



### TEMA I SISTEMAS DE MANEJO DE BASE DE DATOS

## INTRODUCCIÓN

Como se viera en **Ingeniería de Software I**, todas las metodologías propuestas para el análisis y para el diseño, siempre terminan manipulando datos, ya se trate de atributos del modelo de procesos, del modelo conceptual y del modelo de objetos. Independientemente de ellas, siempre se termina de una manera u otra, persistiendo estos datos en algún lugar. Estos lugares pueden ser en general dos: **archivos** o **bases de datos** siendo este último el más indicado cuando se trata de grandes volúmenes información. Existen en el mercado muchas “Bases de Datos”, las cuales pueden valerse de entornos gráficos, facilidades para su utilización y demás características, pero que -sin menospreciar esos tipos de software y aceptando su gran difusión y utilidad- no reúnen varias de las capacidades que debe poseer un Sistema de Manejo de Base de Datos (DBMS). Las aplicaciones desarrolladas con ciertas herramientas de este tipo, pueden bastar para hacer un desarrollo que funcione adecuadamente, pero tienen límites que exigen a quien las utiliza tener que considerar demasiadas cosas directamente por las aplicaciones. De todas maneras se debe reconocer la validez del uso de estas herramientas que si bien por su costo, difusión y facilidad de manejo son accesibles por una gran comunidad de usuarios.

Con respecto a los distintos enfoques que se distinguen para encarar el análisis y diseño de sistemas, tres son los utilizados:

- **Orientados al PROCEDIMIENTO** (Análisis y Diseño clásico)
- **Orientados a los DATOS** (Análisis y Diseño Conceptual)
- **Orientados a OBJETOS**

En la actualidad, si bien todo indicaría que el último de éstos es el más conveniente a la hora de encarar un nuevo proyecto, se debe tener en cuenta también todos los desarrollos existentes los cuales requieren de mantenimiento. Éstos pueden haber sido creados con los otros enfoques, razón por la cual no pueden dejarse de lado.

El software actual, y las técnicas de programación, llevan cada vez más, a utilizar enfoques como el de datos o el de objetos, lo cual no implica que no se deban utilizar herramientas provistas por otra u otras metodologías, ya que la conjugación de todas, o el aporte de distintos conceptos, logran formar ideas distintas y formas de aplicación que sean las adecuadas para modelar un sistema en particular.

## LOS SISTEMAS DE MANEJO DE BASE DE DATOS (DBMS)

Un Sistema de Manejo de Base de Datos (DBMS *DataBase Management System*), consiste en un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a esos datos. El conjunto de datos se conoce comúnmente como BASE DE DATOS (es la agregación de datos). El objetivo primordial de un DBMS, es crear un ambiente en que sea posible guardar y recuperar información de la base de datos en forma conveniente y eficiente. Los sistemas de base de datos, se diseñan para manejar grandes cantidades de información. El manejo de datos, incluye entre otras cosas:

- definición de las estructuras para el **almacenamiento** de la información
- mecanismos para **manejo** de la información.
- debe cuidar la **seguridad** de la información almacenada en la base de datos, tanto contra las *caídas del sistema* como contra los intentos de *acceso no autorizado*.

Debido a la importancia que tiene la información en casi todas las organizaciones, la base de datos es un recurso valioso. Esto condujo al desarrollo de un gran número de conceptos y técnicas para manejar los datos en forma eficiente. En este tema, se presenta una breve introducción a los principios de los sistemas de base de datos.

### Objetivos de los DBMS

En la década del 70, en que surgieron las bases de datos, la gran mayoría de quienes realizaban análisis, diseño y desarrollo de sistemas informáticos, tenían una formación que los llevaba a trabajar con una metodología totalmente orientada a los procesos. Asimismo los lenguajes de programación eran consecuentes con la ideología general. Estos lenguajes orientados al proceso trabajaban con archivos o sistemas de archivos. Entre estos se hallaban el COBOL, RPG, FORTRAN entre otros. Aún hoy se siguen manteniendo sistemas de estas características que inclusive realizan la explotación de los datos de los sistemas de archivos accediéndolos con distintos lenguajes de programación. El ambiente que se acaba de describir, es un sistema de procesamiento de archivos característico, apoyado por un sistema operativo convencional. Los registros permanentes se guardan en diversos ficheros, y se escriben varios programas de aplicaciones para sacar registros y agregarlos a los ficheros apropiados. Estos sistemas, tienen las siguientes ventajas importantes:

#### **Redundancia e inconsistencia de los datos:**

Puesto que los archivos y los programas de aplicaciones fueron creados por distintos programadores en un período largo, es posible que un mismo dato esté repetido en varios sitios (archivos). Ocurre también que a veces se apela a la repetición de los datos para tener un acceso más rápido y acelerar los procesos. Esta redundancia aumenta los costos de almacenamiento y acceso, además de incrementar la posibilidad de que exista inconsistencia en la información, es decir que las distintas copias de la misma información no concuerdan entre sí.

#### **Dificultad para tener acceso a los datos:**

El usuario final, solamente podrá acceder a los datos a través de aplicaciones des-

arrolladas con el fin de acceder a los archivos de una manera determinada para obtener los datos requeridos. No es flexible la consulta que solamente es posible a través de programas.

**Diferentes tipos de datos para la misma información:**

Puesto que los datos están repartidos en varios archivos, y éstos pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicaciones para obtener los datos apropiados. Un mismo dato, además de estar repetido, es común que las diferentes ocurrencias sean de distinto tipo (generalmente de distinta longitud y en ciertos casos, distinto tipo de dato).

**Usuarios múltiples:**

Para mejorar el funcionamiento general del sistema y tener un tiempo de respuesta más corto, muchos sistemas permiten que varios usuarios actualicen la información en forma simultánea. En un ambiente de este tipo, la interacción de las actualizaciones concurrentes puede resultar en información inconsistente. Puesto que muchos programas de aplicaciones diferentes sin coordinación previa pueden tener acceso a los datos, es muy difícil de conseguir un supervisor de este tipo.

**Problemas de seguridad:**

No es recomendable que todos los usuarios del sistema puedan tener acceso a toda la información, por lo que el nivel de permisos debería ser administrado por los mismos archivos contenedores de la información y no que ésta sea administrada solamente por las aplicaciones que los acceden.

**Problemas de integridad:**

Los valores de los datos que se guardan en la base de datos, deben satisfacer ciertos tipos de limitantes de consistencia independientemente de los programas de aplicación.

Estos problemas, entre otros, han fomentado el desarrollo de los DBMS.

## **ABSTRACCIÓN DE LA INFORMACIÓN**

Uno de los objetivos principales de un sistema de base de datos, es proporcionar a los usuarios una visión abstracta de la información, es decir, el sistema oculta ciertos detalles relativos a la forma de cómo los datos se almacenan y mantienen. Sin embargo, para que el sistema sea útil, la información debe recuperarse en forma eficiente. La búsqueda de la eficiencia, conduce al diseño de estructuras de datos complejas para representar la información en la base de datos. Pero como los sistemas muchas veces son utilizados por personal que no cuenta con conocimientos de computación, esta complejidad debe estar escondida para los usuarios. Para ocultarla, se definen varios niveles de abstracción en los que puede observarse la base de datos.

Estos niveles son:

- **Nivel Físico:**

Este es el nivel más bajo de abstracción, en el que se describe cómo se al-

macenan realmente los datos, es decir, como el sistema operativo los agrupa y maneja. En este nivel, se describen en detalle las estructuras de datos complejas del nivel más bajo.

- **Nivel Conceptual:**

Este es el siguiente nivel más alto de abstracción, en el que se describe cuáles son los datos reales que están almacenados en la base de datos y qué relaciones existen entre ellos. Este nivel contiene toda la base de datos en términos de estructuras relativamente sencillas. Aunque es posible que la implantación de estructuras simples del nivel conceptual requiera estructuras complejas en el nivel físico, no es forzoso que el usuario del nivel conceptual se dé cuenta de ello. El nivel conceptual de abstracción lo utilizan los administradores de bases de datos, y también los analistas a cargo de los desarrollos, quienes deciden qué información se debe guardar en la base de datos.

