



UNIVERSIDAD
CAECE

Cámara Argentina de Comercio y Servicios

TECNICATURA EN PROGRAMACIÓN

BASES DE DATOS

GUIA DE LA UNIDAD Nro. 1

Contenidista: Mg. Silvia Cobialca

INDICE – UNIDAD 1:

1. INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS.....	5
1. PRIMERO, UN POCO DE HISTORIA.....	5
2. ¿QUÉ ES UN SISTEMA GESTOR DE BASE DE DATOS (O DBMS EN INGLÉS)?.....	8
AUTOEVALUACIÓN 1	10
3. LOS DIFERENTES TIPOS DE USUARIOS	11
4. INDEPENDENCIA DE LOS DATOS	14
AUTOEVALUACIÓN 2	16
5. LENGUAJES DBMS	16
6. MÓDULOS COMPONENTES DE UN SISTEMA DE BASES DE DATOS	18
AUTOEVALUACIÓN 3	20
SÍNTESIS DE LA UNIDAD	21

Figuras

Figura 1 – Arquitectura de tres Esquemas.....	13
Figura 2. Componentes del DBMS	19

MAPA DE LA UNIDAD 1: INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS

PROPÓSITOS

En esta unidad nos proponemos explicarle los conceptos básicos sobre los sistemas de bases de datos. Veremos las definiciones, su utilidad y un poco del contexto histórico.

OBJETIVOS

- ✓ Conocer qué es un DBMS o Database Management System.
- ✓ Analizar cuál es la utilidad de un DBMS.
- ✓ Conocer qué significa aislamiento y abstracción en el contexto de un DBMS.
- ✓ Analizar la arquitectura de un DBMS y sus componentes más importantes

CONTENIDOS

Para que alcance los objetivos, los contenidos que abordará son los siguientes:

- 1) Bases de datos. Datos e información.
- 2) DBMS. Componentes. Funciones. Usuarios.
- 3) Arquitectura del DBMS.
- 4) Independencia de datos.
- 5) Ventajas y beneficios. Data Sub-language. DDL. DML.
- 6) Tablas. Manipulación de datos. Ambientes de bases de datos.

PALABRAS CLAVES

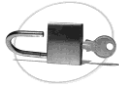
Base de datos, independencia física, abstracción de datos



BIBLIOGRAFÍA DE CONSULTA

ELMASRI, R./NAVATHE, SH.(2011). *Fundamentos de Sistemas de Base de Datos*.6ta Ed, EEUU: Pearson / Addison Wesley.

1. INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS



ABORDEMOS EL LOGRO DEL PRIMER OBJETIVO DE LA UNIDAD:

- ✓ Conocer qué es un DBMS o Database Management System.

Comenzamos con nuestra materia y lo primero que nos proponemos es explicarle los conceptos básicos relacionados con el uso de estructuras de datos. Esta unidad nos servirá para comprender cómo surgieron las bases de datos a partir de estas estructuras, y qué mejoras introdujeron en los métodos de manipulación de datos.

Comencemos...



1. PRIMERO, UN POCO DE HISTORIA...

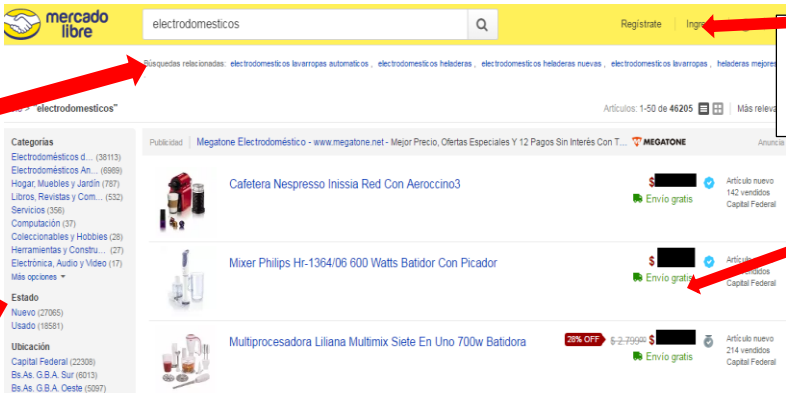
Antes de que se conociera el concepto de bases de datos, las aplicaciones (o programas) utilizaban “procesamiento de archivos” para trabajar con los datos que estaban guardados en ellos. Entonces cada usuario definía e implementaba los archivos necesarios para utilizar los datos como parte de la programación de dicha aplicación.



