



Σavante

3

BASE DE DATOS

Sistemas gestores de bases de datos (SGBD)

ÍNDICE

| | |
|--|-----------|
| / 1. Introducción y contextualización práctica | 4 |
| / 2. Definición, funciones y objetivos de los SGBD | 5 |
| 2.1. Funciones y objetivos | 5 |
| / 3. Arquitectura ANSI/X3/SPARC | 6 |
| / 4. Componentes de un SGBD | 7 |
| / 5. Tipos de SGBD | 8 |
| 5.1. Según el modelo lógico | 8 |
| 5.2. Según su tamaño o capacidad de procesamiento | 8 |
| 5.3. Según su arquitectura de ejecución | 8 |
| 5.4. Según la ubicación de la base de datos | 9 |
| / 6. Funcionamiento de un SGBD paso a paso | 9 |
| / 7. SGBD vs. ficheros clásicos | 9 |
| / 8. Sistemas gestores de bases de datos distribuidas (SGBDD) | 10 |
| 8.1. Objetivos y problemas | 10 |
| 8.2. Ejemplos de SGBDD | 11 |
| / 9. Tipos de lenguajes de bases de datos | 11 |
| 9.1. Lenguaje de definición de datos (DDL) | 11 |
| 9.2. Lenguaje de manipulación de datos (DML) | 12 |
| 9.3. Lenguaje de control (DCL y TCL) | 12 |
| 9.4. Lenguajes procedurales y de consultas específicas | 12 |

ÍNDICE

| | |
|---|-----------|
| / 10. SGBD libres y comerciales | 12 |
| 10.1. SGBD libres o de código abierto | 13 |
| / 11. SGBD comerciales | 14 |
| / 12. Caso práctico 1: “Base de datos para tienda y almacén” | 15 |
| / 13. Caso práctico 2: “Gestión de tienda y almacén” | 16 |
| / 14. Resumen y resolución del caso práctico de la unidad | 17 |
| / 15. Bibliografía | 18 |

OBJETIVOS

Entender qué es un SGBD y cuál es su utilidad.

Conocer la arquitectura y los componentes de los SGBD.

Identificar las funciones y los objetivos de un SGBD.

Conocer las ventajas de los SGBD frente a los sistemas de archivos tradicionales.

Conocer los tipos y las tecnologías de los SGBD.

/ 1. Introducción y contextualización práctica

Los sistemas gestores de bases de datos, o SGBD, nacieron en los años 60 del siglo pasado, principalmente para manejar grandes volúmenes de datos en industrias como la aeroespacial y la automovilística. Estos sistemas estaban muy vinculados al hardware, lo que dificultaba su interoperabilidad y hacía su programación bastante compleja.



Fig.1. Oracle es uno de los SGBD más extendidos.

En los años 80, la proliferación de equipos informáticos en empresas y organizaciones impulsó el desarrollo de aplicaciones informáticas más versátiles e intuitivas para los usuarios. En este contexto, nació Oracle en 1980 y se creó el lenguaje SQL en 1986, lo que revolucionó el sector con la aparición de los SGBD relacionales. La expansión de la informática y los servicios asociados permitió la rápida adopción y desarrollo de los SGBD, contribuyendo a su generalización.

En los años 90 surgió el concepto de bases de datos distribuidas para integrar las diferentes bases de datos de varios departamentos de una empresa. La informática se expandió rápidamente, llegando tanto a usuarios domésticos como a empresas. Hoy en día, los SGBD siguen evolucionando para enfrentar desafíos como la nube, garantizar la alta disponibilidad y soportar arquitecturas multitenant y el análisis masivo de datos (Big Data).

A continuación, vamos a plantear un caso práctico a través del cual podremos aproximarnos de forma práctica a la teoría de este tema.

Escucha el siguiente audio donde planteamos la contextualización práctica de este tema. Encontrarás su resolución en el apartado 'Resumen y resolución del caso práctico'.



Audio intro "SGBD en la actualidad"

<https://on.soundcloud.com/MupMiobSUuPIRcSN7>



/ 2. Definición, funciones y objetivos de los SGBD

Un SGBD es una aplicación de software que facilita a los usuarios, incluyendo analistas, programadores y otros, la gestión y manipulación de los datos en una base de datos, garantizando en todo momento la disponibilidad y seguridad de la información.

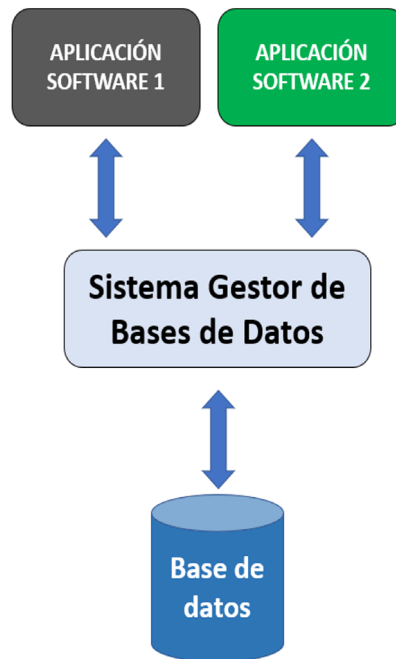


Fig.2. El SGBD sirve a las aplicaciones como interfaz de acceso a las bases de datos.