- **Nivel de Visión:**

Éste es el nivel de abstracción más alta, en el cual se describe solamente una parte de la base de datos. Aunque en el nivel conceptual se utilizan estructuras más simples, todavía queda una forma de complejidad que resulta del gran tamaño de la base de datos. Muchos usuarios de la base de datos no tendrán que ocuparse de toda esta información. Más bien, necesitarán solamente una parte de la base de datos. Para simplificar la interacción entre estos usuarios y el sistema, se define el nivel de abstracción de visión. El sistema puede proporcionar muchas vistas diferentes de la misma base de datos.

La capacidad de modificar una definición de estructura en un nivel, sin afectar la definición de la estructura en el nivel inmediato superior, se denomina independencia de los datos. Existen dos niveles de independencia:

### **Independencia Física**

Es la capacidad de modificar la estructura física sin obligar a que se vuelvan a escribir los programas de aplicaciones. En algunas ocasiones son necesarias modificaciones en el nivel físico para mejorar la seguridad, el rendimiento y/o la *performance* general del sistema.

### **Independencia Lógica**

Es la capacidad de modificar la estructura conceptual sin obligar a que se vuelvan a escribir los programas de aplicaciones. Las modificaciones en el nivel conceptual son necesarias siempre que se altera la estructura lógica de la base de datos generalmente debido a cambios en las reglas de gestión o especificaciones funcionales iniciales.

La independencia lógica de los datos, es mucho más difícil de lograr que la independencia física, ya que los programas de aplicaciones y las consultas registradas, dependen en alto grado de la estructura lógica de los datos a los que tienen acceso. Un ejemplo de ello, podría ser el cambio en los tipos de valores predeterminados que puede tener un elemento de dato. En este caso las

modificaciones a realizar sobre los programas de aplicación pueden ser nulas. En cambio, si la modificación consiste en el agregado de un nuevo elemento de dato para una instancia, será necesaria la modificación de los programas de aplicación que los utilicen.

Lo que se pretende es que los sistemas desarrollados cumplan en la mayor medida posible con los niveles de independencia ya que de esta manera su mantenimiento resulta mucho más sencillo.

## **MODELOS DE DATOS**

Un modelo de datos es una serie de conceptos que pueden utilizarse para describir un conjunto de datos y operaciones para manipularlos. Cuando un modelo de datos describe un conjunto de conceptos de una realidad determinada, se denomina *modelo conceptual de datos*. Los conceptos de un modelo de datos se construyen por lo general utilizando sus mecanismos de abstracción y se describen mediante representaciones gráficas y se apoyan en herramientas textuales adicionales.

Existen dos tipos de modelos de datos: modelos conceptuales, usados en el diseño de bases de datos, y modelos lógicos, apoyados por los sistemas de gestión de bases de datos (DBMS). Los modelos conceptuales son instrumentos para representar la realidad a un nivel alto de abstracción. Utilizando los modelos conceptuales, se puede construir una descripción de la realidad, fácil de entender e interpretar obteniéndose de esa manera los denominados modelos lógicos o físicos que se apoyan en descripciones de datos procesables por una computadora. Estos modelos tienen una correspondencia directa con la estructura física de la base de datos.

En el diseño de bases de datos se usan primero los modelos conceptuales para lograr una descripción de alto nivel de la realidad; después se transforma el esquema conceptual en un esquema lógico o físico.

## **ESTRUCTURA DE LA BASE DE DATOS**

### **Instancias y estructura**

Las bases de datos, cambian con el tiempo, al insertarse y eliminarse información de ella. El conjunto de información almacenado en la base de datos en cierto momento se denomina una instancia en la base de datos. El diseño general de la base de datos, se llama estructura o esquema de la base de datos. Los esquemas se alteran muy raras veces. Existen maneras mediante las cuales se modifican, agregan, borran o consultan instancias de la base de datos y otras formas que permiten trabajar con la estructura de ella. Estas maneras se manifiestan a través de lenguajes distintos creados con objetivos específicos.

### **El lenguaje de definición de datos (DDL)**

Una estructura de base de datos, se especifica por medio de una serie de definiciones que se expresan en un lenguaje especial llamado lenguaje de definición de datos (DDL: *Data Definition Language*). El resultado de la compilación y ejecución de las proposiciones DDL es un con-

junto de información (con su propia estructura) que se almacena en un lugar especial denominado diccionario, catálogo o directorio de datos.

