado de consultas empleado en el modelo relacional

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por Microsoft Jet. SQL se utiliza para crear objetos QueryDef, como el argumento de origen del método OpenRecordset de la propiedad RecordSource del control de datos. También se puede utilizar con el método Execute para las bases de datos Jet y crear consultas SQL de paso a través para manipular bases de datos remotas.

1.1. Componentes del SQL

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estas son instrucciones para crear, actualizar y manipular las bases de datos.

1.2 Comandos

Existen dos tipos de comandos SQL:

- los DDL que permiten crear y definir nuevas bases de datos, campos e índices.
- los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Comandos DDL	
Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Comandos DML	
Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Utilizado para modificar los valores de los campos y registros especificados
DELETE	Utilizado para eliminar registros de una tabla de una base de datos

1.3 Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que se desean consultar.

Cláusula	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico



1.4 Operadores Lógicos

Operador	Uso
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son verdaderas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las condiciones es verdadera.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

1.5 Operadores de Comparación

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor ó Igual que
>=	Mayor ó Igual que
=	Igual que
BETWEEN	Utilizado para especificar un intervalo de valores.
LIKE	Utilizado en la comparación de un modelo
In	Utilizado para especificar registros de una base de datos

1.6 Funciones de Agregado

Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un solo valor para cada grupo de registros.

Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado



RESUMEN UNIDAD 5

Los comandos de SQL se pueden dividir en dos categorías generales: comandos del lenguaje de definición de datos(DDL) y comandos de lenguaje de manipulación de datos(DML)

Los tipos de datos estándar del ANSI están soportados por todos los vendedores de los sistemas gestores de bases de datos los tipos básicos son numero/flotante, entero, carácter, booleano.

Los comandos básicos de definición de datos permiten crear tablas, índices y vistas

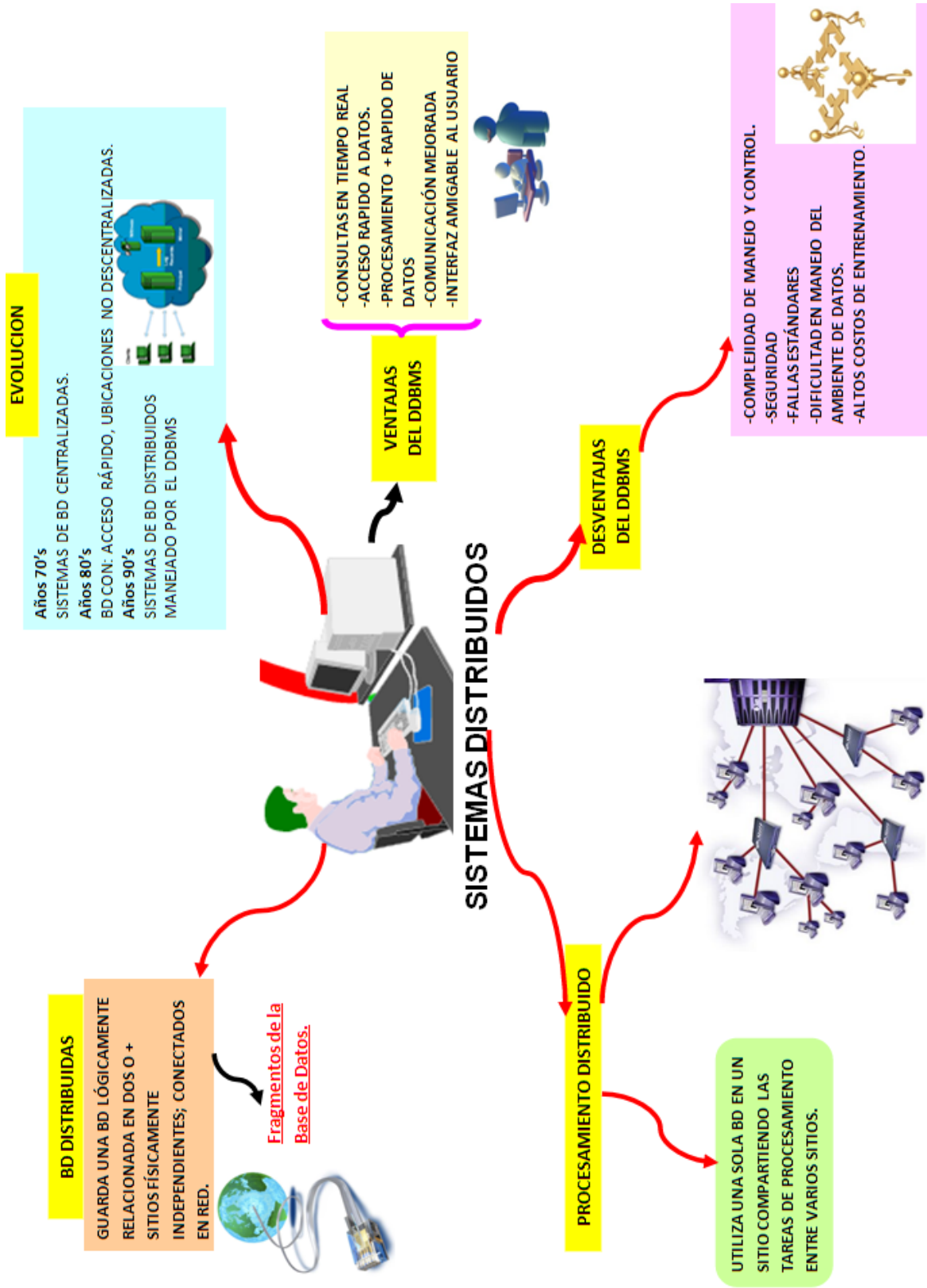
Los comandos de DML(Manipulación de Datos) permiten agregar, modificar y eliminar renglones de las tablas los comandos básicos son