2.1. Funciones y objetivos

Entre las **principales funciones y objetivos** de un SGBD destacan:

- Posibilitan **consultar, almacenar y manipular** datos de forma eficiente, fácil y transparente para el usuario.
- Aseguran el **control del sistema y su seguridad**, gestionando de forma precisa los permisos de los usuarios autorizados.
- Posibilitan la **migración del sistema a otras máquinas** sin importar el sistema operativo, permitiendo la distribución de la base de datos.
- **Solucionan los problemas ocasionados por la concurrencia** de usuarios a los datos (acceso simultáneo a los mismos datos).
- Garantizar la **integridad de los datos**, asegurando consistencia en toda la base de datos.
- Facilitan la **visualización de la arquitectura de las bases de datos**, su estructura en tablas, registros, campos y relaciones entre sí.
- Permiten el **uso de lenguajes de programación** para manipular la información.
- Aportan las herramientas necesarias para la realización y recuperación de **copias de seguridad**.

- Asegura la **redundancia mínima de datos**, optimizando así la capacidad de almacenamiento necesaria.
- Permiten la **consulta de las estadísticas de uso del SGBD**.



Investigamos...

Recomendamos que realices la búsqueda del concepto de sistema de gestión de datos (SGBD) a través de Google Scholar, de modo que puedas ampliar conceptos.

/ 3. Arquitectura ANSI/X3/SPARC

Los SGBD están diseñados en base a un modelo conceptual estandarizado que proporciona una base sólida y de referencia para lograr una **separación en tres niveles que abstraigan al usuario de los detalles técnicos del sistema, permitiendo mejorar la eficiencia, escalabilidad e independencia de los datos**. Este modelo conceptual es el estándar ANSI/X3/SPARC, elaborado por el subcomité X3 de la ANSI en colaboración con SPARC.

Como podemos ver en la siguiente imagen, **el nivel físico y lógico suelen ser únicos, pero existen múltiples niveles externos o de usuario** debido a la variedad de usuarios en el sistema.

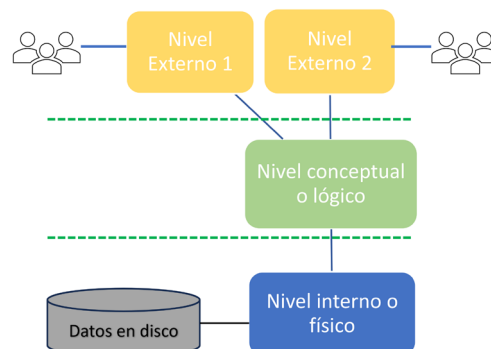


Fig.3. Arquitectura ANSI/X3/SPARC de un SGBD.

En la siguiente tabla, podemos ver las características de cada nivel:

| Nivel | Características |
|---------------------|--|
| Externo o usuario | Representa la visión que el usuario tiene del sistema . Es el nivel más cercano a este. |
| Conceptual o lógico | Se centra en la estructura de los datos y las relaciones que hay entre ellos . Este nivel es ajeno al SGBD utilizado. Es, quizá, el nivel más importante. Contiene: <ul style="list-style-type: none"> • Entidades (clientes, pedidos, etc.). • Atributos: (nombre, DNI, etc.). • Relaciones entre ellos (compras, etc.). • Normas a cumplir por los datos. |
| Interno o físico | Muestra cómo se almacenan los datos : la ubicación de los archivos, sus nombres, tipología y forma de acceder a los registros que los componen. Es el nivel más cercano a la organización real de los datos. Depende del hardware y software utilizado. |

Tabla 1. Niveles de abstracción en un SGBD.

El objetivo de estos niveles es conseguir la **independencia físico-lógica de los datos** que permita representar los datos que necesite cada usuario. Escuchemos el siguiente audio para profundizar nuestros conocimientos sobre ella.



Audio 1. "Independencia físico-lógica de un SGBD"

<https://on.soundcloud.com/1E7PCteLp4LYNDw1A>



/ 4. Componentes de un SGBD

Un SGBD suele estar compuesto por los siguientes elementos:

- **Datos almacenados:** Estructurados en ficheros de manera eficiente y adecuada.
- **Lenguaje de programación:** Permite a los usuarios acceder a los datos para añadir, modificar o consultar información.
- **Diccionario de datos:** Herramienta que detalla los elementos de una base de datos, proporcionando definiciones, formatos y relaciones. Contiene **metadatos** (información sobre los datos y su estructura), como nombres de campos, tipos de datos y restricciones. Se utiliza para garantizar la coherencia y comprensión de los datos. Hay dos tipos: activos (se actualizan automáticamente) y pasivos (requieren actualización manual).
- **Utilidades y herramientas de la base de datos:** Incluye herramientas que facilitan la gestión y administración correcta de la base de datos. Ayudan a los administradores en tareas como la gestión de usuarios, grupos y sus permisos, las copias de seguridad, la generación de estadísticas e informes, etc. Estas herramientas son imprescindibles para el buen funcionamiento y mantenimiento del SGBD.
- **Entorno gráfico:** Facilita el acceso al sistema para usuarios menos avanzados, haciendo su uso más intuitivo.
- **Usuarios:** Son el principal activo de la base de datos junto con los datos, encargándose de su explotación, gestión y mantenimiento. Hay diferentes perfiles de usuarios:
 - » **Administradores (DBA):** Gestionan, mantienen y optimizan el SGBD, garantizando su seguridad, rendimiento y disponibilidad. Sus tareas abarcan instalación, monitoreo, respaldo, recuperación y cumplimiento de normas.
 - » **Analistas:** Diseñan la base de datos y supervisan su implantación y desarrollo.
 - » **Programadores:** Desarrollan aplicaciones de usuario siguiendo las directrices de los analistas.
 - » **Operadores de mantenimiento:** Solucionan incidencias y dan soporte a los usuarios en tareas cotidianas.
 - » **Usuarios finales:** Explotan la base de datos con diferentes permisos y perfiles.