Como seguramente recordará haber visto en las materias de Programación, el programador de la aplicación definía los datos que se iban a necesitar para el archivo “Clientes” y creaba lo que se denominaba un “record” o “registro” (que contenía por ejemplo el nombre, la dirección, el CP, el teléfono del cliente). Quizás los datos del producto que había comprado cada cliente los guardaba en otro archivo, el de “Ventas”. Cuando buscaba, por ejemplo, la historia de las ventas de un cierto cliente, debía recorrer ambos archivos, el de clientes para encontrar los datos del cliente y el de ventas para buscar sus ventas.

Se recorría en forma secuencial (desde el comienzo, registro por registro) cada archivo y se iba mostrando la información a medida que se la encontraba. Si había que realizar cálculos, debían usarse variables incrementales para ir sumando las ventas de cada cliente. Un proceso bastante largo y que generalmente llevaba mucho tiempo.

Veamos ahora otro ejemplo en donde se manejan datos. Quizás usted haya entrado alguna vez a una aplicación en línea de venta de productos como la que le mostramos en la imagen de la página siguiente. Veremos que en cada paso que el usuario realiza cuando efectúa la compra, la aplicación siempre debe ir a buscar los datos relevantes y presentarlos al cliente. Hemos marcado algunos ejemplos de esta interacción en la imagen para ilustrar el ejemplo.



Los productos existen en la base de datos, al buscarlos, la aplicación consulta la lista de productos de acuerdo a la condición de la búsqueda

Para ingresar, los usuarios deben haberse dado de alta primero. Al darse de alta, son ingresados a la base de datos

Las categorías son también datos que se encuentran en la base de datos, por eso se puede contar cuantos productos hay en cada una

Todos los datos de los productos que aparecen en la lista están dentro de la base de datos almacenados para cada producto (precio, ubicación, vendedor, descripción, etc.)



Pensemos juntos ahora otro ejemplo, donde hay una aplicación utilizada para vender autos y hay dos funcionalidades independientes entre sí. La primera es la que efectivamente se utiliza para gestionar la venta propiamente dicha y guardar los datos del auto, el comprador, la fecha, datos del vendedor, etc. La segunda es la que se utiliza todas las semanas para emitir un reporte de ventas para la gerencia.

Utilizando el procesamiento de archivos, antes de poder trabajar en la aplicación del reporte, el programador debe saber dónde se encuentra el archivo de la aplicación de ventas, en qué máquina y en qué disco y carpeta de dicha máquina se encuentra guardado.

Él desarrolla su aplicación y todo anda bien hasta que un día el desarrollador del programa de ventas decide introducir una mejora. Como parte de ésta, decide mover el archivo a otra carpeta dentro del disco.



¿Qué sucederá cuando la segunda aplicación necesite emitir el reporte de ventas el próximo viernes por la tarde?

Seguramente usted nos responderá que, dado que el archivo al que accedía la aplicación del reporte ha sido movido a otra carpeta, no lo podrá acceder y fallará la aplicación con un error de “archivo no encontrado”.

Sigamos con el ejemplo, pero ahora suponiendo que ambos programadores son la misma persona o están en contacto, o sea que se informan mutuamente de los cambios implementados en sus respectivas aplicaciones.



¿Qué información le parece que va a impactar en la aplicación del reporte y qué es clave que el programador tenga en cuenta?

Usted estará respondiendo, seguramente, que es la ubicación del archivo de las ventas y la descripción de los datos que éste alberga.

Entonces para resumir, ¡El cambio en la programación de la aplicación de ventas ha impactado en la aplicación del reporte y ha provocado que falle!

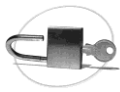
¡Este es precisamente una de las principales motivaciones que impulsó la creación de las bases de datos!

Se crearon para poder tener una independencia entre las aplicaciones o programas y los archivos donde están almacenados los datos (o sea su ubicación física en los discos rígidos).

Otro motivo importantísimo que afectaba la operatoria en la época del procesamiento de archivos era la imposibilidad de poder acceder al archivo al mismo tiempo para operar, en nuestro ejemplo, ambas aplicaciones. Es decir, si mientras se estaba obteniendo el reporte, leyendo el archivo de ventas, un usuario del sistema de ventas trataba de ingresar una venta, dicho sistema arrojaba un error de acceso indicando que el archivo estaba siendo abierto por otro usuario. Es decir que **el procesamiento era monousuario, no multiusuario**.



