

# ADMINISTRACIÓN DE SISTEMAS DE GESTIÓN DE BASES DE DATOS

---

Curso 2019 - 2020

I.E.S. Marcos Zaragoza

2º CFGS ASIX

Alberto Alemany

# UNIDAD 1

---

Introducción a la administración de bases de datos

# Índice

## 1. Sistemas Gestores de Bases de Datos (SGBD)

1. BD vs SGBD
2. Objetivos / Características
3. Arquitectura
4. Funciones
  1. Definición
  2. Manipulación
5. Componentes
  1. Usuarios
  2. Componentes técnicos
    1. Funcionales
    2. Implementación física del sistema
6. Arquitecturas operacionales
7. Tipos de Bases de Datos
8. SGBD comerciales y libres

## 2. Administrador de las bases de datos (DBA)

1. Funciones
2. Herramientas

## 3. Lenguajes

1. Lenguaje de manipulación
2. Lenguaje de definición
3. Lenguaje de control

## 4. Extensiones procedurales

1. Lenguajes procedurales

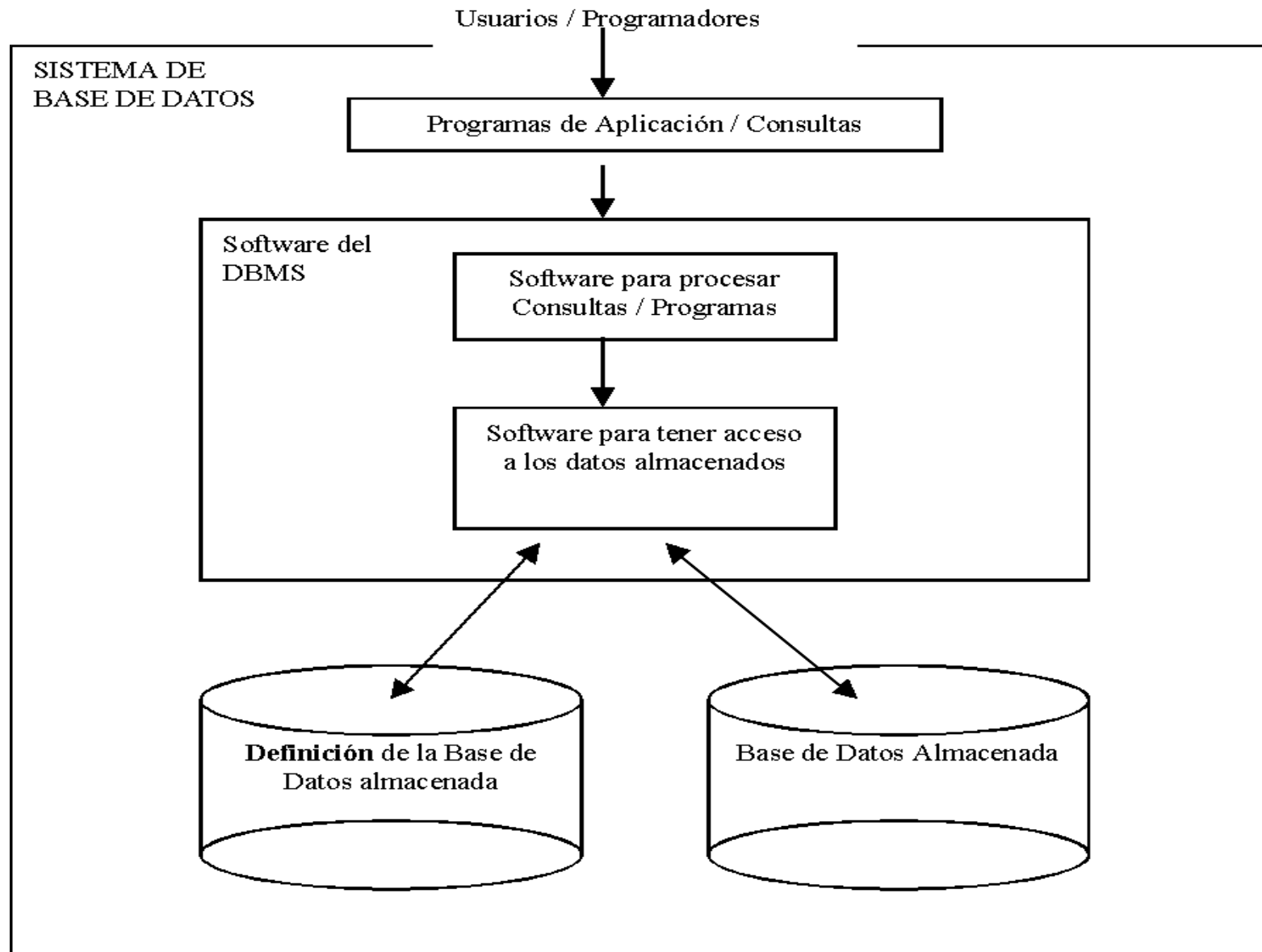
# 1. Sistemas Gestores de Bases de Datos (SGBD)

- El objetivo primordial de un gestor es proporcionar eficiencia y seguridad a la hora de extraer o almacenar información en las BD.
- Los sistemas gestores de BBDD están diseñados para gestionar grandes bloques de información, que implica:
  - La definición de estructuras para el almacenamiento.
  - Mecanismos para la manipulación de la información

# 1.1. BD vs SGBD

- BD: Conjunto de datos relacionados y organizados con cierta estructura. Información relevante de la empresa.
- SGBD: Aplicación que permite definir, crear y mantener la BD. Interfaz entre el Usuario y la BD.
- Sistema BD=BD+SGBD





## 1.2. Objetivos/Características de un SGBD

- **Independencia física y lógica.** Estructuras físicas independientes de la lógica y viceversa, acceso a datos sin necesidad de conocer la estructura interna.
- **Eficaz acceso a los datos:** Sin necesidad de conocer como están guardados, facilidad de manipulación (si está autorizado) sin necesidad de ser Informático. (Herramientas gráficas)
- **Mantener la integridad y consistencia** Garantizar que la transacción se haga o se rechace (Atomicidad)
- **Permiten la concurrencia.** Acceso compartido a la BD, controlando la interacción entre usuarios concurrentes. Que no se enteren que ambos están a la vez.
- **Mecanismos de respaldo y recuperación** para restablecer la información en caso de fallos en el sistema
- **Coherencia de datos: Cumplir restricciones.** Reglas de la realidad
- **Redundancia controlada**
- **Administración centralizada. Herramientas de administración.**
- **Seguridad de los datos:** Acceso controlado a los datos de la BD mediante mecanismos de seguridad de acceso a los usuarios.
- **Uso de transacciones**

