**SISTEMAS DE BASE DE DATOS**:

Un sistema de base de datos es un sistema computarizado cuyo propósito general es mantener información, además de permitir a los usuarios recuperarla y actualizarla cuando se lo necesite. Esta información es cualquier cosa necesaria para apoyar el proceso general de atender los asuntos de esa organización o individuo. Este sistema está formado por cuatro componentes principales: la información, el equipo, los programas y los usuarios.

INFORMACION: Estos pueden estar integrados y compartidos, en sistemas monousuario o multiusuario. Integrada significa que la base de datos puede considerarse como la unificación de varios archivos de datos, donde se elimina del todo o en parte cualquier redundancia entre ellos. Compartidos significa que los elementos individuales de información en la DB pueden compartirse entre varios usuarios distintos, en el sentido de que todos los usuarios puedan tener acceso al mismo elemento de información, por ejemplo los datos del archivo *Alumnos* para el sistema de biblioteca estarán compartidos por el Departamento de Alumnos.

EQUIPO: Son aquellos volúmenes de almacenamiento secundario, junto con dispositivos IO, además de la memoria, procesador, etc.

PROGRAMAS: Entre la base de datos física misma (es decir, los datos tal y como están almacenados en realidad), y los usuarios del sistema, existe una capa de software conocida como *Sistema de administración de base de datos*, DBMS. Este maneja todas las solicitudes de acceso a la base de datos formulada por los usuarios, además de distanciar a los usuarios de la base de datos al nivel del equipo (como los lenguajes de alto nivel evitan la necesidad de ocuparse de detalles de nivel de maquina).

USUARIOS: Existen tres clases amplias de usuarios:

*Programador de aplicaciones*: es quien se encarga de escribir los programas de aplicación que utilizaran la base de datos, estos programas operan sobre los datos (inserción de información nueva, modificación, recuperación y eliminación de datos existentes). Donde todas estas funciones se llevan a cabo dirigiendo las solicitudes apropiadas al DBMS.

*Usuario final*: es el que interactúa con el sistema, obteniendo acceso a la base de datos a través de las aplicaciones desarrolladas por los programadores de aplicación. Estas aplicaciones suelen ser amigables y fáciles de usar para aquellas personas sin conocimiento de base de datos,

*Administrador de base de datos*: es el técnico responsable de poner en práctica las decisiones del DA[[1]](#footnote-1), a diferencia de este -que es un gerente-, el DBA es un profesional en procesamiento de datos, su tarea es crear la base de datos en sí y poner en vigor los controles técnicos necesarios para apoyar las políticas dictadas por el DA, además de encargarse de garantizar el funcionamiento adecuado del sistema y establecer un enlace con los usuarios. El DBA no es solo una persona sino que un equipo de varias personas.

**BASE DE DATOS**:

Una base de datos es un conjunto de datos relacionados entre sí, persistentes en el tiempo que son utilizados por programas de aplicación de una organización. Este se diseña, construye y carga datos para un propósito específico, está dirigida a un determinado grupo de usuarios y la utilizan aplicaciones específicas. La gran ventaja de una base de datos a una empresa es que ofrece un control centralizado de la su información.

¿POR QUE UTILIZAR UNA BASE DE DATOS?

*Es compacto*: no hacen falta archivos de papeles que pudieran ocupar espacio.

*Es rápido*: las computadoras pueden obtener y modificar datos con mucha más velocidad que nosotros, además de poder hacer consultas sin necesidad de búsquedas visuales o manuales que requieren mucho más tiempo.

*Es menos laborioso*: se elimina gran parte del trabajo de mantener archivos a mano, las tareas realizadas por una computadora son mejores realizadas.

*Es actual*: se dispone en cualquier momento de información precisa y al día.

VENTAJAS DE UN SISTEMA DE BASE DE DATOS:

*Es posible disminuir redundancia*: en los sistemas sin base de datos cada aplicación tiene sus propios archivos privados. Esto puede provocar considerable redundancia en los datos almacenados, con el consecuente desperdicio de espacio de almacenamiento.

*Es posible evitar la inconsistencia*: una base de datos es inconsistente si hay dos entradas que representan lo mismo -es decir, hay redundancia- en donde estas no coinciden, en donde no se sabrá cual es el correcto o el más actualizado. Por ende si no hay redundancia, no habrá inconsistencia.

*Es posible compartir los datos*: no solo las aplicaciones ya existentes pueden compartir la información de la base de datos, sino también que aquellas aplicaciones desarrolladas en un futuro puedan podrán trabajar con los mimos datos almacenados.

*Es posible hacer cumplir los estándares*: el DBA al tener un control centralizado de la base de datos, se puede garantizar la observancia de todas las normas aplicadas para la representación de datos. La normalización de formatos de los datos almacenaos es deseables sobre todo como apoyo para el intercambio de información, o migración de los datos entre sistema.