En la siguiente imagen, podemos ver qué perfiles intervienen en cada uno de los esquemas de la estructura del SGBD.

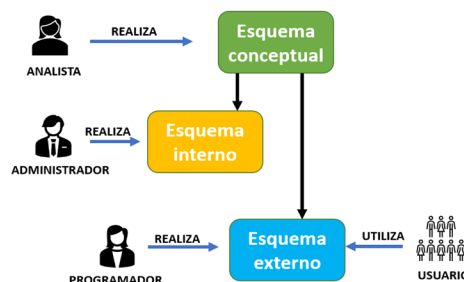


Fig.4. Perfiles de usuarios relacionados con los esquemas.

/ 5. Tipos de SGBD

Tal y como ocurre con las bases de datos, existen múltiples maneras de clasificar los SGBD.

5.1. Según el modelo lógico

Como hemos visto en el tema de bases de datos, estas pueden utilizar diferentes modelos de datos para organizar la información. Por tanto, podemos destacar SGBD de **red**, **relacionales**, **orientados a objetos**, **objeto-relacionales**, **NoSQL**, etc.

5.2. Según su tamaño o capacidad de procesamiento

Según la magnitud de los datos a procesar, distinguiremos:

- **Ligeros:** Tienen un tamaño reducido y se utilizan en equipos con escasa potencia o que requieren un bajo consumo energético y no necesitan un alto rendimiento. Aquí podemos encontrar Microsoft Access, LibreOffice Base o SQLite.
- **Alto rendimiento:** Permiten manipular grandes volúmenes de datos y disponen de funcionalidades complejas como las que hemos mencionado en el apartado de 'Funciones y objetivos'. En este grupo, podemos encontrar Oracle, MySQL, Microsoft SQL Server, MongoDB, Hadoop, etc.



Investigamos...

Recomendamos que realices la búsqueda del volumen de datos que pueden manejar diferentes SGBD a través de Google Scholar, de modo que puedas ampliar conceptos.

5.3. Según su arquitectura de ejecución

Una clasificación importante que podemos destacar es en base al número de capas que intervienen entre la aplicación cliente y el SGBD. Según este criterio, podemos distinguir:

- **SGBD Monocapa:** Tanto el cliente como el SGBD y la base de datos se encuentran en el mismo equipo. Es el esquema que siguen los SGBD Ligeros como LibreOffice Base y SQLite.
- **SGBD Bicapa:** Esta categoría sigue un esquema cliente/servidor, ya que, en él, el SGBD y la base de datos se encuentran en la misma máquina, pero el cliente se halla en otra, teniendo que comunicarse entre sí a través de la red.
- **SGBD Multicapa:** Esta arquitectura destaca por la existencia de varios servidores que hacen de intermediarios entre los clientes y las bases de datos. Es la más habitual en la web.

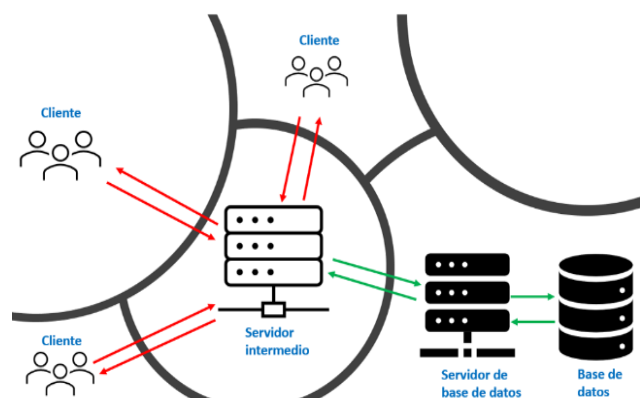


Fig.5. esquema SGBD multicapa.

5.4. Según la ubicación de la base de datos

Podemos distinguir entre SGBD centralizados o distribuidos. Es decir, el SGBD y la base de datos pueden estar centralizadas en un equipo o distribuidas en varios, conectados entre sí a través de la red.

/ 6. Funcionamiento de un SGBD paso a paso

En la imagen siguiente, se representan los procesos que ocurren entre una solicitud de datos hecha por el usuario y el SGBD, en un entorno de acceso a través de la red.