# Concepto de Transacción

- Una transacción es un conjunto de instrucciones que realizan una acción y tiene las características ACID (Atomicity, Consistency, Isolation and Durability):
  - **Atomicidad**: es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
  - **Consistencia**: *Integridad*. Es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.
  - **Aislamiento**: es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información sean independientes y no generen ningún tipo de error.
  - **Durabilidad**: es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

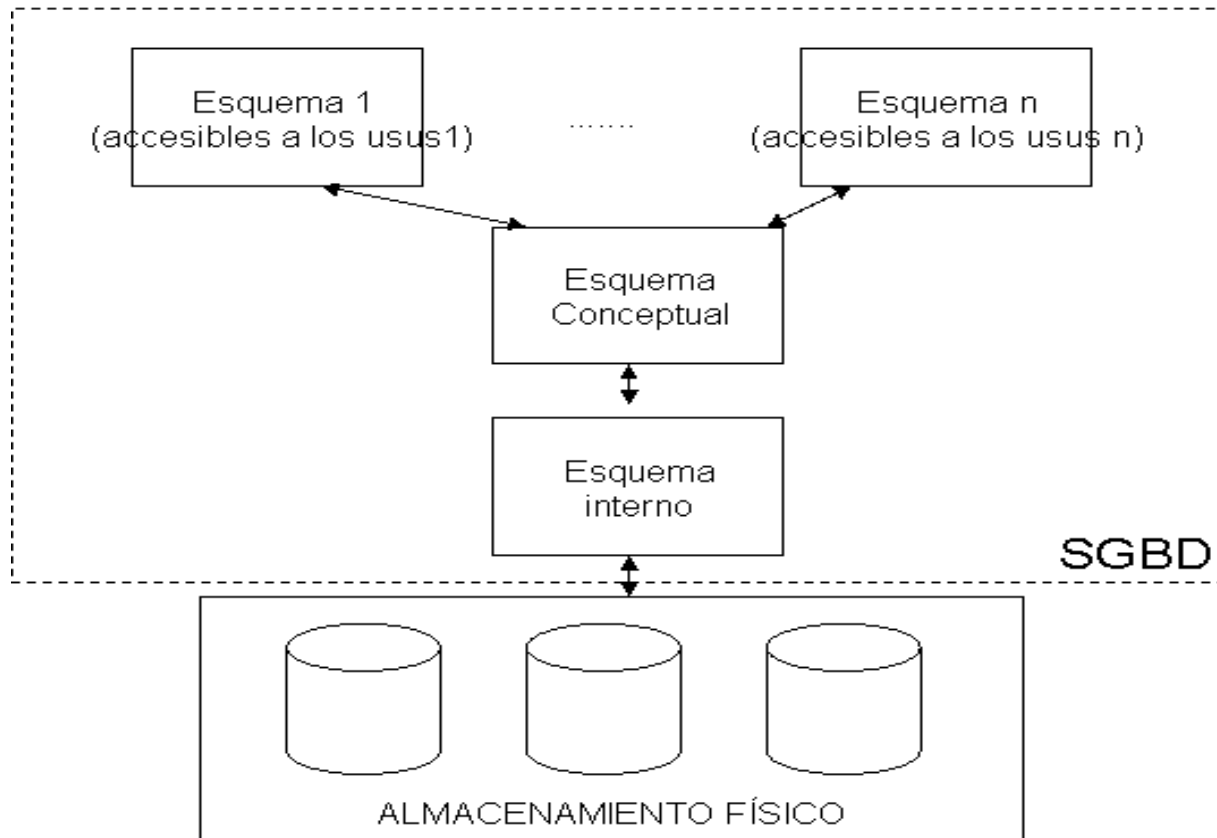


## 1.3. Arquitectura SGBD

- Se basa en la ANSI-SPARC que divide en 3 niveles:
  - **Nivel Interno:** El más cercano al almacenamiento físico, es decir, tal y como están almacenados los datos en el ordenador. (Archivos, tipos registro, longitud...).
  - **Nivel Conceptual o lógico:** Esquema conceptual. (Entidades, relaciones, atributos...), define qué datos hay almacenados y cómo se relacionan.
  - **Nivel Externo:** Representación de los datos y donde se encontrarán la vista del usuario y sus permisos. Es el más cercano a los usuarios. Vistas de partes de la BD, bien para restringir o simplificar.

## 1.3. Arquitectura SGBD

- Se basa en la ANSI-SPARC que divide en 3 niveles:



## 1.3. Arquitectura SGBD: Ej con Modelo Relacional

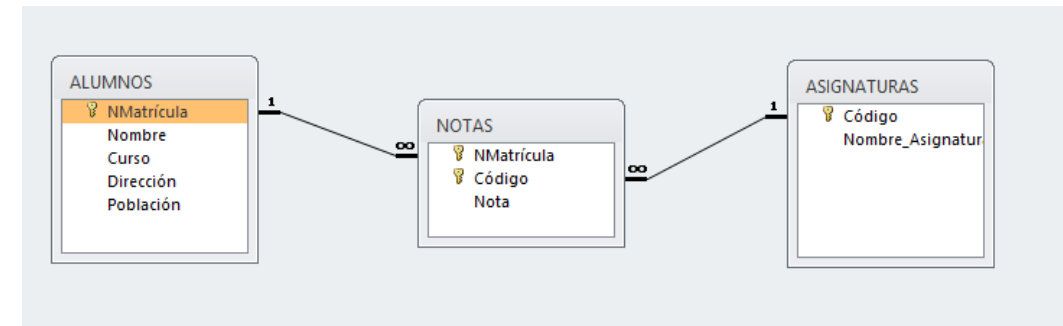
Curso	Nombre	Nombre de asignatura	Nota
1	Ana	Programación en lenguajes estructurados	6
1	Ana	Sistemas informáticos multiusuario y en red	8
2	Rosa	Desa. de aplic. en entornos de 4.ª Generación y H. Case	5
2	Juan	Desa. de aplic. en entornos de 4.ª Generación y H. Case	7
1	Alicia	Programación en lenguajes estructurados	5
1	Alicia	Sistemas informáticos multiusuario y en red	4

- **Nivel externo:** Visión parcial de las tablas de la BD según el usuario. Por ejemplo, la vista que se muestra en la Tabla, obtiene el listado de notas de alumnos con los siguientes datos: Curso, Nombre, Nombre de asignatura y Nota.

