(1) gestión y diseño de bases de datos

(1.1) datos y archivos

(1.1.1) la necesidad de gestionar datos

En el mundo actual existe una cada vez mayor demanda de datos. Esta demanda siempre ha sido patente en empresas y sociedades, pero en estos años la demanda todavía de ha disparado más debido al acceso multitudinario a Internet.

El propio nombre **Informática** hace referencia al hecho de ser una ciencia que trabaja con información. Desde los albores de la creación de ordenadores, la información se ha considerado como uno de los pilares de las computadoras digitales. Por ello las bases de datos son una de las aplicaciones más antiguas de la informática.

En informática se conoce como **dato** a cualquier elemento informativo que tenga relevancia para el sistema. Desde el inicio de la informática se ha reconocido al dato como al elemento fundamental de trabajo en un ordenador. Por ello se han realizado numerosos estudios y aplicaciones para mejorar la gestión que desde las computadoras se realiza de los datos.

Inicialmente los datos que se necesitaba almacenar y gestionar eran pocos, pero poco a poco han ido creciendo. En la actualidad las numerosas aplicaciones de Internet han producido enormes sistemas de información que incluso para poder gestionarles requieren decenas de máquinas haciendo la información accesible desde cualquier parte del planeta y en un tiempo rápido. Eso ha requerido que la ciencia de las bases de datos esté en continua renovación para hacer frente a esas enormes necesidades.

Pero incluso podemos remontarnos más al hablar de datos. El ser humano desde siempre ha necesitado gestionar datos; de esta forma se controlaban almacenes de alimentos, controles de inventario y otras muchos sistemas de datos. Como herramienta el ser humano al principio sólo poseía su memoria y cálculo y como mucho la ayuda de sus dedos.

La escritura fue la herramienta que permitió al ser humano poder gestionar bases cada vez más grandes de datos. Además de permitir compartir esa información entre diferentes personas, también posibilitó que los datos se guardaran de manera continua e incluso estuvieran disponibles para las siguientes generaciones. Los problemas actuales con la privacidad ya aparecieron con la propia escritura y así el cifrado de

datos es una técnica tan antigua como la propia escritura para conseguir uno de los todavía requisitos fundamentales de la gestión de datos, la **seguridad**.

Para poder almacenar datos y cada vez más datos, el ser humano ideó nuevas herramientas archivos, cajones, carpetas y fichas en las que se almacenaban los datos.

Antes de la aparición del ordenador, el tiempo requerido para manipular estos datos era enorme. Sin embargo el proceso de aprendizaje era relativamente sencillo ya que se usaban elementos que el usuario reconocía perfectamente.

Por esa razón, la informática adaptó sus herramientas para que los elementos que el usuario maneja en el ordenador se parezcan a los que utilizaba manualmente. Así en informática se sigue hablado de ficheros, formularios, carpetas, directorios,....

(1.1.2) sistemas de información

la empresa como sistema

Según la RAE, la definición de sistema es *"Conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a un determinado objeto"* .

La clientela fundamental del profesional de la informática es la empresa. La empresa se puede entender como un sistema formado por diversos objetos: el **capital**, los **recursos humanos**, los **inmuebles**, los **servicios** que presta, etc.

El sistema completo que forma la empresa, por otra parte, se suele dividir en los siguientes subsistemas:

- **Subsistema productivo.** También llamado subsistema real o físico. Representa la parte de la empresa encargada de gestionar la producción de la misma.
- **Subsistema financiero.** Encargado de la gestión de los bienes económicos de la empresa
- **Subsistema directivo.** Encargado de la gestión organizativa de la empresa

Hay que hacer notar que cada subsistema se asocia a un departamento concreto de la empresa.

sistemas de información

Los sistemas que aglutinan los elementos que intervienen para gestionar la información que manejan los subsistemas empresariales es lo que se conoce como Sistemas de Información. Se suele utilizar las siglas **SI** o **IS** (de *Information Server)* para referirse a ello).

Realmente un sistema de información sólo incluye la información que nos interesa de la empresa y los elementos necesarios para gestionar esa información.