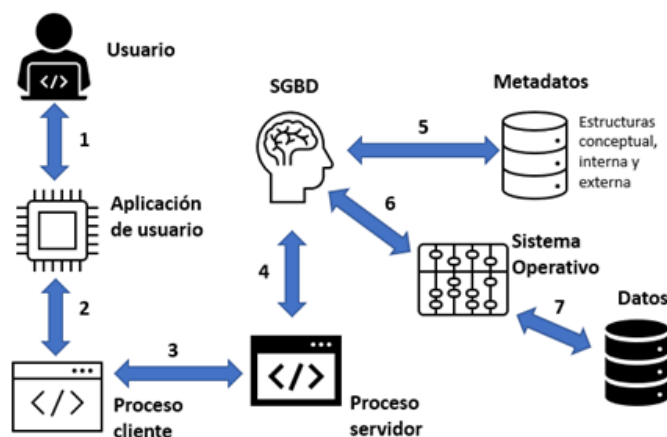


Fig.6. Esquema del funcionamiento del SGBD.

El usuario, mediante la aplicación (nivel externo), realiza una solicitud de acceso a la información contenida en la base de datos **(1)**. Esta solicitud se traduce para que sea comprensible por el sistema.

A continuación, el proceso de cliente **(2)** se activa para manejar la solicitud y enviarla al proceso servidor de la base de datos **(3)**. El servidor recoge la solicitud, la analiza y la traduce antes de pasarla al SGBD **(4)** para su procesamiento.

Una vez que la solicitud llega al SGBD, se verifica su validez mediante el diccionario de datos, y se determina en qué parte específica de los archivos se encuentra la información solicitada **(5)**. Con esta información, el SGBD traduce y envía la solicitud al sistema operativo **(6)**, que accede directamente al soporte físico de los datos **(7)**.

El sistema operativo recibe la solicitud del SGBD, la procesa y responde al SGBD con la información solicitada, siempre y cuando no haya errores.

El SGBD traduce los datos recibidos del sistema operativo y los entrega al proceso servidor. Este último envía la información procesada al proceso cliente a través de la red. El cliente verifica los datos y los entrega a la aplicación del usuario, que los presenta en un formato gráfico adecuado para su visualización.

/ 7. SGBD vs. ficheros clásicos

Antes de que existieran los SGBD, la información se gestionaba mediante ficheros. Estos ficheros consistían en varios registros contenidos en una estructura única, accesible a través de aplicaciones informáticas. En contraste, las bases de datos son **entidades que comprenden varios tipos de ficheros interrelacionados**, y suelen estar gestionadas por un SGBD que facilita y mejora su administración.

El usuario final de la base de datos no necesita saber cómo está organizada o distribuida la información en la base de datos. Algunas ventajas de utilizar SGBD frente a los ficheros clásicos incluyen:

- **Optimización del espacio de almacenamiento:** En los sistemas de archivos, evitar ficheros duplicados es complicado.
- **Acceso concurrente:** Permite que varios usuarios accedan a la información simultáneamente desde diferentes ubicaciones, lo cual puede ser problemático en sistemas de archivos tradicionales.
- **Acceso más sencillo a los datos:** En los sistemas de archivos, acceder a ciertos datos puede ser difícil, especialmente con grandes volúmenes de información.
- **Integridad de los datos asegurada:** Los SGBD controlan que los datos almacenados cumplan con ciertas restricciones de consistencia y coherencia, evitando el mal uso del sistema.
- **Seguridad mejorada:** Solo los usuarios previamente autorizados pueden acceder a los SGBD.



Fig.7. La seguridad es una de las ventajas de los SGBD frente a los sistemas de archivos tradicionales.

Una gran ventaja de los SGBD sobre la gestión mediante ficheros es la capacidad de utilizar **lenguajes estándar de consulta, como SQL, que veremos en este tema y en los próximos, y que pueden combinarse con otros lenguajes como Java o PHP** para crear funcionalidades más potentes y complejas. Además, actualmente, hay una amplia variedad de SGBD adaptados a diferentes mercados y necesidades específicas, tanto libres como comerciales.

/ 8. Sistemas gestores de bases de datos distribuidas (SGBDD)

Los SGBDD deben realizar las mismas funciones que los centralizados, como la manipulación, el control y la consulta de datos, pero, además, deben tener en cuenta que la información puede estar distribuida en múltiples ubicaciones.

8.1. Objetivos y problemas

Para poder asegurar **una correcta gestión de las bases de datos distribuidas, el SGBDD debe garantizar tres objetivos clave:**

- **Transparencia a nivel de red:** El usuario no tiene por qué saber si la información está distribuida o no.
- **Transparencia de fragmentación:** El usuario no debe saber si se ha aplicado fragmentación o no.
- **Transparencia de replicación:** El usuario no necesitará saber si la información está replicada en varios nodos.

Para ello, el SGBDD se enfrenta a los siguientes **problemas** que debe solucionar:

- Fragmentación y replicación de datos.
- Sincronizar las actualizaciones de los datos en las réplicas.
- Procesamiento y optimización de consultas distribuidas.
- Gestión de la concurrencia en las transacciones distribuidas, que se vuelve más complejo por la existencia de varios nodos.

- Recuperación si falla algún nodo o todo el sistema, desviando las solicitudes a nodos operativos y restaurando los datos.
- Garantizar a nivel global la seguridad e integridad de los datos en el sistema mediante permisos y cifrado de datos.
- Balanceo de carga de trabajo en los diferentes nodos.