## 1.3. Arquitectura SGBD: Ej con Modelo Relacional

- *Tabla ALUMNOS.* **Columnas:** NMatrícula, Nombre, Curso, Dirección, Población. **Clave:** NMatrícula. Además tiene una relación con NOTAS, pues un alumno puede tener notas en varias asignaturas.
- *Tabla ASIGNATURAS.* **Columnas:** Código, Nombre de asignatura. **Clave:** Código. Está relacionada con NOTAS, pues para una asignatura hay varias notas, tantas como alumnos la cursen.
- *Tabla NOTAS.* **Columnas:** NMatrícula, Código, Nota. Está relacionada con ALUMNOS y ASIGNATURAS, pues un alumno tiene notas en varias asignaturas, y de una asignatura existen varias notas, tantas como alumnos.

- Podemos representar las relaciones de las tablas en el nivel lógico como en la Figura:



- **Nivel lógico o conceptual:** Definición de todas las tablas, columnas, restricciones, claves y relaciones. En este ejemplo, disponemos de tres tablas que están relacionadas

## 1.3. Arquitectura SGBD: Ej con Modelo Relacional

- **Nivel interno:** En una BD las tablas se almacenan en archivos de datos de la BD. Si hay claves, se crean índices para acceder a los datos, todo esto contenido en el disco duro, en una pista y en un sector, que sólo el SGBD conoce. Ante una petición, sabe a qué pista, a qué sector, a qué archivo de datos y a qué índices acceder.

# Pasos Entre niveles

1. El Usuario solicita unos datos y crea una consulta
2. El SGBD verifica y acepta el esquema externo para ese usuario.
3. Transforma la solicitud al esquema conceptual
4. Verifica y acepta el esquema conceptual
5. Transforma la solicitud al esquema Interno.
6. Selecciona la o las tablas implicadas en la consulta y
  - la ejecuta
7. Transforma del esquema interno al conceptual, del conceptual al externo
8. Finalmente, el usuario ve los datos solicitados

# Que consigue esta arquitectura

- **Independencia lógica:** la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se podrá modificar el esquema conceptual para ampliar la BD o para reducirla, por ejemplo, si se elimina una entidad, los esquemas externos que no se refieran a ella no se verán afectados.
- **Independencia física:** la capacidad de modificar el esquema interno sin tener que alterar ni el esquema conceptual, ni los externos. Por ejemplo, se pueden reorganizar los archivos físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización, o se pueden añadir nuevos archivos de datos porque los que había se han llenado. La independencia física es más fácil de conseguir que la lógica, pues se refiere a la separación entre las aplicaciones y las estructuras físicas de almacenamiento.
- Las correspondencias entre niveles requiere ampliar el diccionario de datos

## 1.4. Funciones del SGBD

- La función principal: Permitir a los usuarios las 4 operaciones fundamentales:
  - Creación e inserción
  - Consulta
  - Actualización
  - Borrado
- Con 2 objetivos:
  1. proporcionándoles una visión abstracta de los datos (esconder como se almacenan y mantienen los datos)
  2. Velocidad (buen tiempo de respuesta) y seguridad (que sea verdad lo que resulta).



## 1.4.1. Función de definición

- Permite especificar:
  - Elementos de los datos que integran la BD.
  - Estructura
  - Relaciones entre esos elementos
  - Reglas de integridad
  - Controles de autorización
  - Características físicas
- Utiliza el lenguaje **DDL (Data Definition Language)**

## 1.4.1. Función de definición

- Lenguaje SQL de definición (DDL vale para cada nivel)
- Nivel Interno:
  - Espacio físico, Longitud de campos, Modo de representación, Caminos de acceso, punteros, índices
- Lenguaje SDL (Storage Definition Language)
- Nivel Conceptual
  - Definición de entidades, atributos y relaciones, Autorización de accesos
- Nivel Externo
  - Construcciones gráficas de tablas y relaciones (Access) o programa de diseño
- Lenguaje VDL (View Definition Language)
- Todo se almacena en el Diccionario de datos

## 1.4.2. Función de manipulación

- Permite buscar, añadir, suprimir o modificar los datos de una BD siempre de acuerdo a las especificaciones y restricciones de seguridad.
- **Utiliza el lenguaje DML (Data Manipulation Language).**
  - Nivel Interno: Algoritmos eficientes de acceso a los datos.
  - Nivel Conceptual: Abstracción de la parte interna, construcción de consultas o actualizaciones. Vistas a programadores
  - Nivel externo: Vistas a usuarios. Facilidad de uso con herramientas gráficas.

## 1.5. Componentes de un SGBD

- Componentes humanos. USUARIOS.
- Componentes técnicos
  - Componentes Funcionales
    - Componentes de procesamiento de consultas
    - Componentes de Gestión de almacenamiento
  - Implementación Física del sistema
    - Archivos de datos
    - Diccionario de datos

## 1.5.1. Usuarios

- **Administrador.** Diseña la base de datos, definiendo su esquema y la administra, pudiendo usar todas las herramientas administrativas y encargándose de conceder la autorización para el acceso a datos.
- **Programadores de aplicaciones.** Profesionales informáticos que interactúan con el sistema a través del DML. Utilizan lenguaje embebido (sentencias de descripción, manipulación y control del SGBD dentro del lenguaje anfitrión).
- **Usuarios sofisticados.** Se comunican con el sistema mediante consultas en lenguaje de consulta de base de datos (instrucciones DML).
- **Usuarios finales.** Interactúan con el sistema mediante alguna aplicación que emplea internamente estas acciones sobre el sistema. Ej: Compra online de una camisa.

