Introducción

Dónde estamos?

Teoría de Bases de Datos

Reseña Histórica

BD, SGDB y SBD

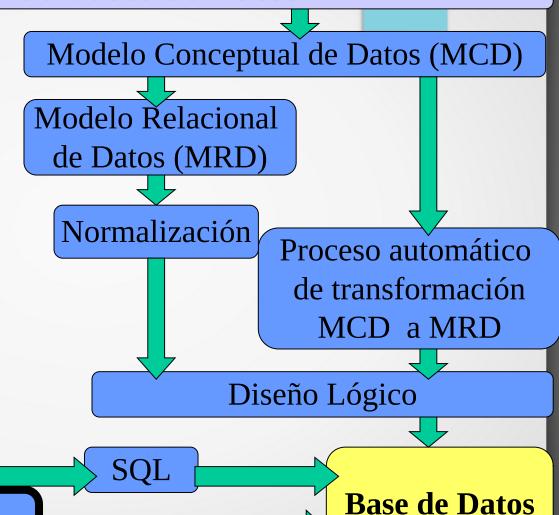
Objetivos y servicios del SGBD

Administrador de BD

• Funciones y objetivos

Flujo de Información en la Organizacion

Ciclo de Vida de la BD Costos y riesgos de una BD



Evolución de las Bases de Datos y de los Sistemas de Gestión de Bases de Datos

Los años 60s y 70s

- Característica: Sistemas Centralizados.
- Estaban pensados para HW y SW de la época (Mainframes).
- Primeras terminales con teclado.
- Procesos por lotes (batch), los datos se almacenaban en cintas.
- Programas relacionados con el nivel físico: existían lenguajes como COBOL, pero se escribía código de bajo nivel para acceder al nivel físico, modificaciones en el nivel físico implicaban modificaciones en los programas.
- Lo crítico: maximizar las transacciones por minuto.

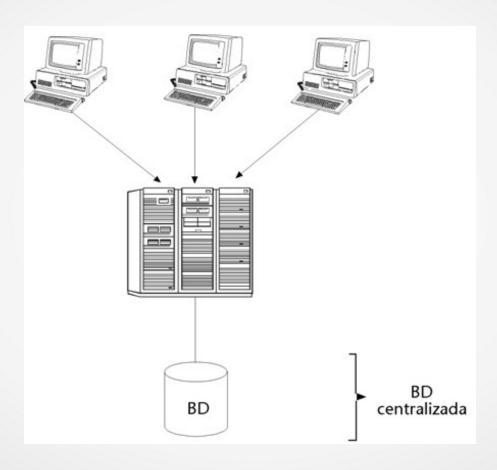
Los años 60s y 70s

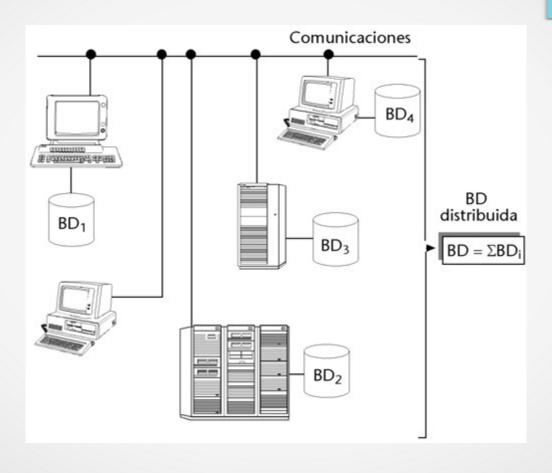
Aplicaciones típicas:

- Reservas y compras de boletos de avión y tren.
- Inventarios de materiales en fábricas militares.
- Administración de cuentas cliente en el ámbito bancario.

- Característica: Sistemas de Gestión de BD relacionales.
- Minis y Micros: extendieron el uso de la informática a pymes e instituciones.
- Creció rápidamente el desarrollo de aplicaciones.
- Los SGBD eran complejos e inflexibles, y solo accesibles para personal muy cualificado.
- Aparecen los SGBD relacionales: permite que las aplicaciones sean independientes de la estructura física de la BD.
- Se estandariza el lenguaje SQL en el año 1986.
- Aparecen los ordenadores personales y con ellos algunos pseudo SGDB personales como dBase.

- Característica: Distribución, C/S y 4GL.
- Innumerables PCs en las empresas, muchas BD personales, muchos proveedores, fusiones de empresas provocan necesidad de distribución.
- Una BD distribuida se da cuando el SGBD permite que un programa de aplicación pueda acceder a varias BD como si se tratase de una sola.
- Ideal para distribución: los SGBD sean de una misma marca (homogéneos).
- Realidad: los SGBD son heterogéneos.
- El estándar SQL tiene una gran relevancia en la tarea de homogeneizar distintos SGBD.