Un sistema de información genérico está formado por los siguientes elementos:

- **Recursos físicos.** Carpetas, documentos, equipamiento, discos,...
- **Recursos humanos.** Personal que maneja la información
- **Protocolo.** Normas que debe cumplir la información para que sea manejada (formato de la información, modelo para los documentos,...)

Las empresas necesitan implantar estos sistemas de información debido a la competencia que las obliga a gestionar de la forma más eficiente sus datos para una mayor calidad en la organización de las actividades de los subsistemas empresariales.

componentes de un sistema de información electrónico

En el caso de una **gestión electrónica de la información** (lo que actualmente se considera un **sistema de información electrónico**), los componentes son:

- **Datos.** Se trata de la información relevante que almacena y gestiona el sistema de información. Ejemplos de datos son: *Sánchez*, *12764569F*, *Calle Mayo 5*, *Azul*...
- **Hardware.** Equipamiento físico que se utiliza para gestionar los datos. cada uno de los dispositivos electrónicos que permiten el funcionamiento del sistema de información.
- **Software.** Aplicaciones informáticas que se encargan de la gestión de la base de datos y de las herramientas que facilitan su uso.
- **Recursos humanos.** Personal que maneja el sistema de información.

(1.1.3) archivos

Los ficheros o archivos son la herramienta fundamental de trabajo en una computadora todavía a día de hoy. Las computadoras siguen almacenando la información en ficheros, eso sí de estructura cada vez más compleja.

Los datos deben de ser almacenados en componentes de almacenamiento permanente, lo que se conoce como **memoria secundaria** (discos duros u otras unidades de disco). En esas memorias, los datos se estructuran en archivos (también llamados ficheros).

Un fichero es una secuencia de números binarios que organiza información relacionada a un mismo aspecto.

En general sobre los archivos se pueden realizar las siguientes operaciones:

- **Abrir** (*open*). Prepara el fichero para su proceso.
- **Cerrar** *(close*). Cierra el fichero impidiendo su proceso inmediato.
- **Leer** *(read*). Obtiene información del fichero.
- **Escribir** *(write*). Graba información en el fichero.
- **Posicionarse** *(seek)*. Coloca el puntero de lectura en una posición concreta del mismo (no se puede realizar en todos los tipos de ficheros).
- **Fin de fichero** *(eof)*. Indica si hemos llegado al final del fichero.

Cuando los ficheros almacenan datos, se dice que constan de **registros**. Cada registro contiene datos relativos a un mismo elemento u objeto. Por ejemplo en un fichero de personas, cada registro contiene datos de una persona. Si el archivo contiene datos de 1000 personas, constará de 1000 registros.

A continuación se explican los tipos más habituales de ficheros.

ficheros secuenciales

En estos ficheros, los datos se organizan secuencialmente en el orden en el que fueron grabados. Para leer los últimos datos hay que leer los anteriores. Es decir leer el registro número nueve, implica leer previamente los ocho anteriores.

ventajas

- Rápidos para obtener registros contiguos de una base de datos
- No hay huecos en el archivo al grabarse los datos seguidos, datos más compactos.

desventajas

- Consultas muy lentas al tener que leer todos los datos anteriores al dato que queremos leer
- Algoritmos de lectura y escritura más complejos
- No se pueden eliminar registros del fichero (se pueden marcar de manera especial para que no sean tenidos en cuenta, pero no se pueden borrar)
- El borrado provoca archivos que no son compactos
- La ordenación de los datos requiere volver a crearle de nuevo

ficheros de acceso directo o aleatorio

Se puede leer una posición concreta del fichero, con saber la posición (normalmente en bytes) del dato a leer. Cuando se almacenan registros, posicionarnos en el quinto registro se haría de golpe, lo único necesitamos saber el tamaño del registro, que en este tipo de ficheros debe de ser el mismo. Suponiendo que cada registro ocupa 100 bytes, el quinto registro comienza en la posición 400. Lo que se hace es colocar el llamado **puntero de archivo** en esa posición y después leer.