## 1.5.2. Componentes técnicos

- 4.5.2.1 Componentes Funcionales
- 4.5.2.2 Implementación Física del sistema

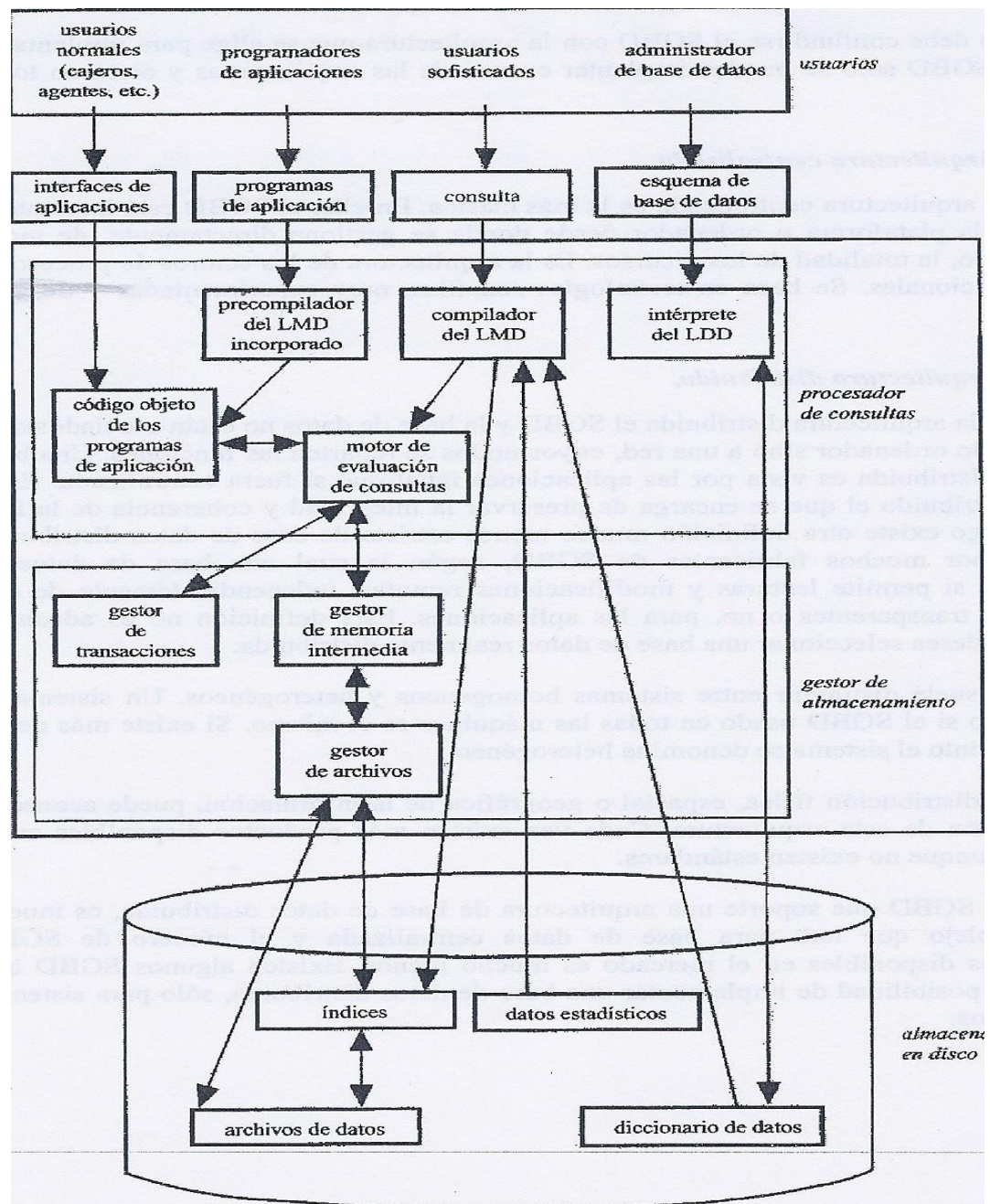
## 1.5.2.1 Componentes Funcionales

- **Componentes de procesamiento de consultas**
  - Compilador DML: Traduce a instrucciones que entiende el motor
  - Precompilador DML incorporado: Traduce a lenguaje objeto
  - Intérprete DDL: Interpreta instrucciones DDL y las introduce en el diccionario
  - Motor de evaluación de Consultas: Ejecuta las instrucciones generadas por el compilador DML
- **Componentes de Gestión de almacenamiento (Interfaz de bajo nivel con los programas de aplicación)**
  - Gestor de autorización e integridad.
  - Gestor de transacciones: Asegura la atomicidad de transacciones.
  - Gestor de archivos: Gestiona la reserva de espacio
  - Gestor de memoria intermedia: Trae los datos del disco a memoria principal. Gestiona la caché.

## 1.5.2.2 Implementación Física del sistema

- Archivos de datos: Almacena la base de datos en sí.
- Diccionario de datos: Almacena metadatos acerca de la estructura de los mismos.
- Índices: Proporciona acceso rápido a los datos.
- Datos estadísticos: Almacena datos estadísticos sobre los datos. El procesador de consultas usa esta información para planificarlas.



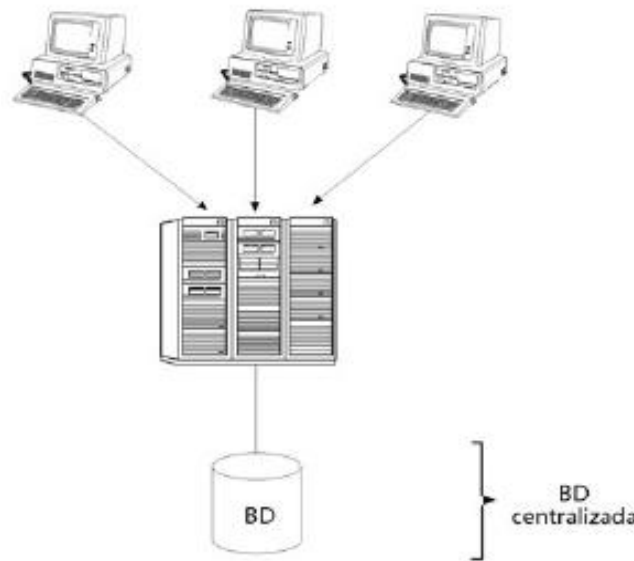


## 1.6. Arquitecturas operacionales

- Arquitecturas centralizadas.
- Arquitecturas cliente/servidor.
- Arquitecturas distribuidas.
- Arquitecturas en la nube.

