

## **Trabajo Practico N°1**

### **Introducción**

#### **Ejercicio N°1:**

##### **Datos persistentes:**

Un dato persistente es un dato que tiene una naturaleza no transitoria y que, por lo tanto, debe ser almacenado. Esta característica lo diferencia de otro tipo de datos, de naturaleza efímera o transitoria, como los datos de entrada-salida, las proposiciones de control, las colas de trabajo, los bloques de control de un programa, los resultados intermedios y, en general, cualquier información transitoria.

Un dato de entrada es el que se recibe por primera vez en el sistema y puede dar pie a la modificación de los datos persistentes o convertirse en uno de ellos. Un dato de salida es uno que se deriva de los datos persistentes.

##### **Bases de Datos:**

Una base de datos es un conjunto de los datos persistentes de una organización, relacionados entre sí.

De la definición podemos extraer las propiedades de una base de datos:

- Una base de datos representa algún aspecto del mundo real y las modificaciones que sufre ese mundo real.
- Una base de datos es un conjunto de datos lógicamente coherente, con significado inherente.
- Toda base de datos se construye y puebla con datos para un propósito específico, está dirigida a un grupo de usuarios y tiene aplicaciones que les interesa a ellos.

##### **Sistemas de Bases de Datos:**

Definimos a un sistema de bases de datos como un sistema cuyo propósito general es mantener datos y hacer que estén disponibles cuando sean solicitados. Un sistema de bases de datos se apoya en cuatro componentes: Datos, Equipo, Software y Usuarios.

##### **Datos:**

Tienen dos características esenciales:

1. Están integrados: La base de datos se considera como la unificación de archivos de datos.
2. Serán compartidos: Los elementos individuales de la base de datos serán accedidos por distintos usuarios en el sentido de que todos pueden tener acceso al mismo elemento para usarlo con propósitos diferentes.

### Equipo:

Es el soporte para el almacenamiento de los datos. Por ello, nos interesa específicamente:

1. los volúmenes de almacenamiento secundario.
2. los dispositivos de entrada-salida asociados.
3. el procesador o procesadores.
4. la memoria principal asociada.

### Software:

Es el conjunto de programas creado para la gestión de los datos almacenados y consiste en programas de aplicación y consulta a la base de datos. Estas aplicaciones y consultas no tienen una interacción directa con la base de datos almacenada.

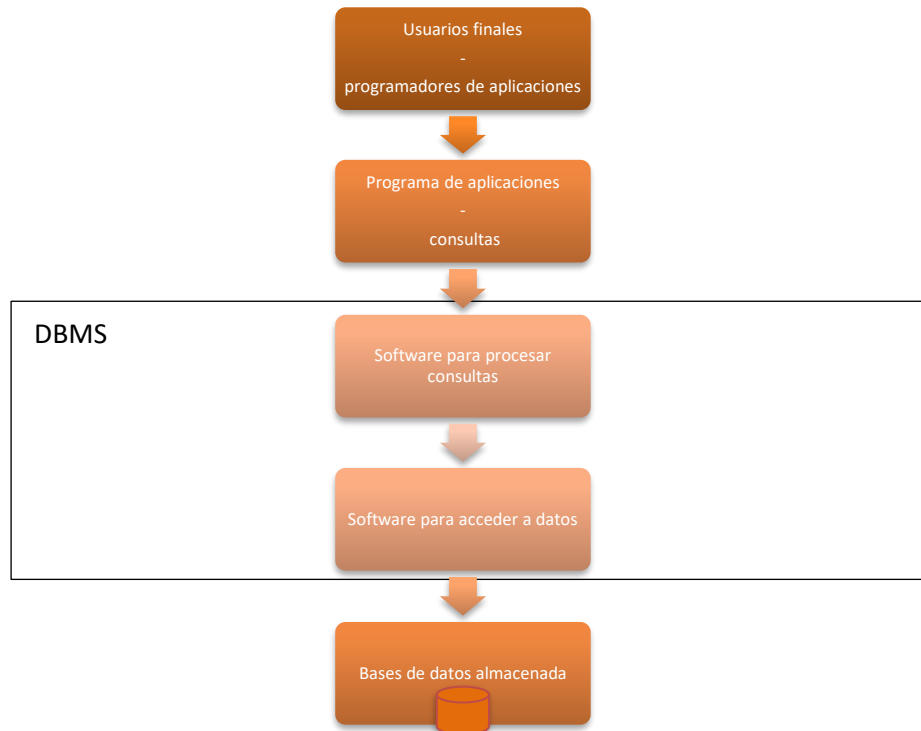
Existe, específicamente, un conjunto de programas que permite a los usuarios crear y mantener una base de datos. Este software de propósitos generales, facilita el proceso de definir, construir y manipular una base de datos, se denomina DBMS: sistema administrador de la base de datos, o simplemente motor de base de datos y su finalidad es distanciar a los usuarios de los detalles a nivel equipo. El DBMS es, definitivamente, la componente de software más importante de todo el sistema.