2. ¿QUÉ ES UN SISTEMA GESTOR DE BASE DE DATOS (O DBMS EN INGLÉS)?



ABORDEMOS EL LOGRO DEL SEGUNDO OBJETIVO DE LA UNIDAD:

- ✓ Analizar cuál es la utilidad de un DBMS.

Hoy es imposible imaginar una aplicación que no esté interactuando con una base de datos.



Veamos en el campus el video “Introducción a las Bases de Datos” donde podrá aprender qué utilidad tienen para las empresas.

Desde los sistemas de comercio electrónico hasta la tarjeta para viajar en un medio de transporte, todas estas aplicaciones utilizan bases de datos. Son tan comunes que ni siquiera nos damos cuenta de que las estamos utilizando.



Pero entonces veamos la definición, ¿qué es en realidad una base de datos?

Una base de datos es una colección de hechos con un significado relevante para el mundo con el que nos interesa trabajar y que tiene un propósito específico.

Ahora bien, usted se podría preguntar lo siguiente: una lista de supermercado en una planilla de cálculo, el conjunto de las inscripciones a una materia en una universidad, ¿se pueden considerar como bases de datos?

Para responder a su pregunta basta repasar la definición, ¿Tienen un significado? ¿Nos interesan para trabajar con ellos? ¿Tienen un propósito? ¡Entonces, sí constituyen una base de datos!

Por lo tanto, no cualquier conjunto de datos se puede considerar como una base de datos, el conjunto debe reunir las condiciones mencionadas; sino, solo es un conjunto de datos.

Una particularidad que deben tener las bases de datos es que su contenido debe reflejar los cambios realizados en los datos, a través de algún medio, tan pronto como sea posible para que su contenido se mantenga fiel a la realidad que representa.

Y entonces, comenzamos a hablar de DBMS, pero ¿qué es un DBMS o motor de la base de datos?

Un DBMS o motor de una base de datos es una colección de programas que permite a los usuarios crear y mantener una base de datos. Es un sistema de software de propósito general que facilita la definición, construcción, manipulación y compartición de bases de datos entre usuarios y aplicaciones.

A continuación, veamos una lista de las posibilidades y ventajas que ofrecen las bases de datos:

- El Sistema de base de datos es capaz de describirse a sí mismo a través de su meta-data (colección de datos que sirve para definir las estructuras básicas de los datos de la base)

de datos), pudiendo entonces servir para gestionar a la propia base de datos. Es decir, en cierta forma lo podemos ver como que las bases de datos se gestionan a sí mismas.

- Aislamiento entre programas y datos, y abstracción de datos: así como la necesidad del aislamiento es lo que originó la idea de crear los DBMS, la abstracción de datos es una técnica que permite definir, diseñar o representar las estructuras escondiendo ciertos detalles de implementación, como por ejemplo dónde está guardado físicamente dicho archivo.
- Soporte de múltiples vistas de los mismos datos: es decir, por ejemplo, que el personal del departamento de Recursos Humanos pueda ver todos los datos de los empleados, pero el coordinador del departamento de Ventas pueda ver solamente el nombre y el teléfono celular de los empleados de su propio departamento.
- Datos compartidos y procesamiento multiusuario: debe permitir que los usuarios tengan disponibles “múltiples vistas de los datos” es decir, ver los datos de diferentes maneras o sea de “diferentes vistas”. Algunos, por ejemplo, podrían necesitar ver los sueldos y otros no tendrán permitido verlos.

Además, el DBMS deberá tener algún “mecanismo de control de concurrencia”, ya que debe permitir que muchos usuarios accedan al mismo tiempo a trabajar con los datos. O sea, algunos podrían estar generando un reporte mientras otros están agregando nuevos datos (por ejemplo, ingresando una venta). El DBMS debe permitir que ambos usuarios trabajen y que los datos sean consistentes para ambas acciones.



AUTOEVALUACIÓN 1

Le pedimos que realice la autoevaluación que se encuentra en el campus de manera de ir corroborando la comprensión de los conceptos vistos.