## 1.6. Arquitecturas operacionales: Centralizada

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



## 1.6. Arquitecturas operacionales: Cliente / Servidor

- Establece una comunicación entre un denominado cliente (ordenador, aplicación o usuario) y el servidor, que es el que le responde a las solicitudes realizadas.
- Es importante que el SGBD soporte sistemas de comunicación normalizados, ya que tendrá que recibir peticiones de diversos clientes, operando con máquinas y protocolos distintos.
- Casi la mayoría de SGBD son de este tipo (MySQL, Oracle, PostgreSQL, Microsoft SQL Server, etc)



## 1.6. Arquitecturas operacionales: Distribuida

- En la arquitectura distribuida el SGBD y la base de datos no están asociados a un determinado ordenador sino a una red, cuyos nodos se reparten las funciones.
- Es el SGBD distribuido el que se encarga de preservar la integridad y coherencia de la BD, externamente es vista como centralizada.
- Un sistema es homogéneo si el SGBD usado en todas las máquinas es el mismo. Si existe más de un SGBD distinto el sistema se denomina heterogéneo.
- La distribución física, espacial o geográfica de la información, puede aconsejar la utilización de esta arquitectura.
- Es muy compleja de gestionar y tan solo existen algunos SGBD que lo soportan, y solo para homogéneas.

## 1.6. Arquitecturas operacionales: En la nube

- Arquitectura según la cual la base de datos es ejecutada en una plataforma del cloud. Los dos métodos primarios para ejecutar esta BD en el cloud son:
  - Imágenes de máquinas virtuales
  - Database as a Service (DBaaS). Se ofrece la base de datos como un servicio. Este modelo proporciona acceso a la base de datos vía la nube sin necesidad de disponer de hardware (máquinas y servidores), instalar software y configurar el rendimiento. El proveedor se encarga de mantener toda esta maquinaria y cobrar un precio, que varía entre distintos modelos.
- Ventajas como la Escalabilidad.
- Desventajas como la **falta de flexibilidad** o el **tratamiento de datos** en una plataforma de un tercero.
- Líderes dentro de este sector de DBaaS son: Microsoft (Azure SQL Database), Google (Google Cloud Spanner), Amazon Web Services (Amazon DynamoDB o Amazon Aurora), MongoDB y alguno nuevo como Alibaba (Alibaba Cloud ApsaraDB)

## 1.7. Tipos de Base de Datos

- Bases de datos jerárquicas
- Bases de datos en red
- Bases de datos relacionales
- Bases de datos orientada a objetos
- Bases de datos NoSQL (no relacionales)

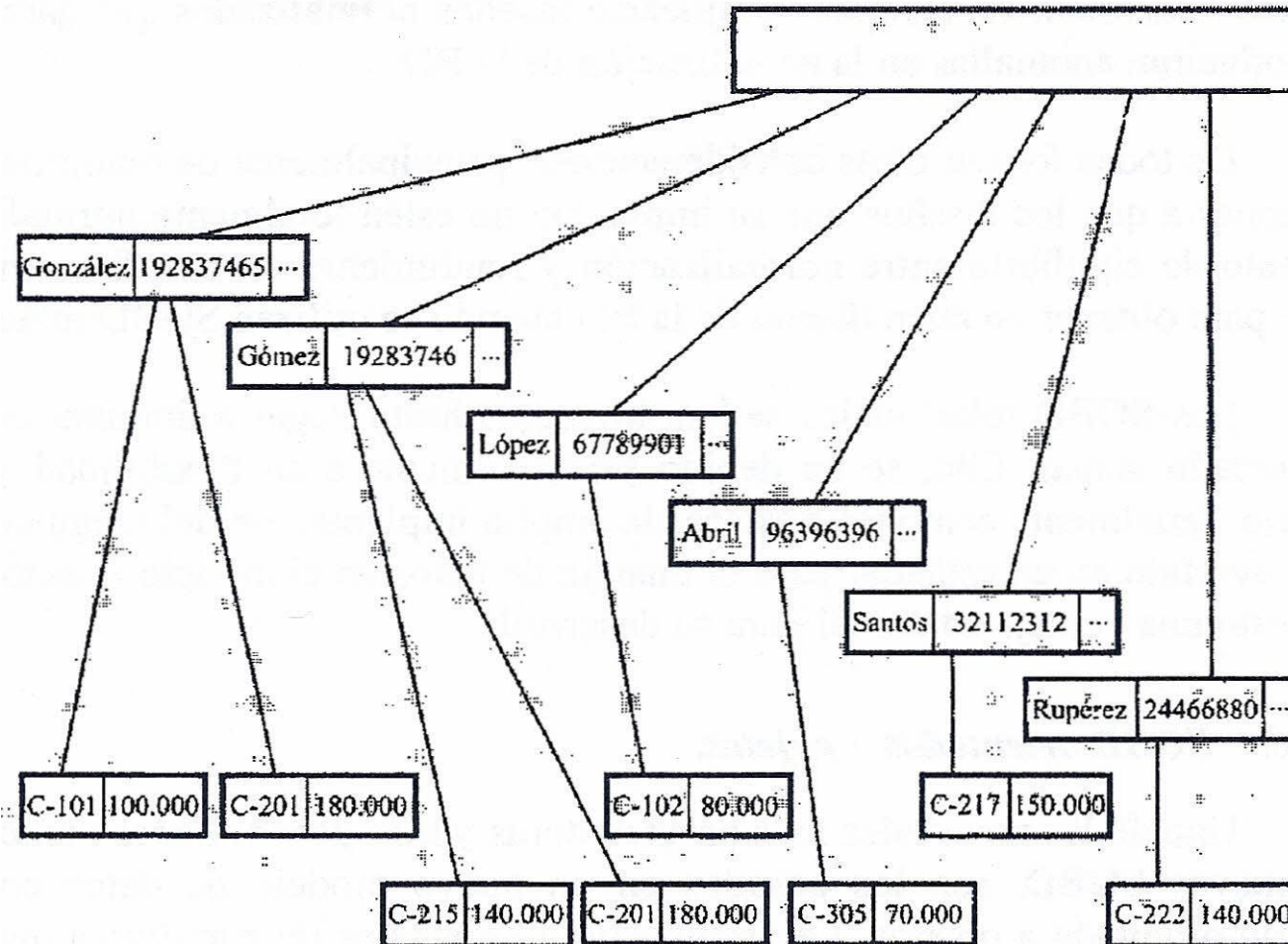


## 1.7.1. Bases de datos jerárquicas

- Consiste en un conjunto de registros conectados entre sí mediante enlaces.
- Estos enlaces son punteros.
- Tiene una estructura jerárquica y es unidireccional de Padre a Hijo, nunca al revés, lo que dificultaba su gestión y representación de la realidad.
- No disponía de relaciones M:M, poco flexible.