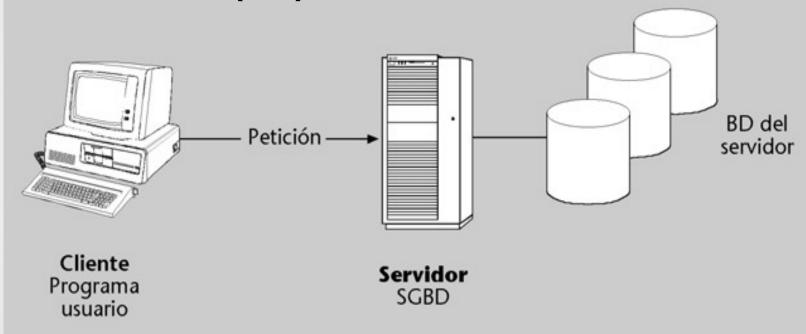




Existen razones por las cuales diseñar una **BD** distribuida:

- Disponibilidad: si una BD está distribuida y replicada asegura que si un sistema falla, los otros siguen funcionando.
- Costo: pude reducirse el costo en comunicaciones acercando las BD a los usuarios, distribuyendo la potencia de cálculo en varios ordenadores en vez de un superordenador.

Cliente/Servidor (C/S):



La tecnología C/S ha tenido éxito, por permitir distribución de datos y por crear potentes conglomerados de hosts. Por ejemplo Grids, Clouds, etc.

Cliente/Servidor (C/S): es la tecnología usada para implementar distribución de datos.

- Dos procesos independientes, uno escucha, el otro peticiona.
- El proceso que escucha, se denomina **servidor**, el que realiza las peticiones **cliente**. El servidor es pasivo, reacciona ante una petición.
- Ambos procesos pueden estar en el mismo host.
- El servidor escucha por un puerto, el puerto suele ser el alias del servicio o servidor. Se requiere un protocolo que ambos conozcan.
- No se requiere que los procesos corran en la misma plataforma.
- Servidores típicos: BD, impresión, archivos, web, mail, etc.

Fourth Generation Languages (4GL):

- · Lenguajes fáciles de usar, de aprender y potentes.
- Especializados en el desarrollo de aplicaciones para BD.
- Avanzados en la definición de GUIs.
- Facilidades para realizar peticiones a servidores.

Tendencias actuales

Actualmente los SGBD están en proceso de adaptación permanente a tecnologías como:

- Multimedia, Orientación a Objetos, Datos no estructurados (mails, etc.), Datos jerárquicos (XML), JSON, Internet y la web.
- Bases de datos en memoria.
- Guiones SQL embebidos en HTML.
- WebServices.
- Data Warehouse, Big Data.

Sistemas
Computacionales
Iniciales



Automatizaban funciones de contabilidad

Los Programas imitaban los procedimientos manuales habituales



MUY LENTO E INEFICIENTE

Registros de Archivos en Ordenador Registros en Archivos de Papel



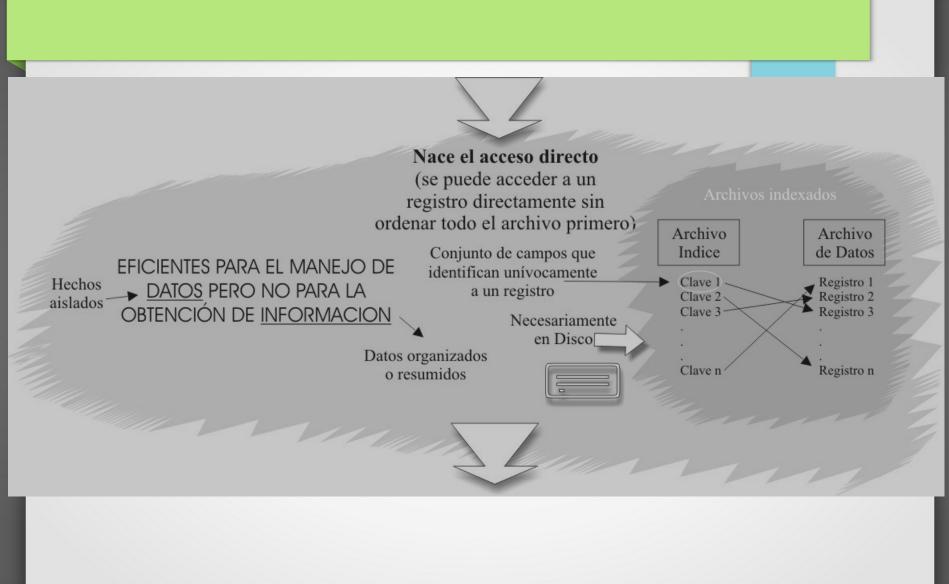
Almacenamiento en Cinta (Discos muy costosos)