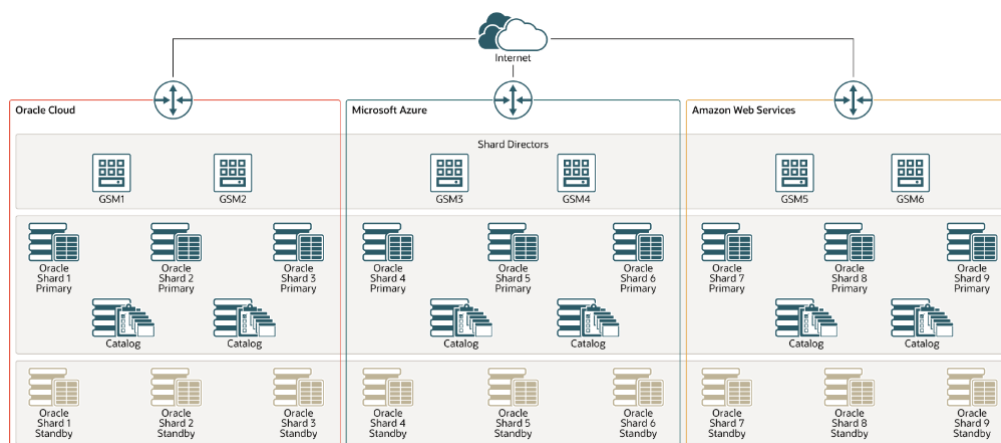


Fig.8. Topología de Oracle Database con particiones horizontales, distribuidas en Oracle Cloud, Azure y AWS.

8.2. Ejemplos de SGBDD

En la actualidad, hay una gran variedad de SGBDD con diferentes características, entre los que podemos destacar:

- PrestoDB (empleado por grandes empresas como Facebook, Netflix, Groupon, AirBnB, etc.).
- Apache Cassandra.
- Impala.

Enlaces de interés...

Te dejamos este enlace de interés a un vídeo donde se muestra el proceso paso a paso para crear bases de datos distribuidas en Oracle Cloud:
<https://www.youtube.com/watch?v=yaZrqXdnrM>

/ 9. Tipos de lenguajes de bases de datos

En la gestión de bases de datos, se utilizan diferentes lenguajes, la mayoría basados en SQL, que se pueden clasificar en varias categorías.

9.1. Lenguaje de definición de datos (DDL)

Permite definir y editar la estructura de la propia base de datos. Con este lenguaje se pueden crear, modificar y eliminar objetos como tablas, índices, etc. Los comandos más habituales en este lenguaje son **Create**, **Alter** y **Drop**.

9.2. Lenguaje de manipulación de datos (DML)

Permite insertar, consultar, actualizar y eliminar datos. Las sentencias más comunes son **Select**, **Insert**, **Update** y **Delete**.

9.3. Lenguaje de control (DCL y TCL)

En este grupo, podemos encontrar dos subgrupos de lenguajes que permiten llevar a cabo acciones de control:

- **Lenguaje DCL:** El objetivo de este lenguaje es controlar los permisos de acceso a los datos almacenados. **Grant** y **Revoke** son los comandos más comunes.
- **Lenguaje TCL:** Tiene la finalidad de gestionar las transacciones para asegurar que las operaciones se lleven a cabo de forma segura. Los comandos más comunes en este lenguaje son **Commit** y **Rollback**.

9.4. Lenguajes procedurales y de consultas específicas

Estos lenguajes se pueden clasificar de la siguiente manera:

- **Lenguajes procedimentales:** Están basados en SQL y permiten crear funciones, procedimientos y disparadores para la gestión de los datos. Ejemplo: T-SQL, PL-SQL, etc.
- **Lenguajes de consulta de datos semiestructurados:** Permiten el manejo de datos en formato XML (XQuery) o JSON (JSONPath).
- **Lenguajes NoSQL:** Utilizados en bases de datos NoSQL que pueden manejar datos en diferentes formatos, como documentos, pares clave-valor, etc.

```
// Seleccionar o crear una base de datos
use miBaseDeDatos

// Crear una colección
db.createCollection("socios")

// Insertar un solo documento
db.socios.insertOne({
  nombre: "Esther",
  edad: 38,
  email: "esther.garcia@example.com"
})

// Consultar todos los documentos
db.socios.find()

// Consultar documentos con una condición
db.socios.find({ nombre: "Esther" })
```

Fig.9. Ejemplo de código MongoDB.

/ 10. SGBD libres y comerciales

En el mercado actual, hay una amplia variedad de SGBD disponibles, cada uno con diferentes tipos, funcionalidades y características. A continuación, describiremos algunos de los más **conocidos y utilizados** hoy en día.

10.1. SGBD libres o de código abierto

Algunos de los **principales** SGBD de código abierto son:

- **MySQL:** Este SGBD relacional es uno de los más populares del mundo. Soporta varios lenguajes de programación, como Java, C++, Python y PHP, y es muy utilizado en aplicaciones web. Aunque comenzó como un sistema de código libre, ahora tiene una licencia libre y una comercial (Oracle). Fue desarrollado por MySQL AB, empresa que fue adquirida por Oracle. MariaDB es una bifurcación de MySQL creada por varios de sus desarrolladores originales.



Fig.10. Logo de MySQL.