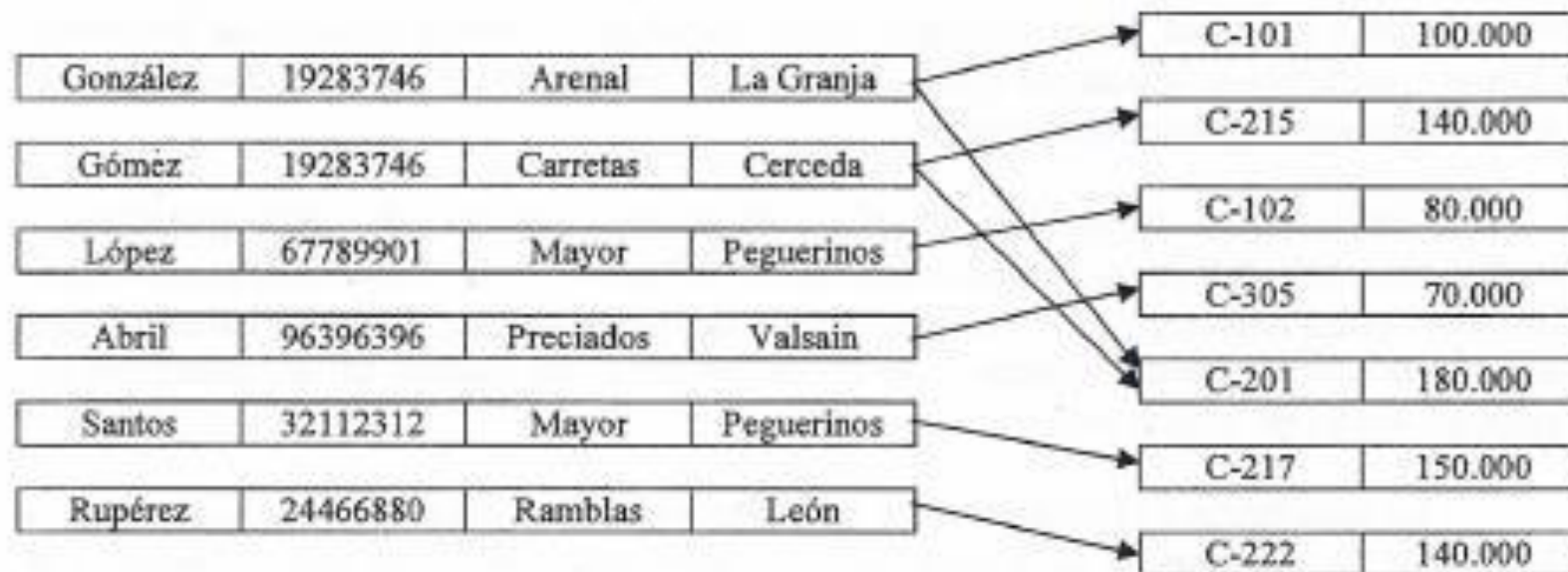


## 1.7.1. Bases de datos jerárquicas



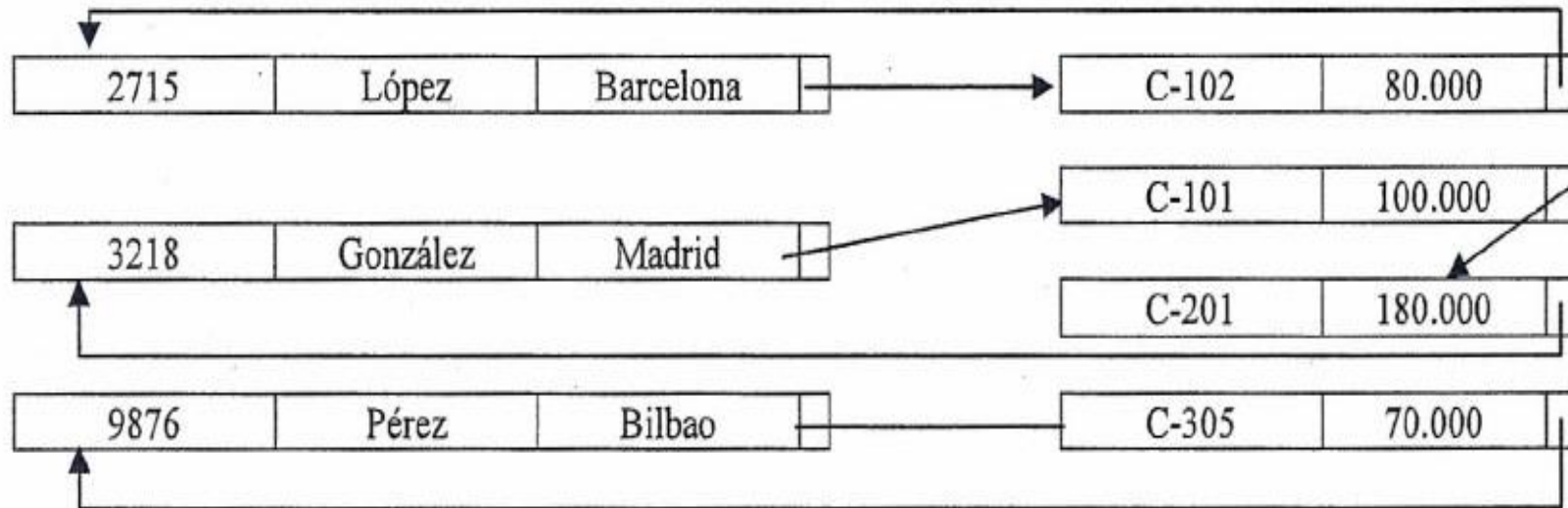
## 1.7.2. Bases de datos en red

- Parecido al jerárquico, conjunto de registros enlazados mediante puntero, pero esta vez en forma de grafo.
- Al menos teóricamente sí disponía de comunicación bidereccional, pero en la práctica era demasiado costoso y lo hacía con varios punteros. Simulaba la M:M con varias 1:M



## 1.7.2. Bases de datos en red

En la siguiente figura se muestra la estructura de anillo del primer ejemplo.