Por cada informe o proceso







Nacen los Sistemas de Base de Datos

(Los Sistemas pasan de procesar Datos a procesar Información)

Base de Datos (BD)

Colección de datos que puede ser utilizada por uno o más programas de aplicación

Sistema de Gestión de Base de Datos (SGBD)

Sistema que facilita la gestión de la Base de Datos

Sistema de Base de Datos (SBD)

Base de Datos

- + Sistema de Gestión de Base de Datos
- + Hardware
- + Recursos Humanos

Definición y mantenimiento de la BD Seguridad y Auditoria Compartir datos Lenguaje de programación Copias de Seguridad Lenguaje de Consulta etc.

Usuarios Profesionales Informáticos

Administrador de Base de Datos (ABD)

Objetivos y Servicios del SGBD

Consultas no predefinidas y complejas

- Objetivo fundamental de un SGBD, que l@s usuari@s puedan crear consultas ad hoc (o no predefinidas) y complejas.
- El SGBD debe responder rápido a dichas consultas.
- Las consultas se expresan en un lenguaje sencillo pero potente.
- Deben poder consultarse múltiples entidades y con criterios de selección complejos.
- La solución a esta necesidad es el lenguaje de consulta SQL

Flexibilidad e independencia

Independecia Física:

 Se debe poder desconocer las características físicas de la BD con que trabajan (SO, SGBD, índices).

Independencia Lógica:

 Se debe poder hacer cambios en la descripción lógica de la BD sin modificar las aplicaciones que las utilizan (agregar columnas, cambiar tipo de dato).

Problemas de redundancia

- No deben existir datos duplicados (evita incoherencias).
- Si existen datos replicados (por seguridad, fiabilidad, etc), el SGBD es responzable del mantenimiento.
- Si existen datos derivados (o calculados), el SGBD es responzable del mantenimiento.

Integridad de datos

- El SGBD debe asegurar la calidad de los datos en todo el ciclo de vida y ante cualquier inconveniente:
 - Fallas de programa
 - Duplicación de Datos
 - Fallas de HW
 - Fallas de comunicación
 - Fallas en el suminitro eléctrico
 - Fallas de operación humana
 - etc.

Integridad de datos

- Se deben poder definir y hacer cumplir reglas de negocios definidas por la organización, ejemplos:
 - Un cliente no debe poder comprar a crédito, si su saldo es superior a \$ 10.000.
 - No puede emitirse un comprobante de ventas de un producto que no esté en stock.
 - Un vendedor debe pedir autorización antes de emitir una venta por más de \$ 50.000.

- Un SGBD debe permitir que varios procesos accedan al mismo tiempo a los mismos datos.
- Cuando la concurrencia es para lectura:
 - Performance en general: índices, datos precalculados, agregaciones, reescritura de consultas, etc.
 - Buffers
 - Acceso a disco (activación, lectura, etc)

- Cuando el acceso es de **escritura** se deben evitar:
 - Datos incongruentes por escrituras no deseadas
 - Datos incongruentes en lecturas
- Parte de la solución es la Transacción o Unidad de Trabajo (UT):
 Conjunto de operaciones simples que se ejecutan como un todo.
- El SGBD asegura que las UT no se ejecuten parcialmente; se ejecutan completamente o NO se ejecutan.
- Se definen dos comandos COMMIT (confirma la ejecución de la UT) y ROLLBACK retrotrae la UT

Bloqueos (lock):

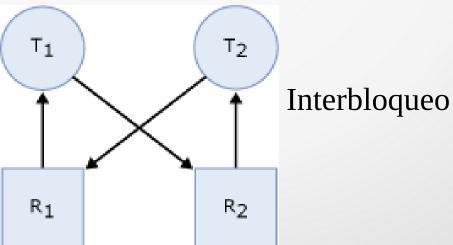
- Qué ocurre cuando hay incongruencias entre UTs concurrentes?
 - Por ejemplo: una UT "A" modifica un dato (digamos un 4 por un 3), otra UT "B" lee ese dato recién modificado (lee un 3), luego "A" realiza un ROLLBACK. Es correcto el dato que tiene "B"?
- Una posibilidad para el caso anterior, es que "A" bloquee el dato hasta finalizar la transacción, "B" quedará esperando, luego "B" leerá un 4. Esta NO es la única posibilidad.
- La idea es aislar un UT del resto, lo que significa poner límites en cuanto a lo que el resto de las UT pueden hacer con el dato actual.