3. LOS DIFERENTES TIPOS DE USUARIOS

Retomando el ejemplo de la aplicación de comercio electrónico, imaginemos por un momento que un usuario va a realizar una compra. Por lo tanto, ingresa con su usuario y contraseña, elige el producto que desea y presiona el botón “Comprar”



En ese momento puede elegir la forma de pago y se le pedirá que ingrese los datos de su tarjeta y banco.

Toda esta información que se está mostrando es propia del usuario que está conectado en ese momento y no deben verla otros usuarios. En las bases de datos existen mecanismos para garantizar la privacidad de los usuarios. Usted se imaginará, a estas alturas, que para poder implementar esos mecanismos tiene que haber “alguien” con permiso para hacerlo.

Por lo tanto:

En una base de datos existen diferentes tipos de usuarios de acuerdo a las actividades que pueden realizar en ella:

- Administrador (o DBA): es el usuario con más poderes dentro de la base de datos. Se ocupa de:

- administrar los permisos de acceso a los demás usuarios
 - que el DBMS funcione correctamente
 - analizar el rendimiento (o velocidad) del motor para que los usuarios puedan trabajar en los niveles esperados.
 - administrar los recursos necesarios para asegurarse que tanto las bases de datos como el DBMS se comporten de acuerdo a los niveles previamente determinados.
- Diseñador: es el usuario que trabajó desde el inicio, mucho antes incluso de la existencia de la base de datos misma. El desarrolló los modelos de datos de acuerdo al requerimiento del cliente (el dueño del mini-mundo que se necesitaba modelar) y también determinó cuáles son las diferentes vistas a las que necesitarán acceder los usuarios.
 - Usuarios Finales: son los usuarios que van a interactuar con las diferentes vistas de la base de datos y para los cuales se la construyó.
 - Analistas y Programadores de Aplicaciones: Los analistas son los que realizaron el relevamiento de las necesidades que debía cubrir la base de datos y los programadores de aplicaciones crearon los programas que garantizaron que dichas necesidades pudiesen cumplirse.



¿Se imagina un ejemplo de cada uno de los diferentes niveles que tienen las bases de datos de acuerdo a los usuarios que pueden acceder a ellos?

Quizás le será más fácil identificar los niveles si piensa hasta qué profundidad de detalle podrá ver cada tipo de usuario los datos de la base:

En la Figura 1 se puede ver la “arquitectura de tres esquemas” de la base de datos. La misma permite:

- 1) Aislamiento de los programas y los datos (independencia programa-datos y programa-operación)
- 2) Soporte de varias vistas de usuario
- 3) Uso de un catálogo para almacenar la descripción de los datos (esquema)