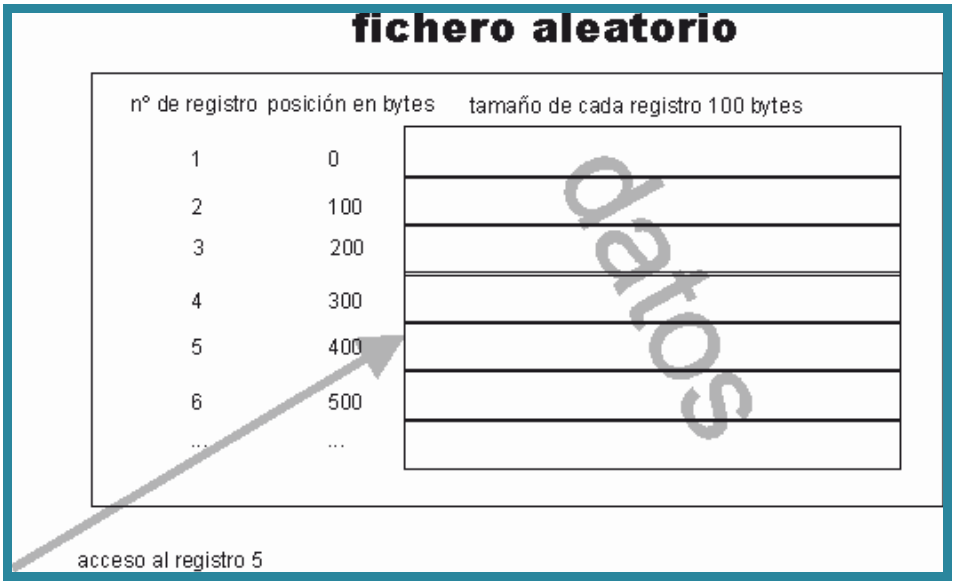

Ilustración 1, Ejemplo de fichero de acceso directo

ventajas

- Acceso rápido al no tener que leer los datos anteriores
- La modificación de datos es más sencilla
- Permiten acceso secuencial
- Permiten leer y escribir a la vez
- Aptos para organizaciones **relativas directas**, en las que la clave del registro se relaciona con su posición en el archivo

desventajas

- Salvo en archivos relativos directos, no es apto por sí mismo para usar en bases de datos, ya que los datos se organizan en base a una clave
- No se pueden borrar datos (sí marcar para borrado, pero generarán huecos)
- Las consultas sobre multitud de registros son más lentas que en el caso anterior.

ficheros secuenciales encadenados

Son ficheros secuenciales gestionados mediante punteros, datos especiales que contienen la dirección de cada registro del fichero. Cada registro posee ese puntero que indica la dirección del siguiente registro y que se puede modificar en cualquier momento. El puntero permite recorrer los datos en un orden concreto.

Cuando aparece un nuevo registro, se añade al final del archivo, pero los punteros se reordenan para que se mantenga el orden.