Un diccionario de datos, es una estructura que contiene metadatos, es decir “datos acerca de los datos”. Esta estructura se consulta antes de leer o modificar los datos reales en el sistema de base de datos.

La estructura de almacenamiento y los métodos de acceso empleados por el sistema de base de datos, se especifican por medio de un conjunto de definiciones de un tipo especial de DDL llamado lenguaje de almacenamiento y definición de datos. El resultado de la compilación de estas definiciones es una serie de instrucciones que especifican los detalles de implantación de las estructuras de base de datos que normalmente no pueden ver los usuarios.

### **El lenguaje de manejo de datos (DML)**

Los datos que estarán contenidos en esa estructura definida mediante el *DDL* necesariamente deben ser mantenidos. Este mantenimiento se traduce en las siguientes operaciones básicas:

- **la recuperación o consulta** de información almacenada en la base de datos
- **la inserción** de información nueva en la base de datos
- **la modificación o actualización** de información existente en la base de datos
- **la eliminación** de información de la base de datos

Un lenguaje de manejo de datos (DML: *Data Manipulation Language*) permite a los usuarios manejar o tener acceso a los datos que estén organizados por medio del modelo apropiado.

Una consulta es una proposición que solicita la recuperación de información. La parte de un DML que implica la recuperación de información, se conoce como lenguaje de consultas. Aunque técnicamente es incorrecto, suelen utilizarse los términos de lenguaje de consultas y lenguaje de manejo de datos como sinónimos.

## **MANEJADOR DE BASE DE DATOS**

Generalmente las bases de datos requieren una gran cantidad de espacio de almacenamiento. La memoria principal de la computadora no puede mantener almacenada esta información, entonces se hace necesario utilizar dispositivos de almacenamiento secundario destinados a tal efecto. Los datos entonces, son transferidos entre el almacenamiento secundario y la memoria principal, según se requiera. Ya que el movimiento de los datos desde y al dispositivo soporte es lento -comparado con la velocidad de los procesadores-, es imperativo que el sistema de base de datos estructure la información de tal manera que se reduzca la necesidad de transferencia entre los discos y la memoria principal.

El objetivo de un sistema de base de datos es simplificar y facilitar el acceso a los datos. Por otra parte, uno de los factores primordiales para la satisfacción o insatisfacción del usuario

con el sistema de base de datos, es su funcionamiento y particularmente - además de las funcionalidades - el tiempo de respuesta a sus requerimientos. Si el tiempo de respuesta para una consulta es demasiado largo, el valor del sistema se reduce. El funcionamiento del sistema depende de la eficiencia de las estructuras de datos utilizadas para representar los datos en la base de datos y de cuán eficiente pueda operar el sistema con esas estructuras. Como sucede en muchos otros aspectos de los sistemas de cómputo, deben hacerse concesiones, no sólo entre el espacio y el tiempo, sino también entre la eficiencia de un tipo de operación y la de otro.

Un manejador de base de datos, es un módulo de programa que constituye la interfaz entre los datos de bajo nivel almacenados físicamente en la base de datos con los programas de aplicaciones y las consultas hechas al sistema. El manejador de base de datos es el responsable de las siguientes tareas:

- a. **Interacción con el manejador de archivos.** Los datos sin procesar se almacenan en el disco mediante el sistema de archivos proporcionado normalmente por un sistema operativo convencional. El manejador de base de datos, traduce las diferentes proposiciones realizadas en DML a comandos de sistema de archivos de bajo nivel. Así, el manejador de base de datos se encarga realmente del almacenamiento, recuperación y actualización de los datos en la base de datos.
- b. **Implantación de la integridad.** Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de limitantes de consistencia. Por ejemplo, el saldo de una cuenta bancaria, no debe bajar de un mínimo previamente especificado, o el tipo de documento de las personas pueden tomar solamente ciertos valores preestablecidos. El administrador y el diseñador de la base de datos, deben especificar estas limitantes en forma explícita. Si se especifican estas limitantes, entonces el manejador de la base de datos puede verificar si las actualizaciones realizadas son válidas o incurren en la violación de cualquiera de estas limitantes, y de ser así, realizará la acción apropiada sin dejar librados los controles a los programas de aplicación y explotación de datos.
- c. **Puesta en práctica de la seguridad.** Como se mencionó anteriormente, no es preciso que los usuarios de la base de datos tengan acceso a todo su contenido, o bien, en caso de tenerlo, no suele ser necesario que puedan manipularlos indiscriminadamente; sino que algunos podrán realizar cierto tipo de operaciones que otros no pueden efectuar. Es labor del manejador de la base de datos realizar el control y exigir el cumplimiento de estos requisitos de seguridad especificados.
- d. **Respaldo y recuperación.** Un sistema de cómputo, como cualquier otro dispositivo mecánico, eléctrico o electrónico, está sujeto a fallas. Existen muy diversas causas de estas fallas, entre ellas la caída de las cabezas lectoras del disco, la interrupción del suministro de energía y los errores propios del software. En cada uno de estos casos se pierde información de la base de datos. Es responsabilidad del manejador de la base de datos sobreponerse a estas fallas y restaurar la base de datos al estado que existía antes de presentarse el desperfecto. Esto se logra normalmente iniciando diversos procedimientos de respaldo y recuperación que el manejador administra automáticamente. Estos mecanismos pueden ser los de espejado, replicación, registro de log de transacciones etc. Otros directamente pueden ser provistos por el hardware.

re trabajando en conjunto con el motor de base.

- e. **Control de concurrencia.** Cuando varios usuarios actualizan la base de datos en forma concurrente, es posible que no se conserve la consistencia de los datos. Es necesario que el sistema controle la interacción entre los usuarios concurrentes; lograr dicho control, es otra de las tareas del manejador de la base de datos.

Algunos sistemas de base de datos, diseñados para utilizarse en computadoras personales, NO CUENTAN con varias de las funciones mencionadas. Esto da como resultado un manejador o motor de base de datos de menor tamaño. Un motor pequeño requiere menos recursos de hardware -sobre todo memoria principal- y su implantación es más económica. Por ejemplo, muchos sistemas pequeños incluyen la restricción de que sólo un usuario puede tener acceso a la base de datos en un momento dado, o bien pueden tener limitaciones en el tamaño de la base, o bien en caso de trabajar sobre un sistema operativo de red, los controles de concurrencia deben ser efectuados por los programas de aplicación desarrollados realizando los bloqueos que crean pertinentes. Otros dejan las tareas de respaldo, recuperación e implantación de la seguridad al usuario. Aunque este enfoque de bajo costo y funciones limitadas es suficiente para bases de datos personales, no es adecuada para cumplir con los requerimientos de una organización de tamaño mediano o grande.

## ADMINISTRADOR DE BASE DE DATOS (DBA)

Una de las razones principales para contar con sistemas de manejo de base de datos, es tener un control centralizado tanto de los datos como de los usuarios y programas que tienen acceso a ellos. La persona que tiene este control centralizado sobre el sistema es el administrador de base de datos (DBA: *DataBase Administrator*). Las funciones del administrador de base de datos son, entre otras:

- a. **Definición de la estructura de almacenamiento y del método de acceso**, es decir, la creación de las estructuras de almacenamiento y acceso apropiados. Esto se lleva a cabo escribiendo una serie de definiciones que posteriormente son traducidas por el compilador de almacenamiento y definición de datos.
- b. **Definición de estructura**, es decir, la creación de la estructura original de la base de datos. Esto se logra escribiendo una serie de definiciones que el compilador de DDL traduce a un conjunto de componentes que se almacenan permanentemente en el diccionario de datos.
- c. **Modificación de la estructura y de la organización física**, ya sea la modificación de la estructura de la base de datos o de la descripción de la organización física del almacenamiento. Estos cambios, aunque son relativamente poco frecuentes, se logran escribiendo una serie de definiciones utilizadas, ya sea por el compilador de DDL o por el compilador de lenguaje de almacenamiento y definición de datos para generar modificaciones a los componentes internos apropiados del sistema.
- d. **Especificación de las limitantes de integridad.** Estas limitantes se conservan en una estructura especial del sistema que consulta el motor cada vez que se lleva a ca-



bo una actualización en los datos.

