Beneficios del uso de una Base de Datos:

- <u>Reducción de redundancia</u>: La redundancia no es mala, solo es mala cuando es descontrolada, por eso tratamos de reducirla y no de eliminarla (lo cual es muy difícil).
- <u>Aumenta la integridad</u>: Cuanto menos redundancia la bases de datos es más integra.
- Asigna restricciones de Seguridad: Como estructura lógica el único que se comunica a la Base e Datos es el soft, los usuarios se conectan al motor del soft y no a los datos en sí.

<u>Transacción</u>: Conjunto de condigo que se ejecuta exactamente igual, solo cambian los datos).

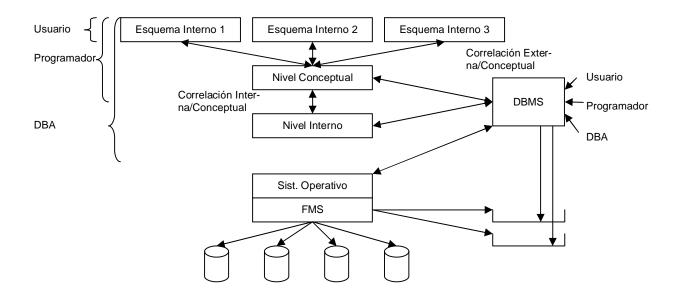
• <u>Facilidad para la modificación de las estructuras</u>: Los datos de las organizaciones son cambiante y por eso me obligado a cambiara esa estructura. Toda Base de Datos tiene una estructura de 3 niveles, un nivel es el externo y es el cual se realizan las modificaciones.

Usuarios de una Base de Datos:

- <u>Usuario</u>: Común y corriente. Se conecta a la base de datos y nada más. Ej.: Aplicación que se conecta y solo hace algo.
- *Programadores*: Conoce una parte de la Base de Datos.
- <u>DBA</u>: Administrador de la Base de Datos. Mucho más técnico, tiene usos de la Base de Datos totalmente distintos a los dos anteriores, por ejemplo le interesa si son muchos los datos, sin son estos son volátiles, etc.

<u>Administrador de datos</u>: No es un usuario en si, en realidad ya no existe. Definía que datos había en la organización, que datos eran públicos y privados, quienes pueden acceder a los datos. Se podía encontrar también en auditorias.

Arquitectura de una Base de Datos:



<u>Nivel interno</u>: Tengo la definición física de la Base de Datos. Doy especificaciones del sistema de datos que estamos usando. Imagen física de la Base de Datos. Tengo la definición de los registros *tal cual* están almacenados.

<u>Nivel externo</u>: Tengo como los usuarios perciben la Base de Datos (no están viendo la Base de Datos en sí, sino con recortes y /o modificaciones). Además puedo mostrarle al usuario la información de la manera que necesite. Me permite definir un lenguaje de Base de datos para un lenguaje particular.

<u>Nivel conceptual</u>: Tengo la definición lógica de la Base de Datos, no me importa como están implementados los datos.

DBA: Funciones que tiene un DBA:

- a) Define una Base de Datos desde el punto de vista del esquema interno y el esquema conceptual en función de las instrucciones y del análisis que se haya realizado. Esto lo hace mediante un DDL que es un lenguaje de definición de datos propio de la Base de Datos. Define tanto el esquema interno como el conceptual.
- b) Relacionarse con los usuarios.
- c) Implementar las políticas de seguridad e integridad: Solo implementa políticas no las define. Las políticas de seguridad se hacen mediante una DSL y las políticas de integridad mediante un DIL.
- d) Implementar políticas de Back-up y Recovery.
- e) Mantener y mejorar el rendimiento/cambios.

Cosas que debe cumplir un DBMS para ser un DBMS

- <u>Debe poseer un DDL (create)</u>: Dado un conjunto de secuencias o archivos sea capaz de entenderlas y ejecutarlas.
- <u>Debe poseer un DML (execute, update, delete)</u>: Lenguaje de manipulación de datos, es el que se utiliza para actuar sobre los datos también debe ser capaz de ejecutar un modulo de ejecución de esas sentencias.
- <u>Debe poseer un DSL y un DIL</u>: Módulos para chequear la seguridad e integridad.

Obligatoriamente un DBMS tiene que recordar DDL, DSL y DIL.

<u>Catalogo del sistema de Base de Datos</u>: Tenía una descripción de la Base de Datos. Es una Base de Datos relacional y esta dentro de la Base de Datos. Contiene la metadata, datos acerca de los datos, del sistema.

Debe poseer herramientas para -Concurrencias (bloqueo lógico y físico)

-Transacciones (muy complejas de resolver)

El bloque físico bloquea también el pedazo físico. Ejemplo: Godio – Godoy

<u>Sección frontal de una Base de Datos</u>: Formada por los programas que tiene como destino al usuario o programador. No tiene una función administrativa. Ejemplo: Programa generador de reportes. Trabaja mediante un ODBC, capa de software. Ejemplo: Generador de código, interfaces de consulta. Todos están van a interactuar con el DBMS.

<u>Sección posterior de una Base de Datos</u>: formada por herramientas de administración que proveen los propios fabricantes. Rutinas de carga: herramientas disponibles en casi todos los motores, para cargas masivas. No la puedo usar en cualquier Base de Datos, debe estar en estado de 'carga' o 'load', solo pueden acceder los DBA.

Rutinas de descarga y de recarga:

- Rutina de descarga: Saca información de la Base de Datos. Solo controla la seguridad, que tenga el permiso necesarios para acceder a esa información.
- <u>Rutina de carga</u>: Es una rutina de 'load' chequeando todos los controles de seguridad e integridad, a diferencia del anterior no requiere que la tabla este vacía.

PORCENTAJE DE LLENADO

Herramientas

- <u>Rutinas de indexación y re indexación de índices</u>: Permiten crear un índice nuevo o hacer una reorganización del mismo.
- <u>Rutinas de reorganización en general</u>: Usadas para datos e índices, su objetivo es no detener la Base de Datos, sino hacer la reorganización mientras los usuarios la están usando.
- <u>Herramientas para hacer Back-up's</u>: Para hacer back-up en caliente, cuando las Bases de Datos están funcionando (generalmente externo)
- <u>Manejo de estadísticas</u>: Un DBA mantiene sus bases y se maneja con estadísticas.