## 1.7.3. Bases de datos relacionales

- El más extendido y comercialmente el único con éxito, los datos se describen como relaciones que se suelen representar como tablas bidimensionales consistentes en filas y columnas.
- Cada fila (tupla, en terminología relacional) representa una ocurrencia.
- Las columnas (atributos) representan propiedades de las filas.
- Cada tupla se identifica por una clave primaria o identificador.
- Los enlaces no son punteros son los mismos datos

## 1.7.3. Bases de datos relacionales

Nombre	Dni	Calle	Ciudad	Número-cuenta
González	19283746	Arenal	La Granja	C-101
Gómez	19283746	Carretas	Cerceda	C-215
López	67789901	Mayor	Peguerinos	C-102
Abril	96396396	Preciados	Valsain	C-305
González	19283746	Arenal	La Granja	C-201
Santos	32112312	Mayor	Peguerinos	C-217.
Rupérez	24466880	Ramblas	León	C-222
Gómez	19283746	Carretas	Cerceda	C-201

Número-cuenta	Saldo
C-101	100.000
C-215	140.000
C-102	80.000
C-305	70.000
C-201	180.000
C-217	150.000
C:Z22--	-

## 1.7.3. Bases de datos orientadas a objetos

- Una de las más novedosas.
- Basado en una colección de objetos (como c++), cada objeto puede tener sus propios procedimientos, única forma de acceder o manipularlos.
- Manejan objetos complejos, y son más sencillos de mantener que los relacionales
- Aún no está suficientemente desarrollado, y comercialmente su salida es muy limitada, aún no son muy estables.

## 1.7.4. Bases de datos NoSQL

- Es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico del sistema de ASGBD relacionales.
- No usan SQL como el principal lenguaje de consultas.
- Los datos almacenados no requieren estructuras fijas como tablas, no soportan operaciones JOIN, ni garantizan completamente ACID.
- Escalan bien horizontalmente.

## 1.7.4. Bases de datos NoSQL

- La aparición de la web, el software como servicio, los servicios en la nube y las startups de éxito, facebook, google..., con millones de usuarios, y con todo ello llegaron los problemas de alta escalabilidad.
- Los sistemas NoSQL intentan atacar este problema proponiendo una estructura de almacenamiento más versátil aunque sea a costa de consistencia y funcionalidad.
- Algunas implementaciones bien conocidas que podríamos como NoSQL son: CouchDB, MongoDB, RavenDB, Neo4j, Cassandra, BigTable, Dynamo, Riak, Hadoop, y otras muchas.



## 1.8. SGBD comerciales y libres

- Con Internet, el software libre se ha consolidado como alternativa, técnicamente viable y económicamente sostenible al software comercial, y cada vez más barato.
- En ofimática tenemos Open office, LibreOffice o Microsoft Office.
- También disponemos SGBD a un nivel de propósito general como MySQL, SQL Server y si hablamos de más potencia y funcionalidad Oracle.
- No conviene decantarse por uno en concreto, debemos usar en primer lugar lo que mejor nos funciona dadas nuestras restricciones particulares.

## 1.8. SGBD comerciales y libres: ¿Cuál elegir?

- En cuanto a si debe ser libre o no la decisión dependerá de si disponemos de personal cualificado en cuyo caso un sistema libre es más barato y potente.
- Por otro lado, los sistemas libres cada vez proveen una mejor y más eficiente documentación tanto a nivel oficial como a través de múltiples foros y *blogs*, lo que hace que cada vez lo use más gente y mejore continuamente. Sin embargo siempre existe el riesgo de que sea comercializado y deje de estar disponible de forma abierta

## 1.8. SGBD comerciales y libres: ¿Cuál elegir?

- No conviene decantarse por uno en concreto, debemos usar en primer lugar lo que mejor nos funciona dadas nuestras restricciones particulares.
- Una aplicación de agenda puede ser perfectamente válido un Access o incluso un Openoffice calc (y en según qué casos incluso un *Bloc de notas*).
- Si por el contrario, mi base de datos requiere cierta cantidad de accesos de usuarios diversos, control de integridad y otras funcionalidades, debería plantearme algo como MySQL.
- Finalmente, si hablamos de una gran corporación que requiere herramientas avanzadas para grandes bases de datos, servicio técnico, operaciones críticas, quizás debamos plantearnos sistemas más potentes como Oracle.

## 2. Administrador de Bases de Datos (DBA)

- El DBA (Data Base Administrator) será el responsable del diseño, control y administración de la base de datos, y es una función que puede ser desempeñada por una persona o por un grupo de personas, dependiendo de la envergadura del proyecto y del marco organizativo de la institución

## 2.1. Funciones principales del DBA

- **Estructura de la Base de Datos:** Determinar qué información va a ser necesario almacenar en la misma, después de haber analizado los requisitos de los distintos usuarios.
- **Diseño de la Base de Datos:** Diseños Conceptual, Lógico y Físico. Tras especificar los requisitos, se debe diseñar modelo conceptual de la Base de Datos, para luego adecuar la estructura conceptual a un SGBD específico y a un modelo convencional de datos concreto. Una vez definido el conceptual, se pasará a realizar el diseño lógico. Si la elección fue el modelo relacional, se definirán las tablas. Por último, se diseñará el modelo físico, tratando de optimizar recursos y hacer la BD eficiente, dependiendo de las necesidades.
- **Especificaciones y vistas para los programas:** A partir de los requisitos de usuario, se definirán vistas externas para los programas de aplicación.
- **Estrategias de transición:** Decisiones sobre posible puesta en marcha en paralelo con otros sistemas, fases de implantación del mismo, controles necesarios, construcción de prototipos, etc.