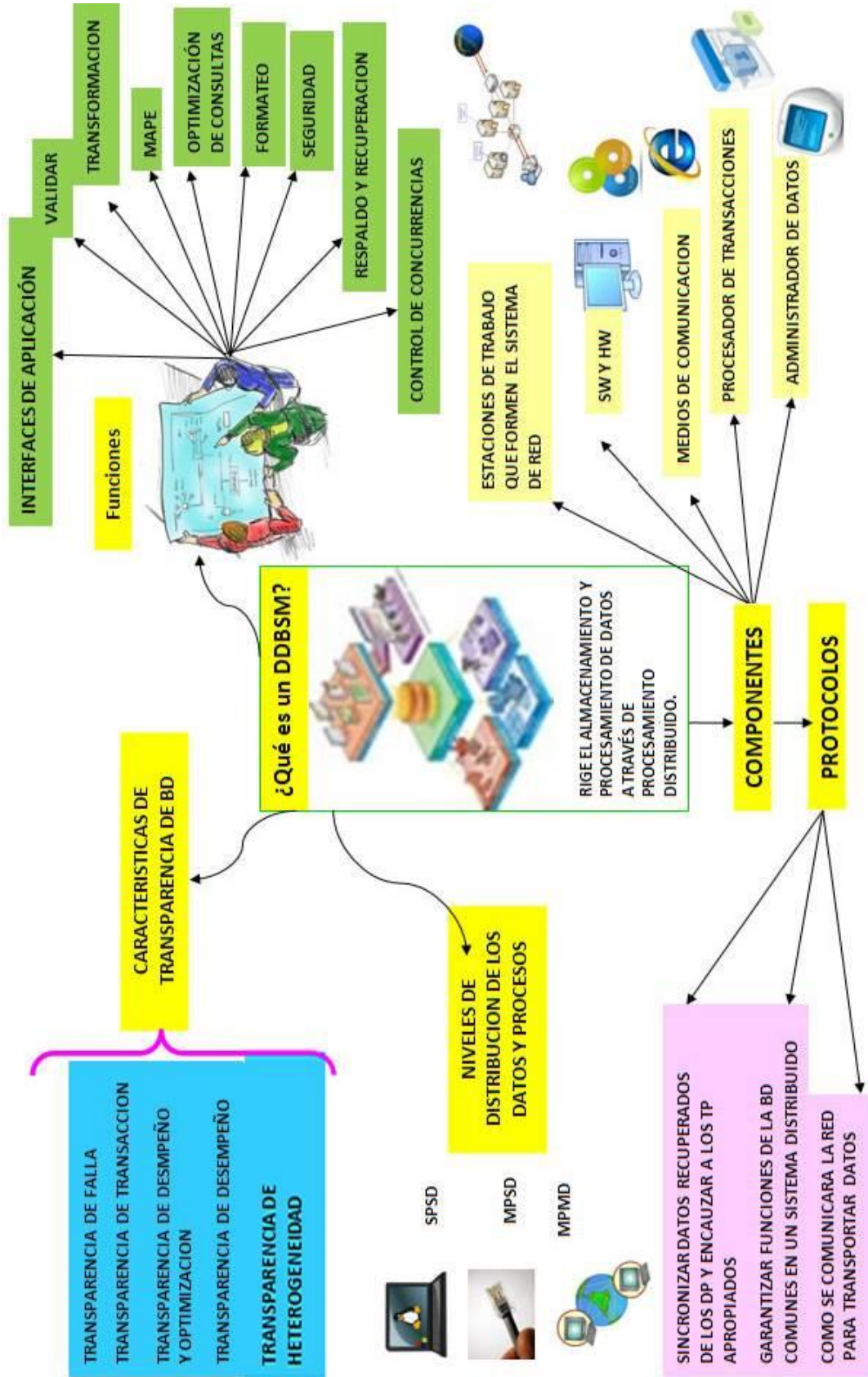
SELECT,INSERT,UPDATE,DELETE,COMMIT Y ROLLBACK