Bloqueos (lock):

 Los SGBD implementan los aislamientos mediante el uso de bloqueos.

 Los bloqueos producen esperas, retenciones de servicio, interbloqueos, entre otros inconvenientes, los SGBD debe poseer herramientas para atacar estos problemas y salir

adelante.



Seguridad

En BDs seguridad se refiere a: confidencialidad, autenticación, autorización, acceso a datos, etc

- **Confidencialidad**: los datos transmitidos por el "Transmisor", solo deben ser accedidos por el "Receptor" definido, y no por terceros. La información solo podrá ser accedida por personas autorizadas a acceder. Suelen usarse técnicas de encriptación, entre otras.
- Autenticación: es un mecanismo que permite asegurar que soy, quien realmente soy en el caso de ser una persona. (Ejemplos: login, lectura de huellas digitales, etc.). En el caso de ser un objeto, se trata de asegurar la procedencia.
- Autorización: es la posibilidad de tener derecho de acceso a objetos a distintos niveles. Requieren de autenticación previa.

Otros Objetivos

Los SGBD se encuentran en constante evolución, las tecnologías de rápido crecimiento establecen los caminos a seguir y las características a soportar, por ejemplo:

- Data Warehouse performance.
- Implementar Diseño Orientado a Objetos.
- Potenciar el tipo de datos "tiempo" agregando características y funcionalidades que permitan utilizarlo con mayor eficiencia en consultas multidimensionales.
- Soporte para servicios web y tecnologías asociadas (xml, xslt, etc).
- Soporte para tecnologías relacionadas con Internet.
- Big Data.

SGBD: Arquitectura de 3 Niveles

Arquitectura de tres niveles

Contamos con 2 niveles de representación informática en una BD:

- Nivel Lógico: Involucra entidades, atributos, reglas de integridad.
- Nivel Físico: Determina cómo se accede físicamente a los datos, cómo y dónde queremos que se agrupen los registros, tamaño que deben ser las páginas, etc.

Por cuestiones de definir un estándar **ANSI para las BDs**, el comité ANSI/SPARC, recomendó que la arquitectura de los SGBD previese tres niveles en vez de dos. Por ello se descompuso el Nivel Lógico en dos: el **Nivel Externo** y el **Nivel Conceptual**; y al que denominabamos Nivel Físico, se lo llamó **Nivel Interno**.

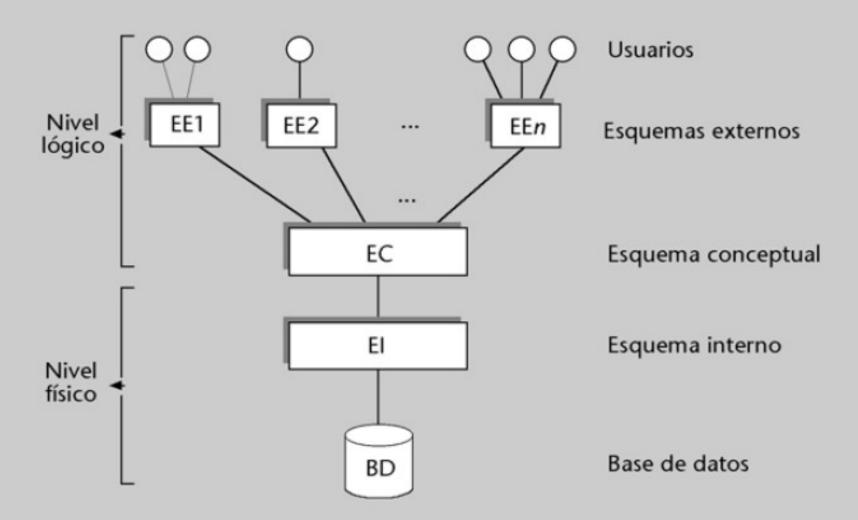
Arquitectura de tres niveles

Los SGBD necesitan que demos una descripción de la BD. Esta descripción recibe el nombre de esquema de la BD. El esquema permite que podamos independizar el SGBD de la BD, es decir, podemos cambiar el diseño sin la necesidad de hacer ningún cambio en el SGBD.

De acuerdo con ANSI/SPARC habría tres niveles de esquemas:

- Esquema Externo
- Esquema Conceptual
- Esquema Interno

Esquemas y niveles



Esquemas

• Esquema Externo: Al definirlo se citaran solo aquellos atributos y entidades que nos interesen, podemos redefinir también una entidad para que parezcan dos, o combinar entidades para que parezcan una sola. Es decir, a través de la definición de Esquemas Externos podemos definir la forma en que determinados usuari@s o programas ven la BD.