## USUARIOS DE LA BASE DE DATOS

Uno de los objetivos primordiales de la base de datos es crear un ambiente para la recuperación de información y para almacenar información nueva en la base de datos por parte de usuarios distintos. Existen tres tipos diferentes de usuarios de un sistema de base de datos, y se distinguen por el modo en que ellos esperan interactuar con el sistema.

- a. **Programadores de aplicaciones.** Estos son profesionales en computación que interactúan con el sistema mediante llamadas en DML, las cuales están incrustadas o embebidas en un programa escrito en un lenguaje huésped. Estos constituyen los programas de aplicación que hacen explotación de los datos contenidos en la base.
- b. **Usuarios casuales.** Son usuarios complejos que interactúan con el sistema sin escribir programas. Sus consultas a la base las hacen a través de la escritura de proposiciones en un lenguaje de consulta de base de datos. Cada una de tales consultas se maneja a través de un procesador de consultas, cuya función es tomar una proposición en DML y descomponerla en instrucciones que pueda entender el motor de la base de datos.
- c. **Usuarios operativos.** Son usuarios poco complejos que interactúan con el sistema llamando alguno de los programas de aplicaciones permanentes escritos previamente.

## ESTRUCTURA GENERAL DE LOS DBMS

Un sistema de base de datos, se divide en módulos que se encargan de cada una de las tareas del sistema general. Algunas de las funciones del DBMS pueden ser realizadas por el sistema operativo. En la mayor parte de los casos, el sistema operativo proporciona únicamente los servicios más elementales y la base de datos debe partir de ese fundamento. Así, el diseño de la base de datos debe incluir una consideración de la interfaz entre el sistema de base de datos y el sistema operativo.

Un sistema de base de datos, consiste en varios componentes funcionales entre los que cuentan:

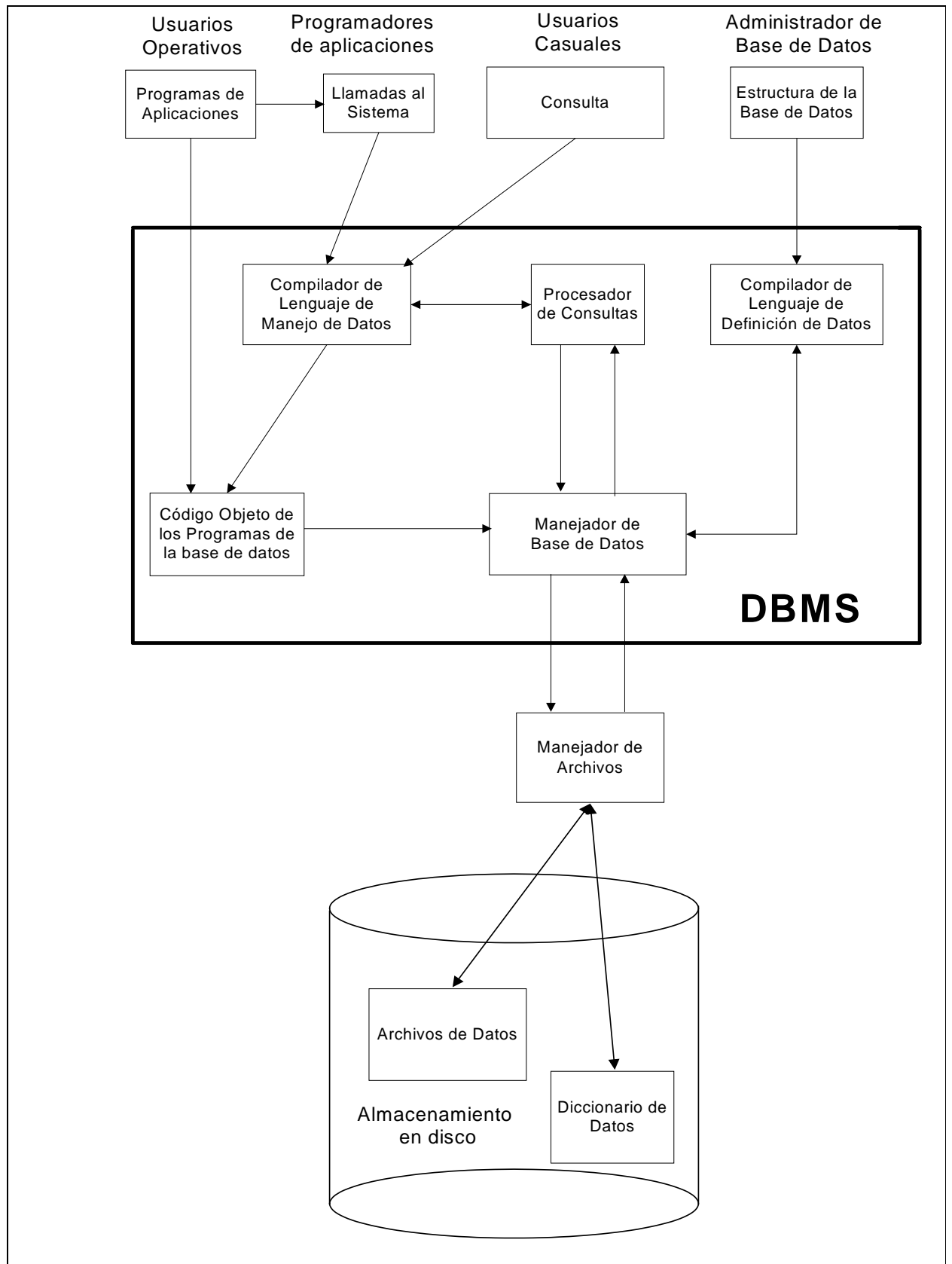
- a. **El manejador de archivos** encargado de asignar espacio en el disco y de las estructuras de datos que se van a emplear para representar la información almacenada en el disco. Es el sistema operativo quien se encarga finalmente de implementar físicamente las estructuras necesarias.
- b. **El manejador de base de datos**, que constituye la interfaz entre los datos de bajo nivel almacenados en la base de datos y accedidos mediante el manejador de archivos y las consultas que se hacen al independientemente de su origen.
- c. **El procesador de consultas**, que traduce las proposiciones realizadas en lenguaje de consulta a instrucciones de bajo nivel que puede entender el manejador de la base de