## 2.1. Funciones principales del DBA

- **Procedimientos de explotación y uso:** Establecer la normativa necesaria para la utilización de la Base de Datos, el modo de solicitar el acceso a la misma, su actualización, etc.
- **Seguridad, integridad y confidencialidad:** Procedimientos de control y auditorías.
- **Control e interacción entre la red y la Base de Datos:** Para Bases de Datos accedidas a través de redes informáticas o de Bases de Datos distribuidas.
- **Normativa de la empresa:** debe tener un profundo conocimientos sobre las políticas y normas de la empresa, para poder ir de la mano en los estándares de la BD, metodologías de diseño de la misma, etc.

## 2.1. Funciones principales del DBA

- **Monitorización del sistema:** Estadísticas del rendimiento, áreas problemáticas a resolver, así como tiempos de respuesta, precisión de datos y facilidad de uso. La mayoría de productos SGBD incluyen tecnologías que reúnen y publican la mayoría de estadísticas mencionadas en informes.
- **Monitorización actividades del usuario.**
- **Actualización versiones del producto:** Análisis de las características que incorpora la nueva versión, sopesando si se decide por la adquisición de dicha nueva versión. Si es así, deberá de notificar a los usuarios, y administrar y controlar la total migración

## 2.2. Herramientas del DBA

- **Utilidades del SGBD y herramientas para copias de seguridad:** Menús para el arranque de la BD ante caídas, asignación de usuarios, etc. Herramienta de carga y descarga de datos a ficheros externos, etc.
- **Estándares a seguir** en la documentación del sistema y manipulación de datos.
- **Simuladores y monitores:** Que permiten al administrador estudiar distintas alternativas de instrumentación física con el fin de optimizar la BD.
- **Herramientas de ayuda y diseño:** Denominadas CASE (Computer Aided Software Engineering), que permiten automatizar en parte y dar soporte en el diseño de la Base de Datos, desde el análisis de los requisitos hasta la instrumentación.
- **Diccionario de datos:** Repositorio centralizado de información sobre datos tales como significado, relación con otros datos, origen, uso y formato.



# 3. Lenguajes

- 3.1. Lenguaje de manipulación
- 3.2. Lenguaje de definción

## 3.1. Lenguaje de manipulación de datos (DML)

- **Insertión de datos (INSERT):** Nos permitiría insertar datos en la Base de Datos mediante la siguiente sintaxis:
  - **INSERT INTO** *nombre\_tabla* (atributo<sub>1</sub>, atributo<sub>2</sub>, atributo<sub>n</sub>..) **VALUES** (dato<sub>1</sub>, dato<sub>2</sub>, dato<sub>n</sub>...)
- **Actualización de datos (UPDATE):** Nos permitiría actualizar datos de la Base de Datos mediante la siguiente sintaxis:
  - **UPDATE** *nombre\_tabla* **SET** valor = dato **WHERE** *condición*
    - Ejemplo: UPDATE pacientes SET diabetico = 1 where genero = 'H'
- **Borrado de datos (DELETE):** Mediante la siguiente sintaxis se podrá borrar datos (filas o tuplas) de las tablas. Se podrá indicar una condición que queramos que cumplan los datos con el WHERE.
  - **DELETE FROM** *nombre\_tabla* **WHERE** *condición*

## 3.2. Lenguaje de definición de datos (DDL)

	CREAR	MODIFICAR	ELIMINAR
BD	CREATE SCHEMA nombre AUTHORIZATION user		DROP SCHEMA nombre {RESTRICT   CASCADE}* TRUNCATE TABLE nombre_tabla (Para datos)
TABLA	CREATE TABLE nombre_tabla (definición_campo <sub>1</sub> , definición_campo <sub>2</sub> , definición_campo <sub>n</sub> )	ALTER TABLE nombre_tabla ADD definicion_campo	DROP TABLE nombre_tabla TRUNCATE TABLE nombre_tabla (Para datos)
VISTA	CREATE VIEW nombre_vista AS consulta_query (En la consulta escribiremos la que recoja los datos que queremos mostrar)	CREATE OR REPLACE VIEW nombre_vista AS consulta_query (En la consulta escribiremos la que recoja los datos que queremos mostrar)	DROP VIEW nombre_vista
ÍNDICE	CREATE UNIQUE INDEX nombre_indice ON nombre_tabla(campo_a_indexar) CREATE INDEX nombre_indice ON nombre_tabla(campo_a_indexar)	N/A	DROP INDEX nombre_indice
REGLA INTEGRIDAD	• Añadir clave primaria ALTER TABLE nombre_tabla ADD CONSTRAINT nombre_pk PRIMARY KEY (atributo)	N/A	ALTER TABLE nombre_tabla DROP CONSTRAINT campo_fk DROP CONSTRAINT campo_pk
	• Añadir clave ajena ALTER TABLE tabla ADD CONSTRAINT nombre_fk FOREIGN KEY (atributo_origen) REFERENCES atributo_destino		

## 3.3. Lenguaje de control de datos

- Se emplea para asignar diferentes niveles de permisos de acceso y/o manipulación sobre los objetos de la base de datos a los distintos usuarios de la misma.
- Las operaciones sobre las que se puede realizar son: INSERT, UPDATE, DELETE, SELECT, EXECUTE, CONNECT, USAGE, CREATE, DROP, etc.
- Sus sentencias más importantes son GRANT y REVOKE, para conceder o quitar permisos:
  - **GRANT** <operación> **ON** <objeto\_sobre\_el\_que\_dar\_permiso> **TO** <nombre\_usuario>
  - **REVOKE** <operación> **ON** <objeto\_sobre\_el\_que\_dar\_permiso> **FROM** <nombre\_usuario>

## 4. Extensiones procedurales

- SQL es un lenguaje creado con el propósito de consultar y gestionar bases de datos relacionales. Al no tratarse, por tanto, de un lenguaje de propósito general, carece de las estructuras necesarias para realizar programas de aplicación. Por ello, diversos fabricantes de SGBD e intérpretes SQL han creado y soportan diferentes lenguajes de programación (integrados con SQL), denominados extensiones procedurales.

## 4.1. Lenguajes procedurales

- PL/SQL de Oracle.
- SPL, publicado por IBM para la base de datos Informix.
- T-SQL (Transact SQL), publicado por Microsoft.
- PL/pgSQL, de Postgre, basado en Oracle PL/sQL.