Por ejemplo tenemos dos Esquemas Externos ("Autos" y "Camiones"). Puede que nos interese que para los dos esquemas sea visible el atributo "NumeroPatente". Pero que solamente para el esquema "Camiones" sea visible el atributo "CapacidadCarga".

Esquemas

Esquema Conceptual: se describen las entidades tipo, sus atributos, las interrelaciones y las restricciones o reglas de integridad.

El Esquema Conceptual responde a las necesidades de la empresa, y cada aplicación podra tener su visión particular y/o parcial del Esquema Conceptual a través del uso de los Esquemas Externos.

Esquemas

Esquema Interno: Este esquema contendrá la descripción de la organizacion física de la BD, como caminos de acceso, gestión del espacio, tamaño de las páginas en memoria, etc.

El Esquema Interno responde a conceptos relacionados con el rendimiento de la BD, importantes en las decisiones a tomar para hacer el diseño físico de la BD. Si este último esquema NO se define, el SGBD tendrá que decidir los detalles de la organización física, realizando las correspondencias entre los tres niveles de esquemas. SGBD: Flujo de Datos y Control

Flujo de datos y control Programa usuario Área donde se debe poner el resultado Registro Buffers SGBD 2 3 Conceptual Externo Interno Página Base de datos Esquemas

Funciones y Objetivos del Administrador de Bases de Datos (ABD)

FUNCIONES DEL ABD



Comunicación con los Usuarios

- Capacitación continua
 - Resistencia al Cambio
 - + Efectividad
 - + Eficiencia
- ◆ Preparación para el cambio
- Presentaciones a los niveles superiores





Establecer normas y procedimientos

- Normas apuntadas al diseño, a la programación a la operación de la BD
- Procedimientos de copia y restauración, control de hardware, actualización de software, informe de errores, auditoría, etc.
 Presentaciones a los niveles superiores



OBJETIVOS DEL ABD



Integridad de la BD

"Realizar operaciones correctas y mantener la consistencia"

- ◆ Restricciones (1 operación)
- ◆ Transacciones (1 serie de operaciones)
- ◆ Control de concurrencia (bloqueo)



Seguridad en la BD

"Limitar solo a operaciones permitidas"

- ♦ Autenticación
- Autorización y Vistas
- ◆ Encriptación



Recuperación de la BD

"Restaurar la BD ante una falla de cualquier índole a su operación normal, y con la menor pérdida de datos posible"

OBJETIVOS DEL ABD



Rendimiento

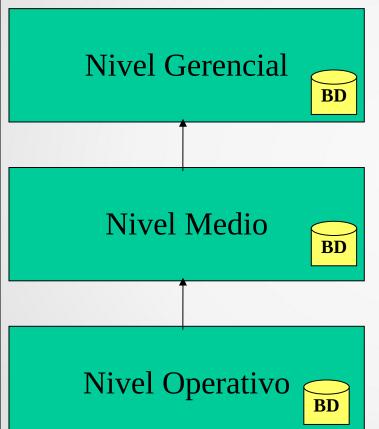
"Realizar operaciones y que tiendan a mejorar y/o mantener el rendimiento esperado del SGBM"

- ◆ Creación de índices
- ◆ Datos precalculados
- ◆ Compra de Hardware (discos, memoria, etc)
- ♦ Mejoramiento de algoritmos

Se puede hacer a 3 niveles:

- 1. Niveles de usuario
- 2. Unidad operativa
- 3. Zonas o localidades geográficas

Niveles de usuario

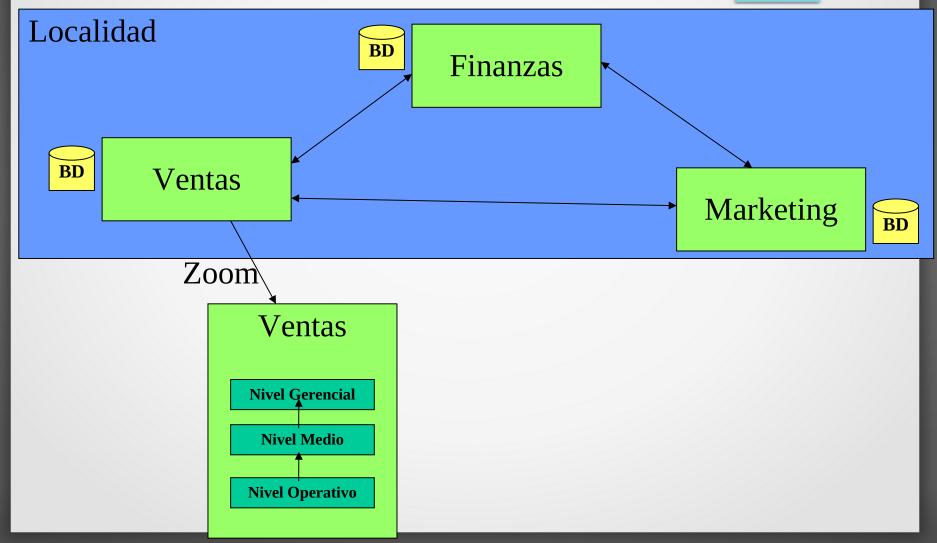


- •Manejo de información para la toma de decisiones
- Datos resumidos (propios + externos)
- Tendencias