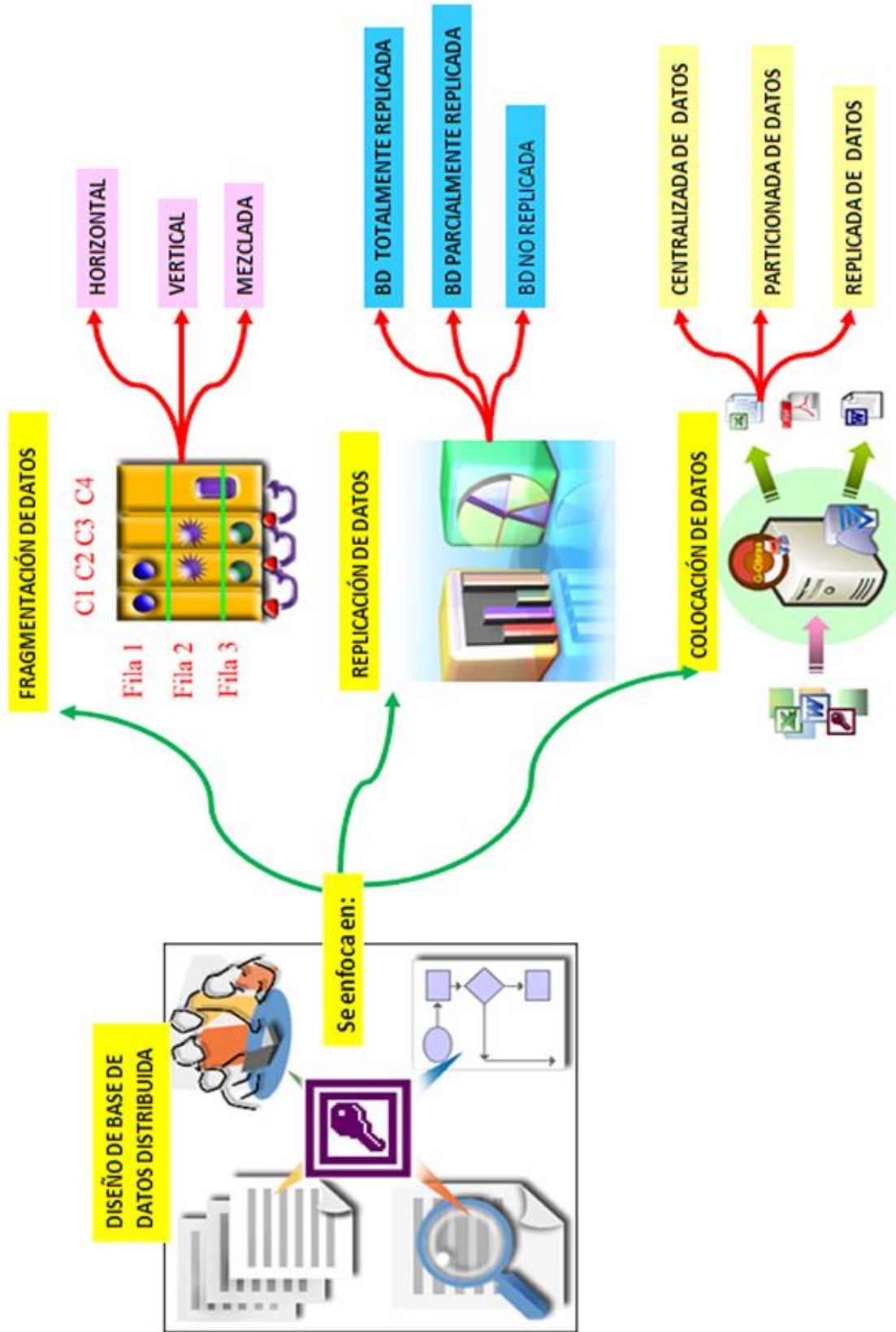
Los operadores lógicos, and, or y not.