*Es posible aplicar restricciones de seguridad*: el DBA al tener jurisdicción completa sobre la base de datos, este puede asegurar que el acceso a la base de datos sea solo a través de los canales apropiados, además de poder definir las verificaciones de seguridad por realizar cuando se intente acceder a información delicada –estableciendo diferentes verificaciones para cada tipo de acceso (consulta, modificación, etc.)-.

*Es posible mantener la integridad*: el problema de la integridad radica en asegurar que la información de la base de datos sea correcta. Aunque no haya redundancia en la misma, es evidente que la base de datos puede contener información errónea –un control centralizado de la base de datos puede contribuir a evitar este tipo de problemas-

*Es posible equilibrar requerimientos opuestos*: al conocer los requerimientos generales de la empresa –en contraste con los requerimientos de cualquier usuario individual- el DBA puede estructurar el sistema con miras a proporcionar un servicio general.

**INDEPENDENCIA DE LOS DATOS**:

La independencia de los datos es un objetivo primordial de los sistemas de bases de datos. Esta independencia puede definirse como la inmunidad de las aplicaciones ante los cambios en la estructura de almacenamiento y en la técnica de acceso, lo cual indica que las aplicaciones en cuestión no dependen de una estructura de almacenamiento o una técnica de acceso específicas. En un sistema de base de datos no es recomendable tener aplicaciones dependientes de datos por dos razones:

1. Cada aplicación requiere una vista diferente de los mismos datos, por Ej.: existan dos aplicaciones, cada una con su archivo privado *Saldo* en la cual uno lo almacena al valor en binario y otro en decimal. Aunque es posible integrar los dos archivos en uno solo y eliminar la redundancia, uno deberá hacer una conversión de los datos.
2. El DBA debe tener libertad para modificar la estructura de almacenamiento o la técnica de acceso para adaptarlas a cambios en los requerimientos, sin tener que modificar las aplicaciones ya existentes, por Ej.: podrían agregarse datos de otra clase a la base de datos, cambiar prioridades de las aplicaciones.

*Campo almacenado*: es la unidad más pequeña de información almacenada que recibe un nombre. La base de datos incluirá, en la mayor parte de los casos, muchas ocurrencias de cada uno de los diversos tipos de campo almacenado. Por Ej.: nombre, dni, etc.

*Registro almacenado*: es un conjunto de campos almacenados, relacionados entre si, que cuenta con su propio nombre. Una ocurrencia de un registro almacenado está formada por un grupo de ocurrencias de campos almacenados relacionados entre sí. Por Ej.: persona, materia, etc.

*Archivo almacenado*: es el conjunto –con nombre- de todas las ocurrencias de un tipo de registro almacenado.

REQUISITOS PARA LOGRAR LA INDEPENDENCIA DE LOS DATOS:

*Representación de datos numéricos*: un campo numérico puede almacenarse en forma aritmética interna o como cadena de caracteres. El DBA deberá escoger la base apropiada, la escala, el modo y la precisión.

*Representación de datos de caracteres*: puede almacenarse en cualquiera de los tipos de códigos de caracteres.

*Unidades para datos numéricos*: podrían cambiarse las unidades de un campo numérico -pulgadas a cm. por Ej.- durante un proceso de conversión al sistema métrico.

*Codificación de los datos*: en algunas situaciones podría ser conveniente representar datos almacenados mediante códigos –por Ej. Rojo=1, Verde=2-.

*Materialización de los datos*: Casi siempre el campo lógico percibido por una aplicación correspondiente a un campo almacenado único – aunque puede haber diferencias en el tipo de datos o unidades-, en casos así, puede considerarse directo –por Ej. Edad y fecha de nacimiento-. Pero a veces un campo lógico se materializa mediante un cálculo sobre un conjunto de ocurrencias –por Ej. Cantidad total -, siendo este un proceso de materialización indirecto.

*Estructura de los registros almacenados*: podrían combinarse dos registros almacenados existentes para formar uno solo, por Ej. Un registro – nombre, sexo- y otro –fechaNacimiento-. Esto se podría presentar cuando se integran a la base de datos aplicaciones anteriores a su instalación.

**ADMINISTRADOR DE BASE DE DATOS**:

El sistema de administración de la base de datos –DBMS- es el conjunto de programas que maneja todo acceso a la base de datos. Conceptualmente sucede lo siguiente:

1. Un usuario solicita acceso, empleando algún sublenguaje de datos determinado.
2. El DBMS interpreta esa solicitud y la analiza.
3. El DBMS inspecciona, en orden, el esquema externo de ese usuario la correspondencia externa/conceptual asociada, el esquema conceptual, la correspondencia conceptual/interna, y la definición de la estructura de almacenamiento.
4. El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenados.

FUNCIONES DEL DBMS:

1. Administrador de datos: este conoce la información y las necesidades de la empresa. Su tarea principal es decidir qué datos serán almacenados, además de establecer políticas para mantener y manejar los datos almacenados. [↑](#footnote-ref-1)