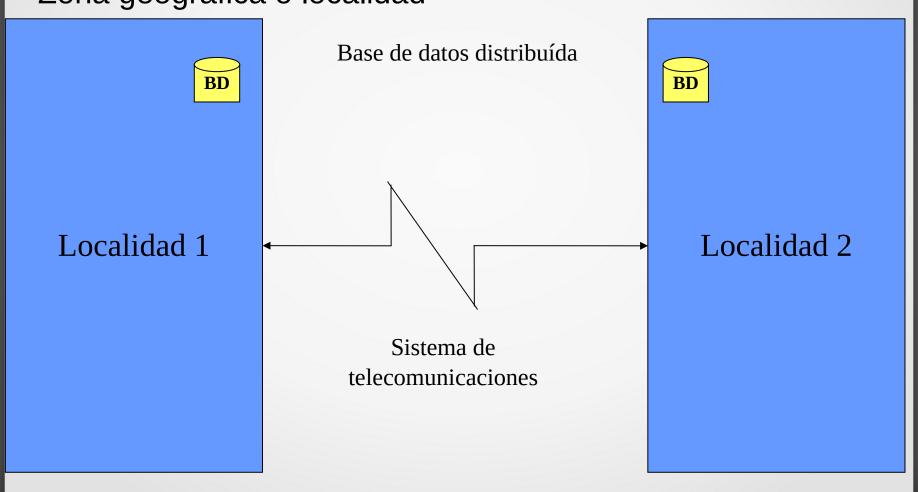
- •Manejo de información para decisiones a corto plazo
- •Listados y reportes elavorados a partir del nivel operativo

- Transacciones diarias
- •Gran cantidad de datos

Unidad operativa



Zona geográfica o localidad



Riesgos y costos de las BD

"No todo lo que brilla es oro"

- Conflictos en las organizaciones.
 - Ciertos usuarios pueden no estar de acuerdo en compartir la información (las famosas "quintitas"), esto complica la integración.
- Fracasos en el desarrollo de proyectos.
 - Dirigentes que no están convencidos del valor del sistema de BD.
 - Los proyectos suelen ser muy largos.
 - El personal clave puede abandonar de repente la compañía.

Riesgos y costos de las BD

- Malfuncionamiento del sistema.
 - Cuando no esté operativo, casi todo se para, esto ocasiona altos costos. Esto suele evitarse con BD distribuidas, que por otro lado son costosas.
- Costos imprevistos.
 - Una BD puede requerir altos costos de HW y SW.
 - El SGBD puede ser muy costoso.
- Necesidad de personal.
 - La administración de una BD requiere de personal altamente capacitado y por ende costoso.

Desarrollo de la BD

- Diseño tradicional:
 - Orientado a funciones.
 - Los usuarios tienden a requerir nuevas funciones.
 - El descubrimiento de nuevos datos hace que se creen nuevas funciones.
 - Los datos son más importantes que las funciones.