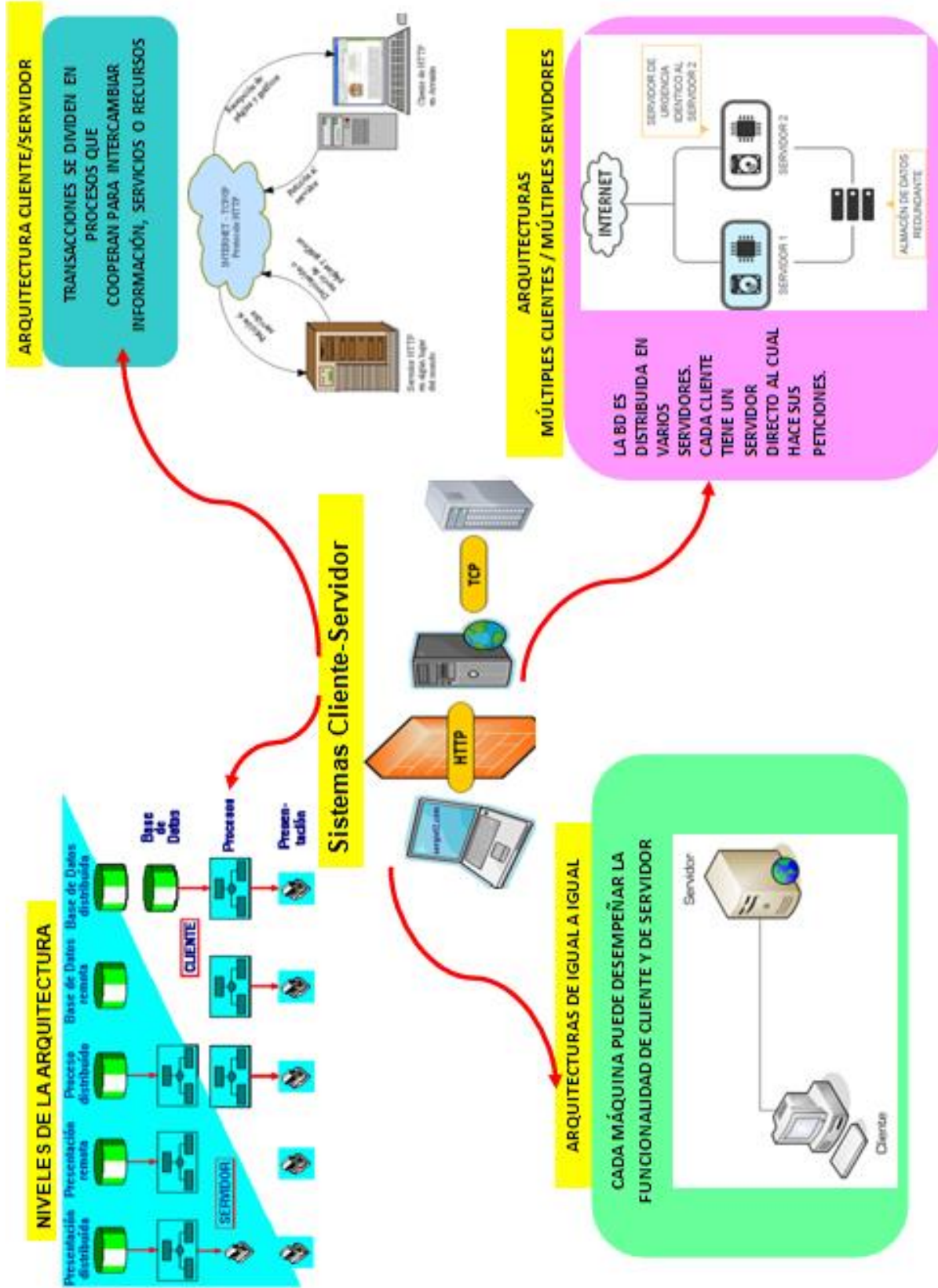
Los operadores de comparación son <,>,<>=,between,like