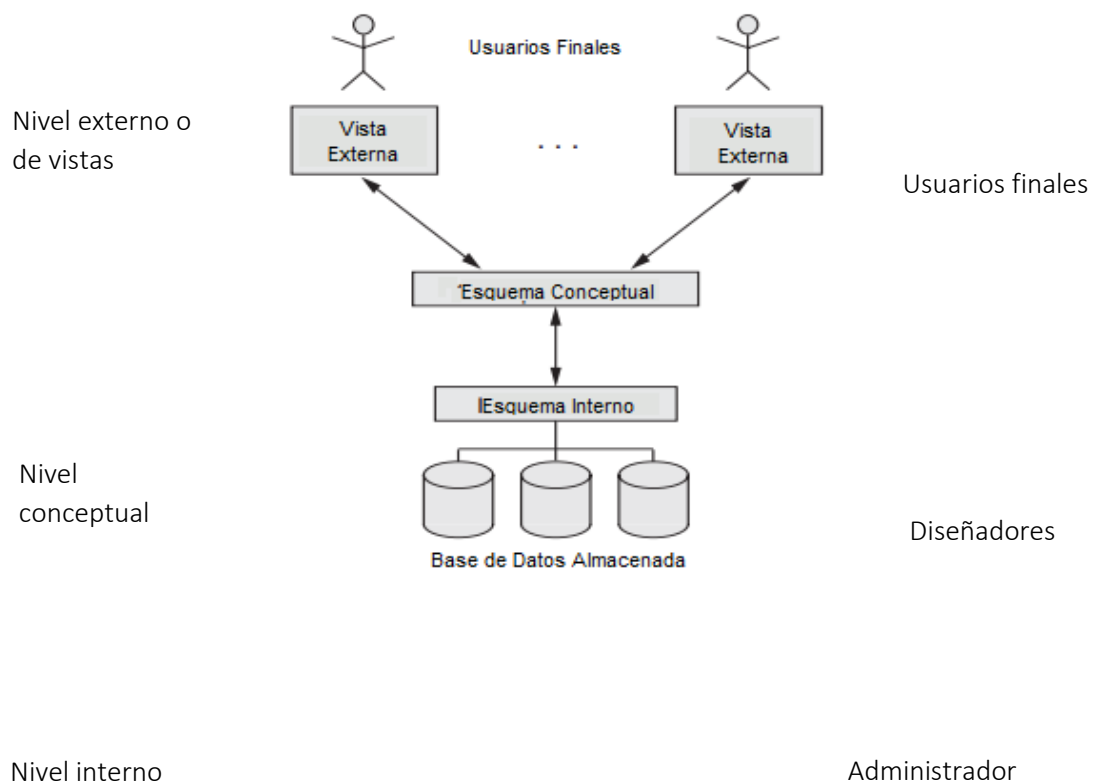


Figura 1 – Arquitectura de tres Esquemas

Cada nivel oculta hacia arriba algunas características del esquema, así el nivel interno expone tanto el modelo físico como las rutas de acceso, de esta manera es posible para el administrador trabajar con ellos para mejorar los rendimientos de los accesos a los datos.

Por otro lado, el nivel conceptual permitirá ver la estructura de toda la base para la comunidad de desarrolladores, pero oculta todo el modelo físico y de almacenamiento. Esto es porque no son los desarrolladores a quienes les concierne cómo se organiza la parte física de la base de datos. Piense en la parte física como los archivos que están en los discos rígidos, donde se guardan los datos propiamente dichos.

Y finalmente, en el nivel externo, solo podrán ver los usuarios finales aquellas vistas que estén disponibles para ellos, en algunos casos de manera personalizada o por grupos según su interés.

Pero se les ocultará tanto la parte física como la conceptual.



4. INDEPENDENCIA DE LOS DATOS



ABORDEMOS EL LOGRO DEL TERCER OBJETIVO DE LA UNIDAD:

- ✓ Conocer el significado del aislamiento y la abstracción en el contexto de un DBMS.

Imaginemos que estamos trabajando y se solicita agregar un dato adicional a los clientes: un cálculo de una métrica que afecta luego su scoring o sea un número que indica cómo es visto por el negocio (dato preexistente calculado). Entonces el desarrollador, agrega en el esquema la nueva métrica haciendo un cambio en el nivel conceptual pero no debe cambiar nada en los archivos físicos o en las vistas.

La independencia de los datos puede verse desde dos puntos de vista: lógico y físico.

Lógico sería que se cambia el modelo que se hizo originalmente para reflejar algo nuevo. Este cambio sin embargo no impacta en cómo se organiza desde el punto de los archivos donde se guardan los datos.

Y a continuación la definición:

La independencia lógica de los datos es la capacidad de cambiar el esquema conceptual sin tener que cambiar los esquemas externos o los programas de aplicación.

Físico sería lo contrario, es decir cambiamos algo en la forma en que guardamos los datos en el/los archivo/s del disco; por ejemplo, agregamos un archivo nuevo o un disco nuevo a la estructura existente. Pero al hacer este agregado, no hemos cambiado nada en nuestro diseño conceptual original.

Y a continuación su definición:

La independencia física de los datos es la capacidad de cambiar el esquema interno sin tener que cambiar el esquema conceptual. Por lo tanto, no es necesario cambiar el esquema externo.



A estas alturas usted se preguntará: ¿cómo es posible cambiar algo en un esquema sin modificar los otros dos? Esta tarea es una de las tareas más frecuentes del DBA, por ejemplo, cuando los discos están llegando a su capacidad debido al crecimiento de los archivos de datos. El DBA agrega archivos adicionales en otro disco sin tocar nada del diseño.

¡Esto lo puede realizar incluso mientras la base de datos está operando y ninguno de los usuarios se ve afectado por dicha actividad, lo que prueba que, en efecto, la independencia física funciona perfectamente!



¿Se le ocurren a usted más ejemplos de actividades que realicen los diferentes usuarios en sus respectivos niveles y que no afecten a los demás niveles?