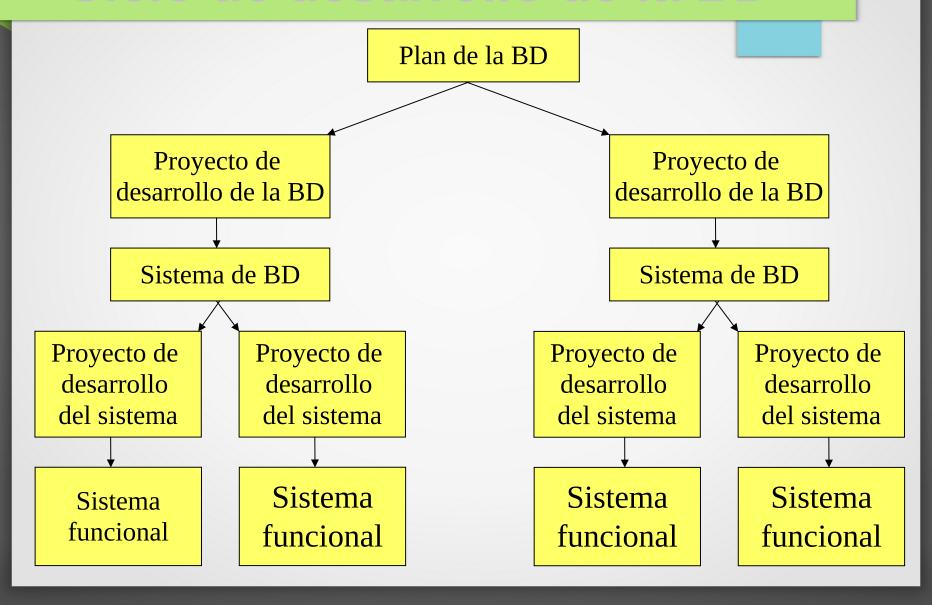


Ciclo de desarrollo de la BD

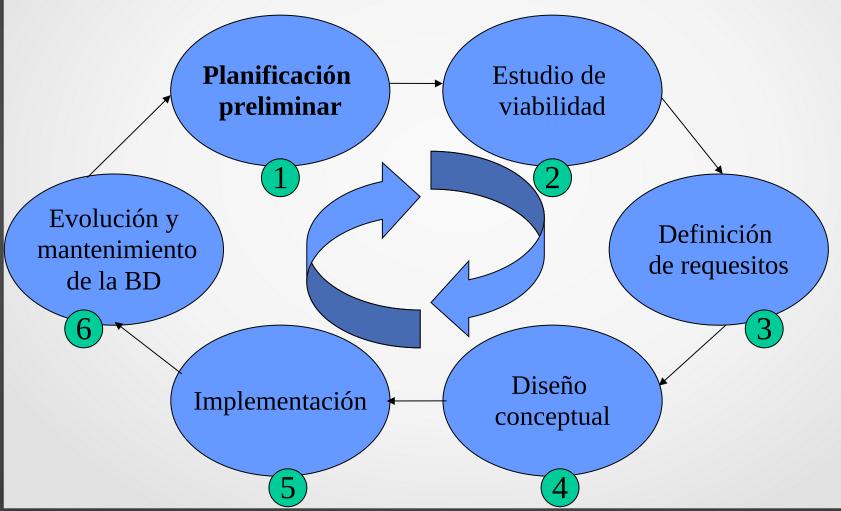
- Diseño de BD:
 - Orientado a datos.
 - Los datos son utilizados por las funciones.
 - Los datos son más estables que las funciones.
 - Pueden combinarse de muchas maneras para dar solución a múltiples necesidades.
 - Una vez creado un buen esquema de datos,
 pueden crearse los sistemas funcionales basados en el, estos serán más productivos y adaptables.



Ciclo de desarrollo de la BD



(CVDBD) consta de 6 etapas:



1) Planificación preliminar:

- Tiene lugar durante el proyecto de planificación estratégico de la BD.
- Asegura que se tendrán en cuenta las necesidades a largo plazo.
- Responde preguntas como:
 - Qué aplicaciones se utilizan y cuáles son sus funciones?
 - Qué archivos están asociados a cada una de las aplicaciones?
 - Se están creando nuevas aplicaciones, archivos nuevos?

2) Estudio de viabilidad:

Se compone de:

- a) Viabilidad tecnológica:
 - Existe y está disponible la tecnología adecuada para dar soporte al desarrollo de la BD?
- b) Viabilidad operacional:
 - La organización dispone de los RRHH y presupuesto para que el sistema de BD tenga éxito?
- c) Viabilidad económica:
 - Se pueden identificar los beneficios?
 - Los beneficios costean el sistema?
 - Se pueden medir los costos y beneficios?

3) Definición de requisitos:

Involucra la definición del alcance de la BD:

- Requisitos de información de áreas funcionales y administrativas.
 - Forma de recolección: formularios, encuestas, cuestionarios, entrevistas con los distintos niveles, etc.
- Requisitos de HW y SW: dependen directamente del nivel de rendimiento que deban satisfacer.

4) Diseño conceptual:

Es el esquema conceptual de la BD:

- Las especificaciones no deben dejar dudas a la futura implementación.
- Se crean los modelos detallados de las vistas de usuario

5) Implementación:

- 1. Selección de un SGBD.
- 2. Convertir el modelo conceptual en modelo de implementación, (Relacional por ejemplo) tiene directamente que ver con el SGBD seleccionado.
- 3. Construir el diccionario de datos (seguridad, acceso, controles, etc).
- 4. Poblar la BD.
- 5. Desarrollar los programas de aplicación.
- 6. Formación de los usuarios.

6) Evolución y mantenimiento de la BD:

- Entrevistas con los usuarios para determinar datos omitidos y nuevas expectativas.
- Afina sucesivamente el sistema, con el tiempo tiende a estabilizarse.
- A medida que las necesidades del negocio cambian las implantamos en el sistema de BD.