Las funciones de agregado constan de las sentencias, count , avg, sum, max, y min estas por lo regular se utilizan con la instrucción group by.







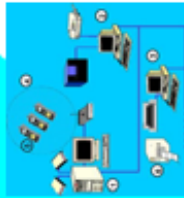


CARACTERÍSTICAS FUNCIONALES



- ORDENADOR CENTRAL O ÚNICO.
- CONTIENE TODOS LOS DATOS.
- CONTROLA EL ACCESO MULTIPLE A TERMINALES
- TERMINALES FUNCIONAN COMO "ESCLAVOS" DEL ORDENADOR CENTRAL.
- USUARIO TIENE PRIORIDADES Y DERECHOS.

CARACTERÍSTICAS FÍSICAS



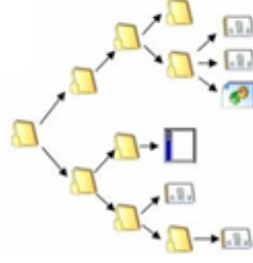
- ORDENADOR ÚNICO SOPORTA TODOS LOS PROCESOS DE LA ORGANIZACIÓN
- UNA GRAN BASE DE DATOS DONDE RESIDEN TODOS LOS DATOS DEL ORGANISMO.
- IMPRESORAS Y TERMINALES, U ORDENADORES PERSONALES CON EMULACION DE TERMINAL, PUESTOS DE TRABAJO CONECTADOS EN GRUPOS (CLUSTERS) AL ORDENADOR CENTRAL.