- **PostgreSQL:** Considerado un SGBD avanzado, PostgreSQL es un sistema relacional orientado a objetos y multiplataforma. Soporta múltiples lenguajes de programación, como Perl, Python, PHP y Java. Ha sido desarrollado por una comunidad de programadores de forma voluntaria.



Fig.11. Logo de PostgreSQL.

- **SQLite:** Este gestor de bases de datos es muy ligero pero potente, con tiempos de acceso muy rápidos. Es multiplataforma y soporta múltiples lenguajes de programación. Fue desarrollado en C y es muy utilizado en aplicaciones móviles.



Fig.12. Logo de SQLite.

- **MongoDB:** Es una base de datos no relacional (NoSQL) distribuida, diseñada específicamente para aplicaciones en la nube. Es también multiplataforma.
- **Firebird:** Este SGBD relacional es conocido por su buena gestión de la concurrencia y bajo consumo de recursos, aunque sigue siendo potente y multiplataforma. Soporta varios lenguajes de programación.



Enlaces de interés...

En el siguiente vídeo se muestran algunas características de varios SGBD:
<https://www.youtube.com/watch?v=JecdMJDAR8I>

/ 11. SGBD comerciales

Hay una amplia variedad de SGBD comerciales disponibles, de distintos tipos y clases. A continuación, se describen las principales **características de los más ampliamente utilizados**:

- **Oracle:** Este SGBD objeto-relacional es probablemente el más reconocido y utilizado a nivel mundial. Destacado por su potencia, robustez, seguridad y flexibilidad, está basado en una arquitectura cliente/servidor. Es muy popular entre grandes empresas y corporaciones debido a su fiabilidad y soporte robusto. Oracle también es propietaria de Sun Microsystems (Java) y MySQL.



Fig.13. Logo de Oracle.

- **Microsoft SQL Server:** Creado por Microsoft, es un SGBD relacional ampliamente utilizado. Se caracteriza por su rapidez en la conmutación y alta disponibilidad, además de su robusta integración con Microsoft Server. Es escalable y ofrece diversas versiones adaptadas a diferentes usos, como web, empresa, Data Center y Business Intelligence. También puede ser utilizado en máquinas virtuales de Azure, proporcionando seguridad y facilidad de gestión.



Fig.14. Logo de SQL Server.

- **Sybase-SAP:** Este SGBD relacional es conocido por su escalabilidad y buen soporte para bases de datos con grandes volúmenes de datos. Ofrece opciones para implementar funcionalidades en la nube y está disponible en diferentes paquetes según las necesidades específicas. Soporta múltiples plataformas, incluyendo Windows, Linux, Unix y Solaris.
- **DB2:** Este SGBD multiplataforma fue desarrollado por IBM en los años 70 y cuenta con un motor potente desarrollado por IBM. Sobresale por su capacidad para automatizar tareas, lo que mejora la rapidez de respuesta. Existen diferentes versiones de DB2, incluida una versión gratuita con funcionalidades limitadas.



Fig.15. Logo de IBM-DB2.



Vídeo 1. "SGBD más extendidos y utilizados actualmente"

<https://bit.ly/3dRir3O>



/ 12. Caso práctico 1: “Base de datos para tienda y almacén”

Planteamiento: Continuando con el planteamiento de los temas anteriores, seguimos apoyando a nuestro amigo en la implementación de la base de datos que necesita para su tienda.

Nudo: En el tema de bases de datos, ya abordamos las entidades, las tablas, los registros y los campos que consideraremos en este primer diseño. Recordemos que el objetivo principal es crear un sistema de base de datos que facilite la gestión de su almacén, así como los procesos de ventas y compras.

Con más información sobre la estructura y los componentes de un SGBD, vamos a definir cómo será esa estructura y cuáles serán los componentes en nuestro caso.

Desenlace: Una propuesta básica puede ser la siguiente:

En cuanto al nivel interno, que se determinó en parte en el caso práctico del tema de bases de datos, queda pendiente la decisión sobre la ubicación de los ficheros y sus nombres. Nuestra propuesta es que los nombres de los ficheros coincidan con los de las tablas, por simplicidad, y que se almacenen en un dispositivo tipo NAS con un sistema RAID redundante para asegurar la integridad de los datos.

Respecto al esquema lógico o conceptual, en el tema de bases de datos se definieron las tablas y sus atributos, pero aún falta determinar cómo se relacionarán entre sí. Una posible propuesta puede ser la siguiente, aunque pueden existir otras alternativas

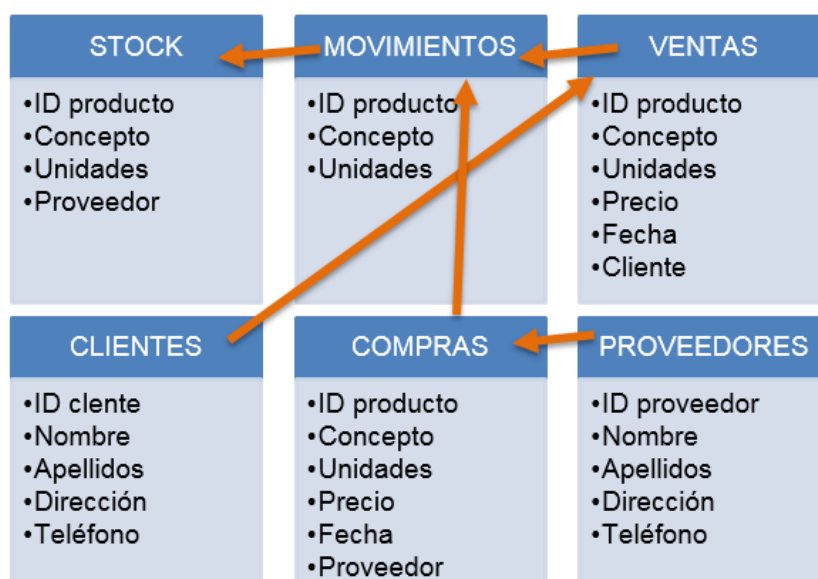


Fig.16. Nivel lógico o conceptual.