datos. Además, el procesador de consultas trata de convertir la solicitud del usuario a una forma equivalente pero más eficiente, encontrando una estrategia adecuada para ejecutar la consulta.

- d. **El compilador de DML**, que convierte las proposiciones en DML incrustadas en un programa de aplicaciones, en llamadas normales a procedimientos en el lenguaje huésped. El compilador debe interactuar con el procesador de consultas para generar el código apropiado.
- e. **El compilador de DDL**, que convierte las proposiciones en DDL en un conjunto de tablas que contienen metadatos. Tales tablas se almacenan después en el diccionario de datos.

Además, se requieren varias estructuras de datos como parte de la implantación del sistema físico incluyendo:

- **Archivos de datos**, que guardan la agregación de datos.
- **Diccionario de datos**, que almacena la información relativa a la estructura de la base de datos. Se usa constantemente, por lo que debe tenerse mucho cuidado de desarrollar un diseño apropiado y una implantación eficiente. La estructura del diccionario es impuesta por la empresa que construye el motor de base.
- **Índices**, que permiten el acceso rápido a elementos de información que contienen valores determinados.



## CONCLUSIONES

El nivel de justificación para cambiar de un sistema convencional de archivos, a una base de datos, es equivalente al necesario para pasar de uno manual a uno computarizado. Podría expresarse la evolución como:

**Sist. Manual** → **Sist. Computarizado Convencional** → **Sist. de Base de Datos (DBMS)**

Las principales ventajas de un sistema frente al otro son:

- Independencia entre DATOS y PROGRAMAS
- Facilidad de DISEÑO
- Facilidad de PROGRAMACIÓN
- Utilización de Lenguajes HUÉSPEDES
- Procesos CONCURRENTES
- PROTECCIÓN de la Información
- Facilidad de ACCESO y CONSULTA
- Sistemas EVOLUTIVOS

Las desventajas que podrían enumerarse son:

- Mayor necesidad de MEMORIA PRINCIPAL
- Mayor potencia de RECURSOS de HARDWARE
- Mayor ESPACIO EN DISCO.
- Más cantidad de CANALES de ENTRADA/SALIDA.
- DISMINUCIÓN de VELOCIDAD DE PROCESO.
- INCOMPATIBILIDAD CON OTRAS DBMS.