SISTEMAS CENTRALIZADOS



DEFINICION

ESTRUCTURA CENTRALIZADA Y JERÁRQUICA, DIVIDIDA EN DEPARTAMENTOS



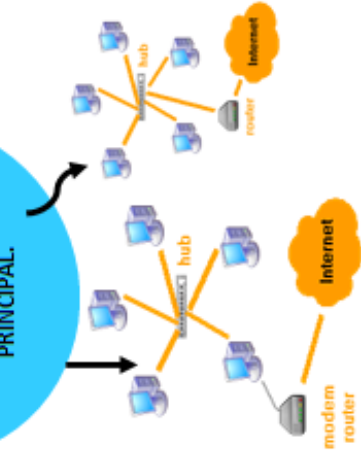
EL SISTEMA ABARCA

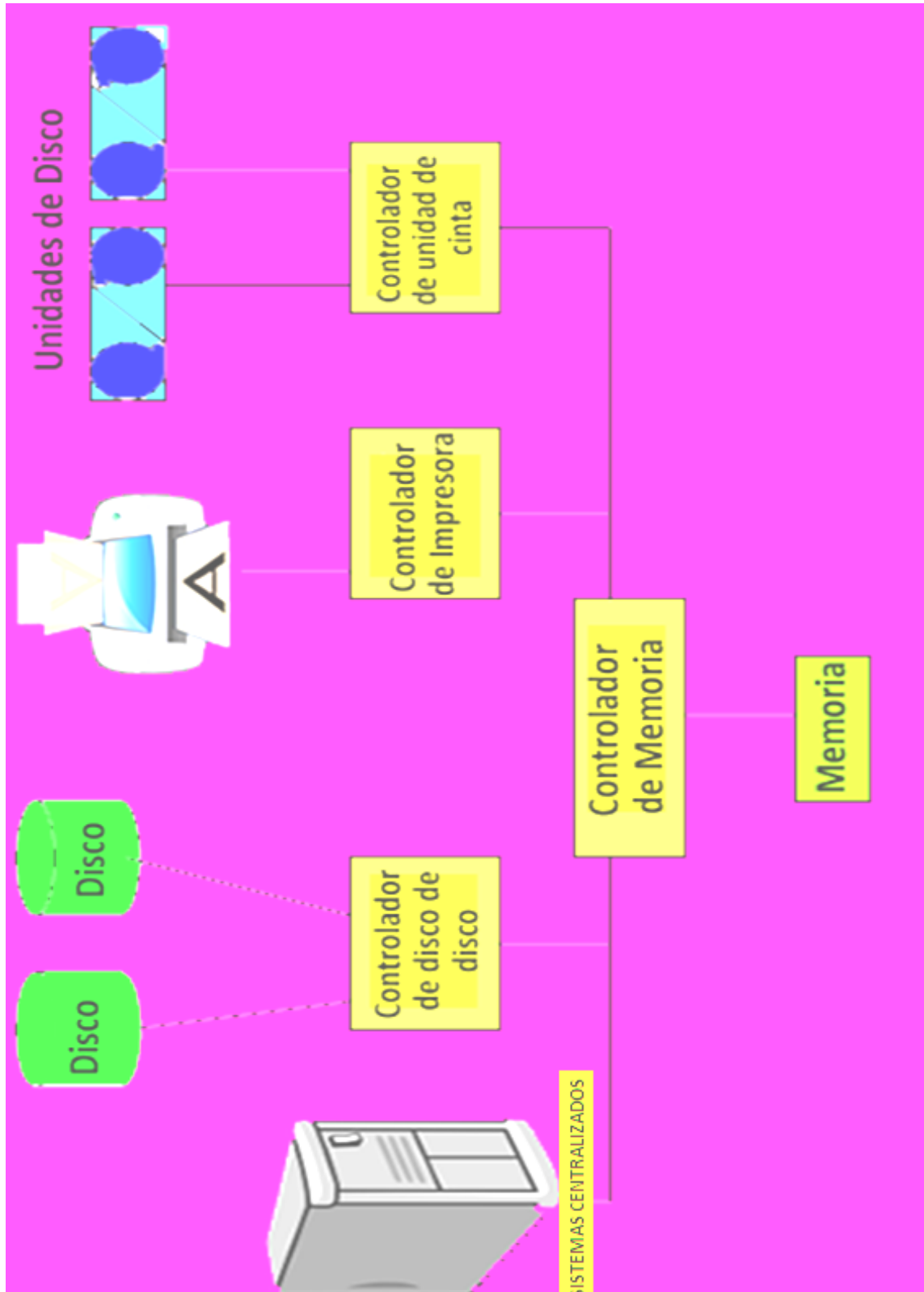
EQUIPO MONOPUESTO1 y Multipuesto2 y la BD CENTRALIZADA EN EL SISTEMA PRINCIPAL.