### Usuarios.

Se tomarán en cuenta dos clases de usuarios:

1. Programador de aplicaciones: es el encargado de escribir los programas que utilizarán la base de datos, dirigiendo las solicitudes al DBMS.
2. Usuario final: tiene acceso a la base de datos a través de las aplicaciones efectuadas por el programador de aplicaciones o a través de una aplicación integrada al sistema de base de datos, que es un procesador de un lenguaje de consulta interactivo, mediante el cual el usuario puede formular mandatos de alto nivel.

## Ejercicio N°2: Esquema de un sistema de bases de datos



## Ejercicio N°3: Fundamentos del enfoque de bases de datos

- 1. Minimización de la redundancia.** La redundancia es la repetición del mismo dato en lugares diferentes. Lo deseable es la eliminación de la redundancia. Pero si, por la naturaleza de la aplicación, ésta debe existir, debe ser mínima y el DBMS debe tener conocimiento de su existencia, para que pueda propagar las actualizaciones. Esto es, la actualización de un dato redundante en cualquier lugar, implica la actualización automática del mismo dato en todos los lugares en donde se encuentre.
- 2. Eliminación de la inconsistencia.** La inconsistencia es la falta de concordancia en los valores de un mismo dato. Cuando no hay redundancia o cuando la redundancia está controlada por el DBMS, es posible eliminar la inconsistencia.
- 3. Compartimento de los datos.** Los datos son compartidos por diversas aplicaciones y también están disponibles para nuevas aplicaciones, sin necesidad de almacenar datos adicionales.
- 4. Cumplimiento de normas.** Es posible representar los datos de acuerdo a un conjunto de normas preestablecidas. Esta normalización es deseable para el intercambio de datos o para la migración de datos de un sistema a otro. También es posible definir y mantener la seguridad en el acceso a los datos de manera que cada usuario sólo perciba los datos destinados a él.
- 5. Mantenimiento de la integridad.** La integridad de los datos es la posibilidad de asegurar que los datos contenidos en la base de datos son correctos. La falta de consistencia deriva en la no

integridad, pero no es el único caso. En un enfoque de bases de datos es posible definir restricciones de integridad como, por ejemplo, verificaciones de las actualizaciones.

6. Equilibrio de requerimientos opuestos. Un requerimiento opuesto son diferentes peticiones de los usuarios a la misma base de datos. Se pueden estructurar los datos de forma de proporcionar un servicio óptimo al usuario, aunque la naturaleza de las peticiones de estos sea radicalmente diferente.

7. Autodescripción de los datos. En un catálogo del sistema se almacena información sobre la estructura de cada archivo, el tipo de cada elemento de dato y las restricciones de integridad. Esta información almacenada se denomina metadatos.