AUTOEVALUACIÓN 2

Le pedimos que realice la Autoevaluación 2 que se encuentra en el campus de forma de ir corroborando los conceptos vistos hasta aquí.



5. LENGUAJES DBMS

Una vez que se ha realizado el diseño de la base de datos, luego de elegir el DBMS apropiado, se debe implementar la misma, usando los lenguajes propios del DBMS.

Para realizar el diseño, se utiliza el “lenguaje de definición de datos” o DDL por sus siglas en inglés. Mediante este lenguaje es que se crea la base de datos, todas las estructuras y sus objetos.



Usted recordará que en Programación Básica había visto la función de un compilador y de un traductor. En este caso el DBMS utiliza un compilador DDL que interpreta dicho lenguaje y procesa las sentencias.

Hay motores de bases de datos que tienen implementado un lenguaje para cada nivel, en esos casos existe el lenguaje SDL o “lenguaje de definición del almacenamiento” (con S de storage) con el cual se especifica el nivel interno. Con este lenguaje es que crearíamos los archivos de la base de datos y definiríamos qué objetos van en cada uno. Es decir que nos basamos en el esquema conceptual para especificar cómo y dónde quedarán los archivos donde estarán los datos, estos archivos conforman lo que llamamos la parte física de la base de datos.

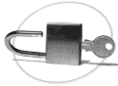
Y por último se utiliza el DML o “lenguaje de manipulación de datos” para especificar las vistas de los usuarios finales y los diferentes mapeos a los niveles conceptual y exterior.

En los DBMS actuales, el lenguaje utilizado es el mismo, y son las sentencias las que realizan las diferentes tareas. Estas sentencias se clasifican en sentencias DML, DDL y SDL.

Puede ver entonces que con estos lenguajes se especifican primero los esquemas conceptual e interno y el mapeo entre ambos, y luego recién es posible trabajar en la especificación del esquema externo. No podría hacerse en otro orden. Usted lo debe pensar como cuando se construye una casa, primero se hacen los planos, luego, de acuerdo a ellos se comienza a construir, primero los cimientos y las columnas que van a soportar las paredes y el techo. Una vez hecho todo esto, se puede pintar y adornar cada habitación para que se vea linda estéticamente.



6. MÓDULOS COMPONENTES DE UN SISTEMA DE BASES DE DATOS



ABORDEMOS EL LOGRO DEL CUARTO OBJETIVO DE LA UNIDAD:

- ✓ Analizar la arquitectura de un DBMS y sus componentes más importantes.

En los apartados anteriores hemos visto algunos componentes de las bases de datos y de los DBMS, utilizando ejemplos para que puedas darte una idea de su funcionalidad y cómo interactúan entre ellos.

Teniendo en cuenta todo lo trabajado hasta ahora, vamos a ver un diagrama que al principio puede resultar un poco complejo de entender, pero si lo vemos por partes, claramente podremos relacionar cada una de ellas con los temas vistos.

Analicemos el diagrama simplemente como una forma de “atar cabos”, o sea tener un contexto común a todos para que puedas ver en el DBMS cómo contribuye cada uno de ellos.

Así en la Figura 2, podemos observar que el sistema de base de datos o DBMS está formado por una serie de componentes interdependientes entre sí.

En la parte superior, se puede ver cómo se relacionan los esquemas con los diferentes niveles de usuarios. Así usted notará que el DBA interactúa con lenguaje DDL y comandos privilegiados, y que los programadores de aplicaciones interactúan con el lenguaje DML, por ejemplo.

El programador, se encarga de armar las consultas que luego se ejecutarán cuando los usuarios interactúen a través de la aplicación.

El DBA tiene acceso a **comandos privilegiados** para realizar sus actividades de administración

El compilador de lenguaje DDL procesa las definiciones del lenguaje y guarda en el catálogo los metadatos generados. El compilador DML recibe los comandos y los transforma en código objeto que accede a la base de datos.

El catálogo del sistema y la base de datos se almacenan en el disco. El acceso al disco es controlado por el sistema operativo. El módulo administrador de los datos almacenados del DBMS controla el acceso a la información del DBMS almacenada en disco (catálogo o base de datos).

El compilador de consultas analiza sintácticamente los comandos escritos para verificar que los objetos y operaciones existan en el catálogo y que las sentencias estén bien formadas.