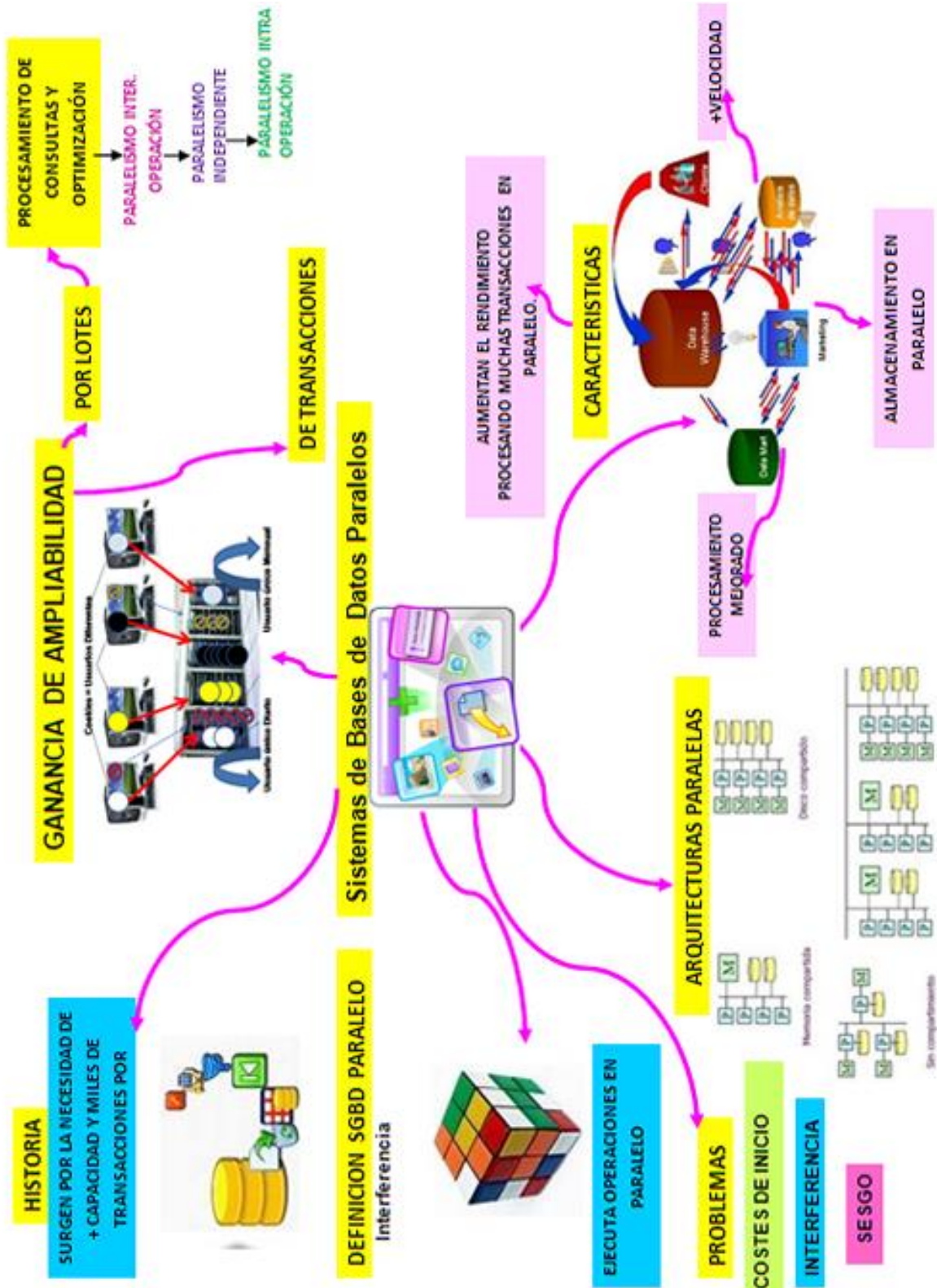


CARACTERÍSTICAS LÓGICAS

Ejecución de todos los procesos en el ordenador corporativo. Si la empresa dispone de comunicaciones, todos los puestos de trabajo están conectados al ordenador formando una "estrella".









BIBLIOGRAFIA

- **“Bases de datos”** Deakin, Rose. Trillas(2006)
- **“Concepción y diseño de bases de datos”**, Miguel, Adoración y Pattini, Mario.. Alfaomega, Computec (2007)
- **“Distributed Databases, Principles and Systems”**, Stefano Ceri, Giuseppe Pelagatti Ed. Mc Graw Hill (2005)
- **“Diseño y programación de base de datos”** por Angel Cabo, editorial Visión, ISBN 9788498214598, 2007.
- **“ Diseño y administración de bases de datos”**, Hansen, Gary y otros. Ed. Prentice Hall (2007)
- **“Fundamentos de bases de datos”** , Korth, Henry.. Mc. Graw Hill (2008)
- **“Introducción a los sistemas de bases de datos”**, Escrito por C. J. Date
- **“Procesamiento de bases de datos”**, Kronke, David. Prentice Hall 2006
- **“Sistemas de base de datos: diseño, implementación y gestión”**, 9^a edición, Carlos Coronel, Steven Morris, Peter Rob ISBN-10: 0538469684 ISBN-13: 9780538469685, 700 Páginas CB, © 2010 Publicado
- **“Sistemas de bases de datos, administración y uso”**, Tsai, Alice, Ed. Prentice Hall(2004)