#### **Ejercicio N°4: Indicar la diferencia entre redundancia controlada y no controlada**

La redundancia controlada es aquella que se permite para efectos de mejorar el rendimiento (respuestas rápidas en sistemas distribuidos, por ejemplo), pero el DBMS debe ser capaz de llevar un control de tal manera que no hayan inconsistencias. La redundancia no controlada, sin embargo, se produce sin tener conocimiento de que está ocurriendo, por ejemplo, en el sistema tradicional de archivos, cuando cada usuario crea sus propias aplicaciones con archivos de datos, lleva a la inconsistencia de datos, puesto que distintos archivos con los mismos datos presentarán distintas realidades. Otro ejemplo: En una pizzería se almacenan los datos del cliente y su domicilio, como puede haber varios clientes en el mismo domicilio se hace una redundancia controlada en el domicilio pero si es descontrolada habrá errores en la actualización.

#### **Ejercicio N°5: Dependencia de datos y dependencia de operaciones**

En el procesamiento de archivos tradicional, la estructura de los archivos de datos viene integrada en los programas de acceso. Así, deberemos implementar todas las aplicaciones que acceden a los datos utilizando la misma herramienta. Por ejemplo, el muy usado gestor de archivos Fox maneja una estructura de archivos conocida como archivos DBF. Para gestionar los datos es necesario programar aplicaciones en Fox u otra herramienta que disponga de un driver especial para tratar este tipo de archivos.

En cambio, en un sistema de bases de datos, la estructura de los archivos de datos se almacena en el catálogo, aparte de los programas de acceso. Por lo tanto es posible realizar aplicaciones que accedan a la base de datos en cualquier herramienta. Por ejemplo, se podrían almacenar los datos utilizando un motor MS SQL Server y acceder a ellos mediante aplicaciones realizadas en Visual Basic, Java, Delphi, C++, etc.

Esta propiedad se denomina independencia de los datos.

Además, el enfoque orientado a los objetos permite a los usuarios definir operaciones sobre los datos. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando estas operaciones a través de sus nombres y argumentos, independientemente de la forma en que están almacenadas. Además, estas operaciones podrían redefinirse sin que sea necesaria una modificación en la aplicación que las invoca.

### **Ejercicio N°6: Definir funciones del DA y del DBA**

#### Administrador de datos (DA)

Un sistema de bases de datos ofrece a la organización un control centralizado de sus datos. Pero es la organización la que deberá definir la política a seguir con sus datos. Dentro de la organización, entonces, deberá existir una persona que no necesariamente conozca el aspecto informático pero que ocupe un nivel gerencial superior y que deberá decidir cuáles son los datos que deberán ser persistentes y quiénes pueden acceder a los datos. Esta persona se conoce como el Administrador de los Datos.

#### Administrador de base de datos (DBA)

El DA es un gerente, no necesariamente un técnico. El técnico responsable de poner en práctica las decisiones del DA es el Administrador de la Base de Datos. El DBA es un profesional en el procesamiento de datos, cuyas funciones son:

- Crear la base de datos
- Implementar los controles técnicos definidos por el DA,
- Garantizar el funcionamiento adecuado del sistema,
- Adquirir los recursos necesarios de software y de hardware.

## **Sistemas Administradores de Bases de Datos**

### **Ejercicio N°1: Definir Sistema Administrador de la Base de Datos y enunciar sus características distintivas.**

El DBMS es un conjunto de programas que maneja todo el acceso a la base de datos. Esto lo hace siguiendo ciertas pautas y normas que definen un proceso como el siguiente:

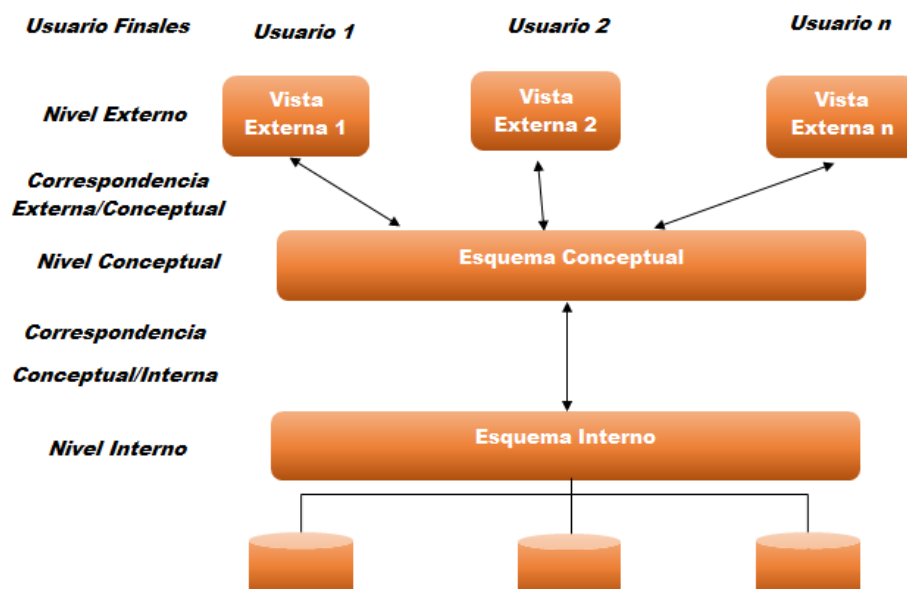
1. Un usuario solicita un acceso.
2. El DBMS interpreta y analiza la solicitud.
3. El DBMS consulta el catálogo almacenado.
4. El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenada.

Las pautas y normas que intervienen son las que aseguran el desempeño eficiente del DBMS en cuanto a tiempo de respuesta. Estas pautas y normas contienen referencias específicas a dos aspectos importantes:

Integridad: el DBMS debe ser capaz de entender y de hacer cumplir un conjunto de restricciones sobre los datos, llamadas restricciones de integridad. Por ejemplo, verificaciones de tipo a un ingreso de datos, verificaciones de relaciones entre datos, verificaciones de valores de los datos. Es de hacer notar que ningún DBMS puede descubrir automáticamente un error. Si el usuario ingresó 10 en lugar de 100 y el tipo de dato coincide y está dentro del rango esperado, el DBMS lo admitirá.

Seguridad: implica el control de los accesos a la base de datos. Este control se hace usualmente asignando contraseñas a los usuarios y/o un sistema de control y revocación de privilegios. Este ultimo Sistema es controlado por el DBMS.

### **Ejercicio N°2: Construir el esquema que representa la arquitectura de tres niveles para un DBMS, describiendo cada uno de dichos niveles.**



**El Nivel Externo:** Incluye varias vistas del usuario, que utiliza la parte de los datos almacenados que es de su interés. Cada esquema o vista externa muestra al usuario los datos que necesita y le oculta el resto.

**El Nivel Conceptual:** Contiene el esquema conceptual, que describe la estructura de toda la base de datos, ocultando los detalles de almacenamiento.

**El Nivel Interno:** Tiene un esquema interno que describe la estructura de almacenamiento de la base de datos.

### **Ejercicio N°3: Clasificar los Sistemas Administradores de Bases de Datos.**

#### **Clasificación de los Sistemas Administradores de Bases de Datos.**

Un DBMS puede clasificarse de acuerdo a varios criterios.

**1-** Según el modelo de datos en que se basa. Un modelo de datos es la forma en que se representan los datos, es una descripción de la realidad con el propósito de estudiarla.

1.1. Relacionales.

1.2. De Red.

1.3. Jerárquicos.

1.4. Orientados a los objetos.

**2-** Según el número de usuarios a los que presta servicio.

2.1. Monousuarios: solo atienden a un usuario a la vez.

2.2. Multiusuarios: atienden a varios usuarios al mismo tiempo.

**3-** Según la cantidad de sitios donde se encuentra almacenada la base de datos.

3.1. Centralizados: la base de datos y el DBMS se encuentran en un solo computador.

3.2. Distribuidos: la base de datos y el DBMS están en varios sitios, conectados por una red.

**4-** Según su propósito.

4.1. De propósito general, se diseñan para cualquier aplicación de gestión de bases de datos.

4.2. De propósito especial, se diseñan para una aplicación específica.

#### **Ejercicio N°4: Definir lenguajes de un DBMS e indicar sus componentes, describiendo la finalidad de cada una de ellas.**

##### **Lenguajes de un DBMS y sus componentes**

Como hemos visto, un DBMS puede ser utilizado por diferentes tipos de usuarios. Por lo tanto, deberá manejar lenguajes e interfaces apropiadas para cada categoría.