Finalmente, en lo que se refiere al nivel externo, dependerá en gran medida del SGBD que decidamos utilizar. También es necesario definir los perfiles de usuario o grupos de usuarios que implementaremos. En nuestro caso, habrá un perfil de administrador con permisos completos, un perfil de usuario final avanzado y un perfil de usuario final de consulta con acceso limitado a ciertas funciones.

/ 13. Caso práctico 2: “Gestión de tienda y almacén”

Planteamiento: Necesitamos conocer las alternativas de SGBD en el mercado para seleccionar uno para la base de datos que estamos construyendo para nuestro amigo.

Nudo: Como diseñadores, necesitamos recabar información sobre ocho SGBD distintos de nuestra elección.

La información necesaria de cada uno para la comparativa es:

- Características principales.
- Ventajas.
- Inconvenientes.

Desenlace: Una posible solución es la siguiente tabla con los SGBD más conocidos, aunque hay muchos otros en el mercado que también pueden ser interesantes.

| SGBD | Características | Ventajas | Inconvenientes |
|------------|--|--|---|
| ACCESS | Perteneciente a Microsoft. Es muy gráfico, métodos simples y directos con formularios, para trabajar con la información. | Asequible para personas con poco manejo de bases de datos. Crea varias vistas para una misma información. | No es multiplataforma. No funciona con bases de datos grandes, tanto para usuarios como para registros. |
| SQLite | Los tipos de datos se asignan a valores, no a columnas. | Multiplataforma. Configuración innecesaria. Acceso muy rápido. No requiere servidor. | El dinamismo de los datos no es portable a otras bases de datos. Sin claves foráneas. |
| SQL SERVER | Software propietario. El lenguaje es TSQL. | Multiplataforma, aunque pertenezca a Microsoft. Transacciones. | Utiliza mucha RAM. Tamaño de página fijo y pequeño. Relación calidad/precio inferior a Oracle. |
| MYSQL | Pertenece a Oracle. Licencia GPL/ licencia comercial. | Agrupamiento de transacciones. Varios motores de almacenamiento. Instalación sencilla. | No tiene soporte. Capacidad limitada. |
| POSTGRESQL | Tiene la extensión POSTGIS para bases de datos especiales. | Código abierto y gratuito. Multiplataforma. Gran volumen de datos. Transacciones, disparadores y afirmaciones. | Respuesta lenta. Requiere hardware. No es intuitivo. |
| ORACLE | Dispone de su propio lenguaje PL/SQL. Soporta bases de datos de gran tamaño. | El más usado mundialmente. Multiplataforma, intuitivo y fácil de usar. | Precio muy elevado, alto coste de formación por Oracle. |
| MongoDB | Almacena documentos BSON. Esquema flexible. Escalabilidad horizontal. | Modelo de datos flexible sin esquema fijo. Alta disponibilidad. | Mayor consumo de disco por redundancia, consistencia entre réplicas lenta. |

Tabla 2. Comparativa de características de SGBD.

/ 14. Resumen y resolución del caso práctico de la unidad

Hemos explorado varios aspectos esenciales de los SGBD y los SGBDD. Hemos analizado su funcionamiento, la arquitectura ANSI/X3/SPARC, sus componentes principales y los distintos tipos de SGBD.

Además, hemos abordado conceptos como la transparencia en los SGBD, destacando las ventajas de estos sistemas sobre los ficheros tradicionales. Por último, hemos explicado diversos lenguajes de bases de datos y revisado ejemplos tanto de SGBD libres como comerciales.