Modelos de Bases de Datos

Modelos

- Los modelos son representaciones simplificadas de la realidad. Las BD representan una parte de la realidad, por tanto son modelos
- Un modelo de BD es el conjunto de componentes o herramientas conceptuales que un SGBD proporciona para implementar una BD.
- Los modelos más conocidos son: jerárquico, en red, relacional y relacional con objetos.

Herramientas de los Modelos de BD

- 1) Estructuras de Datos: diseñar tablas, árboles, etc.
- 2) Restricciones o reglas de Integridad: dominios, claves, relaciones, etc.
- Operaciones para trabajar con datos: por ejemplo seleccionar filas en un modelo relacional, buscar hijos de un nodo en un modelo jerárquico, etc.

Evolución de los Modelos de Bases de Datos

Evolución: modelo Jerárquico

- Nace en los años 60s.
- La primera implementación fue IMS (Information Management System) de IBM para el programa Apollo de la NASA.
- Caraterísticas:
 - Las referencias se definen al nivel físico, direcciones físicas (no existe una vista lógica y física).
 - Los datos se almacenan en registros.
 - Los registros con los mismos campos forman parte de un fichero (*record type*).
 - Las relaciones son unidireccionales hijo-padre N:1.

Evolución: modelo Jerárquico

Ventajas:

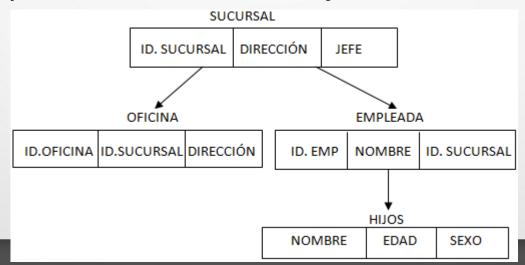
- El acceso a los datos es muy rápido, ya que los nodos se apuntan mediante direcciones físicas.
- Responde a ciertas preguntas de índole jerárquico, de manera muy rápida.

Desventajas:

- La lectura se realiza desde los nodos hoja hacia arriba.
- No responde a preguntas de forma flexible.
- No hay relación padre-hijo, solo hijo-padre
- No se soportan las relaciones N:M, se simulan con varias
 1:N

Evolución: modelo Jerárquico

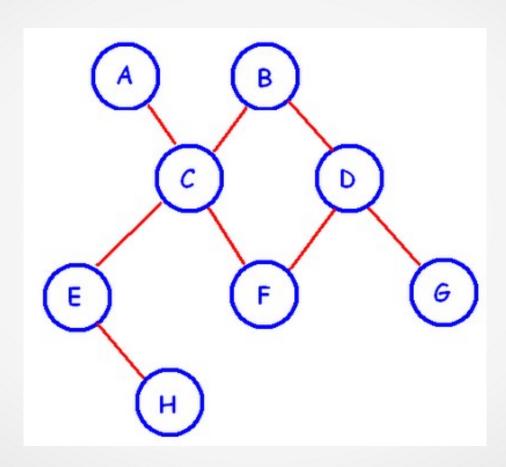
- Desventajas: (continuación...)
 - No implementa ningún control sobre datos propios, la responsabilidad es de las aplicaciones.
 - No se garantiza la inexistencia de registros duplicados.
 - No existe integridad referencial. Ej: puede borrarse un nodo padre, sin borrar los hijos.



Evolución: modelo En Red

- Surgen a principios de los años 70s.
- Mejoró al modelo jerárquico, un nodo podía estar conectado con más de un nodo a la vez.
- Caraterísticas:
 - Un tipo de registro representa un nodo.
 - Un elemento es un campo de datos dentro de un registro o nodo.
 - Un agregado de datos es un conjunto de datos con nombre.
 - Se pueden establecer relaciones N:M.
 - Los enlaces son a nivel físico.
 - El resto de las características son similares al modelo jerárquico.

Evolución: modelo En Red



Evolución: modelo Relacional

- En los años 80s, surgieron muchos SGBD que implementaron el modelo relacional propuesto por el Dr. E. F. Codd de IBM, la mayoría utilizaba SQL como lenguaje nativo.
- Se tratará este modelo con detalle en próximos encuentros.

Evolución: modelo Relacional con Obj

- Incorporan la posibilidad de crear tipos abstractos de datos, y el control estricto de tipado.
- Acercan a los SGBDR a la OO.
- Los primeros en dar este soporte fueron: IBM Informix
 V9 (1996), IBM DB2 UDB V5 (1996) y Oracle V8 (1997)
- Permiten crear estructuras como clases y atributos, también pueden definirme los métodos que proporcionarán el dinamismo y el control de datos. Se puede implementar el encapsulamiento.

Modelos de BD