Estos lenguajes deberán ser tales que nos permitan las especificaciones de los esquemas externo, conceptual e interno asociados a cada uno de los tres niveles de la arquitectura del DBMS, y las correspondencias entre estos niveles. Para la especificación de las vistas de los usuarios y la correspondencia externa-conceptual, se utiliza un lenguaje denominado VDL: View Definition Language. La especificación del esquema conceptual y de la correspondencia conceptual-interna, se realiza en lenguaje DDL: Data Definition Language. El DBMS cuenta con un compilador cuya misión es procesar los enunciados DDL para identificar la descripción de los elementos del esquema conceptual en el catálogo almacenado. El SDL: Storage Definition Language es utilizado para la especificación del esquema interno y también se puede utilizar este lenguaje para especificaciones de la correspondencia conceptual-interna, tal como el DDL. Una vez que se han compilado los esquemas de la base de datos y se han introducido los datos, se requerirá de algún mecanismo que permita la manipulación de los datos persistentes. Las operaciones de manipulación más comunes son la obtención, inserción, eliminación y modificación de los datos almacenados. Para tal propósito, el DBMS ofrece el DML: Data Manipulation Language.

#### **Ejercicio N°5: Definir interfaces de un DBMS, describiendo cada uno de los tipos.**

##### **Interfaces de un DBMS y sus tipos**

Como dijimos, los DBMS actuales no distinguen una clara separación entre los niveles, por ello les basta con un DDL y de un DML para las especificaciones de los tres esquemas y lo ofrecen en forma de un lenguaje integrado

Con respecto a las interfaces, el lenguaje de un DBMS ofrece las siguientes:

1. Interfaces basadas en menús desplegables: presentan al usuario una lista de opciones que lo guían en la formulación de solicitudes. La ventaja de este tipo de interfaz es que brinda al usuario la posibilidad de formular peticiones sin necesidad de memorizar la sintaxis específica para cada una.
2. Interfaces gráficas: presentan al usuario los esquemas en forma de diagrama y le permite la selección de la parte del diagrama que se exhibe.
3. Interfaces basadas en formas: presentan al usuario un formulario estandarizado en el cual es posible agregar datos nuevos y recuperar datos persistentes.
4. Interfaces de lenguaje natural: aceptan solicitudes e intentan entenderlas a través de un análisis sintáctico y léxico de las mismas. Si la interpretación tiene éxito, la interfaz genera una



consulta de alto nivel y la envía al DBMS para que se encargue de su procesamiento, de lo contrario inicia un diálogo con el usuario para esclarecer la petición. 5. Interfaces para usuarios paramétricos: son un conjunto pequeño de operaciones, como las disponibles en el cajero de un banco. Este tipo de interfaces son diseñadas por los analistas del sistema para una clase de usuarios simples. 6. Interfaces para el DBA.: especiales para éste, como órdenes para crear cuentas, asignar niveles de seguridad, modificar esquemas, parametrizar el sistema y reorganizar la estructura de almacenamiento.

**Ejercicio N°6: Efectuar el esquema del proceso de diseño de una base de datos, haciendo una breve referencia a cada una de las fases.**

### Esquema del proceso de diseño de una base de datos

- El proceso de diseño de una base de datos consta de seis fases:
- Fase 1: recolección y análisis de requerimientos.
- Fase 2: diseño conceptual de la base de datos.
- Fase 3: elección del DBMS.
- Fase 4: diseño lógico de la base de datos.
- Fase 5: diseño físico de la base de datos.
- Fase 6: implementación de las aplicaciones.