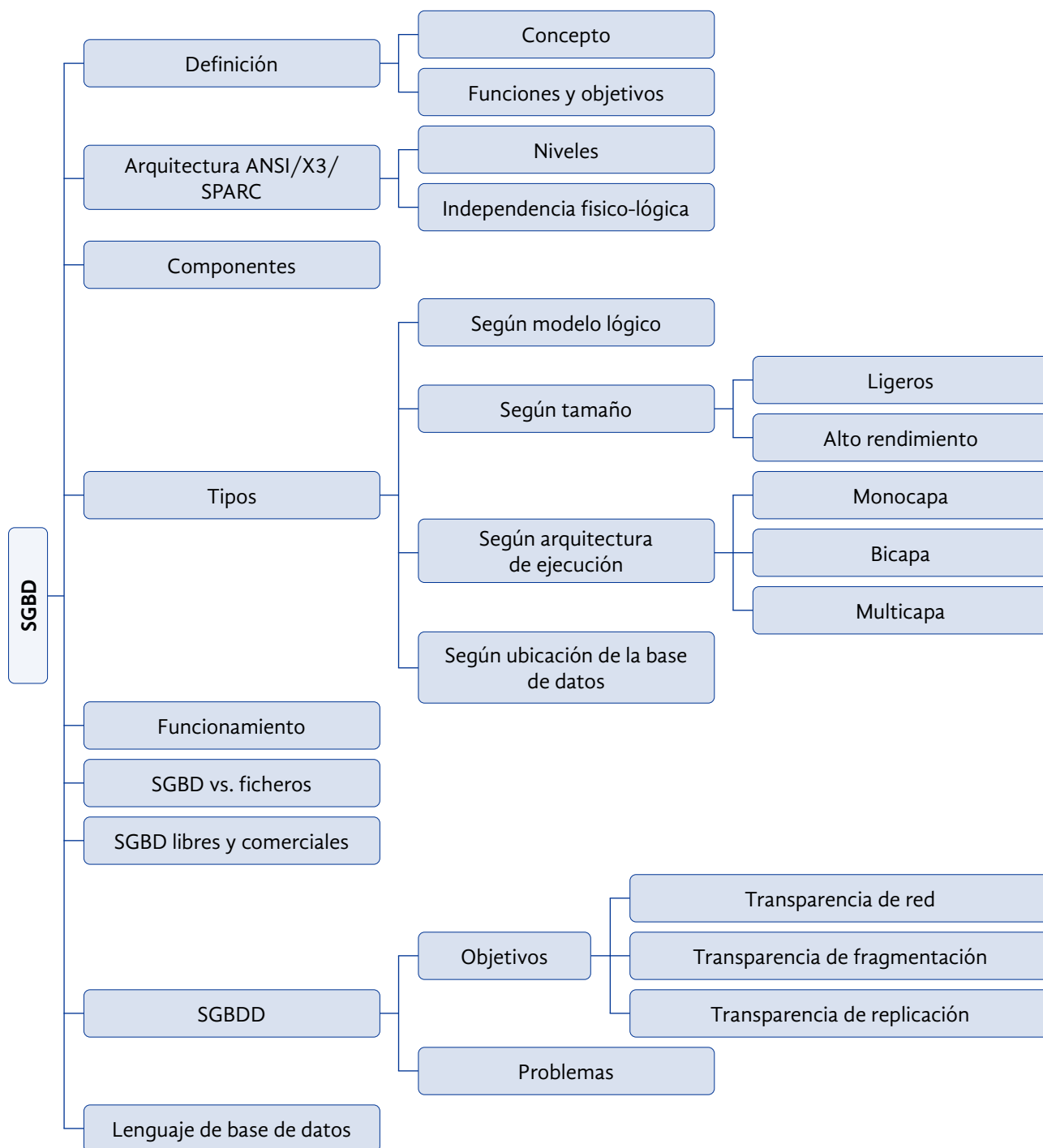


Fig.17. Esquema resumen del tema.

Resolución del caso práctico de la unidad

Las respuestas a las preguntas planteadas podrían ser las siguientes:

¿Qué SGBD sería el más recomendable para usar?

Depende, ya que algunos SGBD se ajustan mejor que otros a determinadas casuísticas. Además, esto dependerá en gran medida de los objetivos que se persigan y de cómo se diseñe y administre la base de datos.

¿Qué SGBD brindará mejores prestaciones y tecnología?

Igualmente, tras todo lo analizado en este tema, deberíamos responder con un 'depende' a esta pregunta, ya que efectivamente variará según el caso y la problemática específica. En un entorno empresarial más profesional, probablemente lo más sensato sea optar por tecnologías establecidas como Oracle, MySQL o SQL Server. En otros contextos, puede ser más adecuado utilizar modelos alternativos, como SQLite o PostgreSQL.

¿Es importante para nosotros disponer de un SGBD para administrar nuestra base de datos?

En este caso, la respuesta debe ser sí. Actualmente, un SGBD es esencial para aprovechar al máximo el rendimiento de una base de datos, incluso si su uso es simple. La inclusión de herramientas gráficas y visuales potentes facilita cada vez más su utilización.

/ 15. Bibliografía

Sánchez, G. C. (2001). *Sistemas gestores de bases de datos*. Paraninfo.

Elmasri, R., & Navathe, S. (2007). *Fundamentos de sistemas de bases de datos*. Addison Wesley.

López, I.; Castellano, M. J., y Ospino, J. (2011). *Bases de datos*. Garceta.

Oppel, A. (2009). *Databases A Beginner's Guide*. McGraw-Hill Education.

Jesús (2023, 27 noviembre). Domina la Arquitectura de 3 Niveles en Bases de Datos. *Dongee*. Recuperado de:
<https://www.dongee.com/tutoriales/arquitectura-de-3-niveles-bases-de-datos/>

Peterson, R. (2024, 9 marzo). Data Independence in DBMS: Physical & Logical with Examples. *Guru99*. Recuperado de:
<https://www.guru99.com/es/dbms-data-independence.html>

Greyrat, R. (2022, 5 julio). Independencia de datos físicos y lógicos. *Barcelona Geeks*. Recuperado de:
<https://barcelonageeks.com/independencia-de-datos-fisicos-y-logicos/>