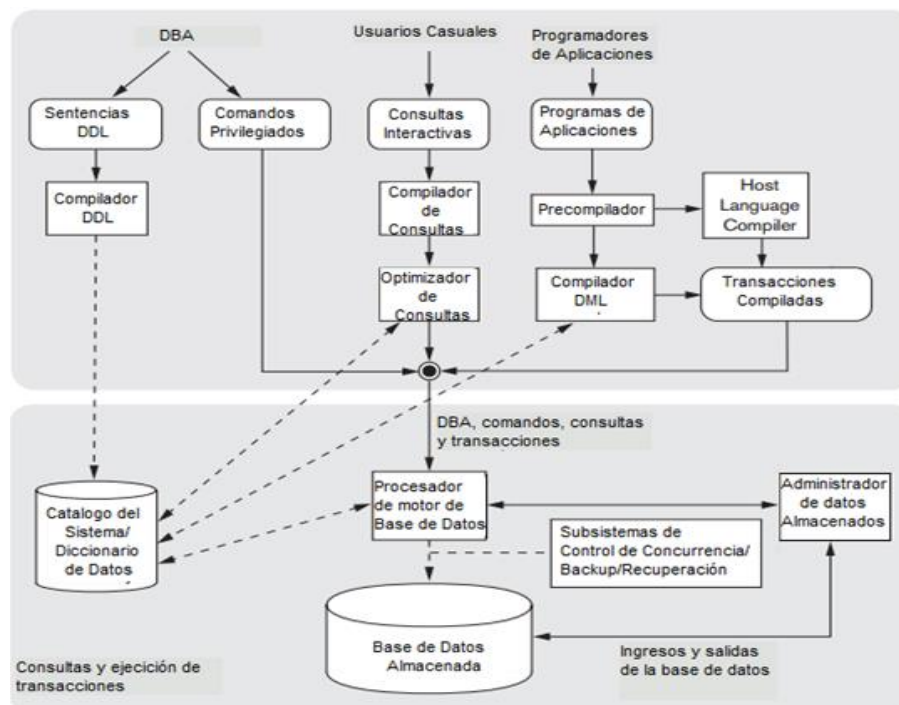


Figura 2. Componentes del DBMS

Los subsistemas de control de concurrencia se ocupan de asegurar que varios usuarios puedan operar al mismo tiempo sin interferir entre ellos. O sea, en el ejemplo de la aplicación de comercio electrónico, podrían comprar al mismo tiempo el mismo o diferente producto.



AUTOEVALUACIÓN3

Le pedimos que realice la Autoevaluación 3 que se encuentra en el campus de forma de ir corroborando los conceptos vistos.



Si usted ha logrado realizar las actividades con éxito, seguramente ya podrá:

- ✓ Conocer qué es un DBMS o Database Management System.
- ✓ Analizar cuál es la utilidad de un DBMS.
- ✓ Conocer qué significa el aislamiento y abstracción en el contexto de un DBMS.
- ✓ Analizar la arquitectura de un DBMS y sus componentes más importantes

¡FELICITACIONES!

Ha logrado alcanzar los objetivos propuestos para esta unidad y podrá pasar a la siguiente



SÍNTESIS DE LA UNIDAD

En esta Unidad vimos los conceptos básicos que debe conocer antes de comenzar a trabajar con modelado de datos y bases de datos. Estos conocimientos podrás utilizarlos en la Unidad 2 para pensar los modelos y en la Unidad 3 cuando comencemos a usar las sentencias de los lenguajes DML y DDL.

Hemos visto que generalmente un cliente solicita a un consultor realizar un modelo de un sistema de su interés y crear una BASE DE DATOS que refleje el MINI-MUNDO de su interés.

A partir de este modelo, el DISEÑADOR crea la BASE DE DATOS y el ESQUEMA CONCEPTUAL utilizando lenguaje DDL.

Luego el DESARROLLADOR accederá a esta BASE DE DATOS y creará los programas y el ESQUEMA EXTERNO al que accederán los usuarios.

Él utilizará lenguaje DML dentro de sus programas lo que permitirá acceder a los datos almacenados dentro de la BASE DE DATOS.

Por último, el DBA será el encargado de administrar el ESQUEMA INTERNO de la base de datos utilizando COMANDOS PRIVILEGIADOS.

Veamos el siguiente mapa conceptual donde se encuentran los conceptos más importantes de la unidad y sus relaciones.