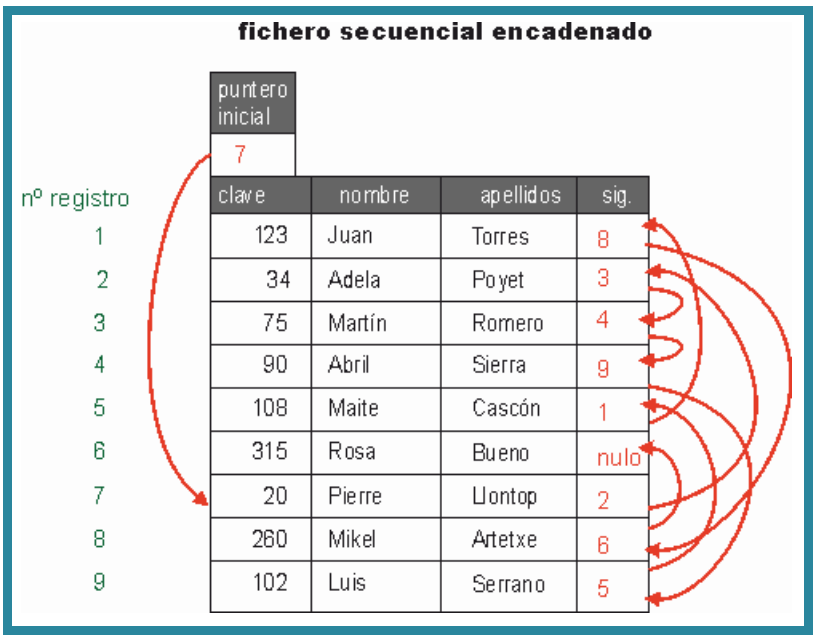

Ilustración 2, ejemplo de fichero secuencial encadenado. Los punteros le recorren por la clave

ventajas

- El fichero mantiene el orden en el que se añadieron los registros y un segundo orden en base a una clave
- La ordenación no requiere reorganizar todo el fichero, sino sólo modificar los punteros
- Las mismas ventajas que el acceso secuencial
- En esta caso sí se borran los registros y al reorganizar, se perderán definitivamente

desventajas

- No se borran los registros, sino que se marcan para ser ignorados. Por lo que se malgasta espacio

- Añadir registros o modificar las claves son operaciones que requieren recalcular los punteros

ficheros secuenciales indexados

Se utilizan dos ficheros para los datos, uno posee los registros almacenados de forma secuencial, pero que permite su acceso aleatorio. El otro posee una tabla con punteros a la posición ordenada de los registros. Ese segundo fichero es el **índice,** una tabla con la ordenación deseada para los registros y la posición que ocupan en el archivo.

El archivo de índices posee unas cuantas entradas sólo en las que se indica la posición de ciertos valores claves en el archivo (cada 10, 15 ,20,... registros del archivo principal se añade una entrada en el de índices). El archivo principal tiene que estar siempre ordenado y así cuando se busca un registro, se busca su valor clave en la tabla de índices, la cual poseerá la posición del registro buscado. Desde esa posición se busca secuencialmente el registro hasta encontrarlo.

Existe un archivo llamado de **desbordamiento** u **overflow** en el que se colocan los nuevos registros que se van añadiendo (para no tener que ordenar el archivo principal cada vez que se añade un nuevo registro) este archivo está desordenado. Se utiliza sólo si se busca un registro y no se encuentra en el archivo principal. En ese caso se recorre todo el archivo de overflow hasta encontrarlo.

Para no tener demasiados archivos en overflow (lo que restaría velocidad), cada cierto tiempo se reorganiza el archivo principal. Ejemplo:

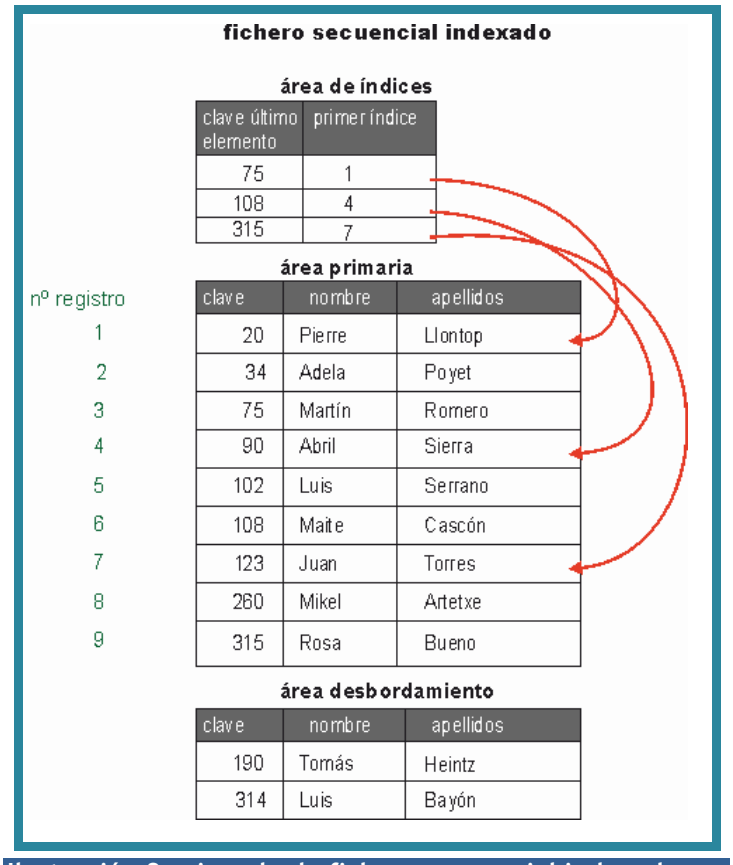

Ilustración 3, ejemplo de fichero secuencial indexado

ventajas

- El archivo está siempre ordenado en base a una clave
- La búsqueda de datos es rapidísima
- Permite la lectura secuencial (que además será en el orden de la clave)
- El borrado de registros es posible (aunque más problemático que en el caso anterior)

desventajas

- Para un uso óptimo hay que reorganizar el archivo principal y esta operación es muy costosa ya que hay que reescribir de nuevo y de forma ordenada todo el archivo.
- La adición de registros requiere más tiempo que en los casos anteriores al tener que reordenar los índices

ficheros indexado-encadenados

Utiliza punteros e índices, es una variante encadenada del caso anterior. Hay un fichero de índices equivalente al comentado en el caso anterior y otro fichero de tipo encadenado con punteros a los siguientes registros. Cuando se añaden registros se añaden en un tercer registro llamado de desbordamiento u ***overflow***. En ese archivo los datos se almacenan secuencialmente, se accede a ellos si se busca un dato y no se encuentra en la tabla de índices.

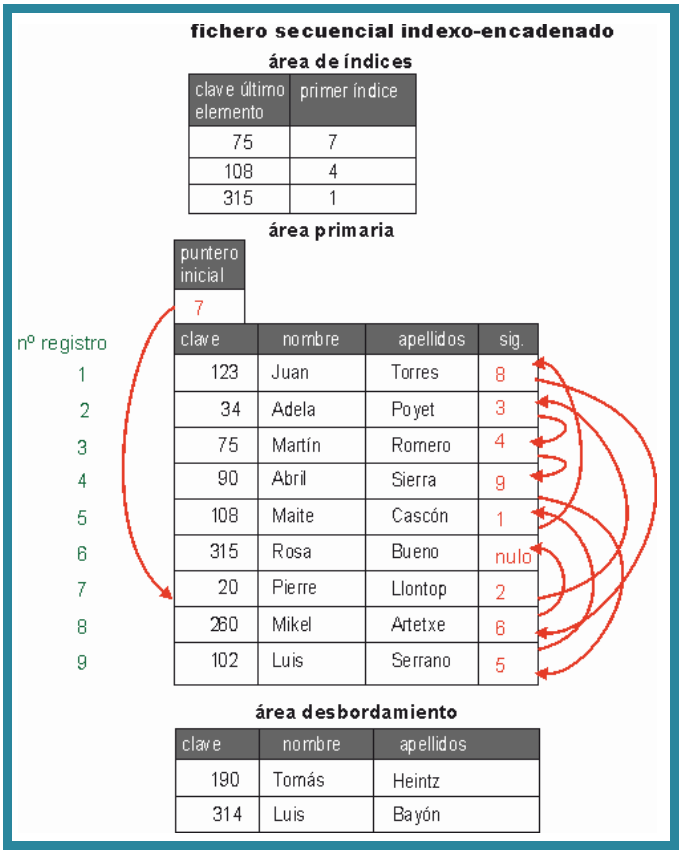

Ilustración 4, Ejemplo de archivo secuencial indexado y encadenado

ventajas

- Posee las mismas ventajas que los archivos secuenciales indexados, además de una mayor rapidez al reorganizar el fichero (sólo se modifican los punteros)

desventajas

- Requieren compactar los datos a menudo para reorganizar índices y quitar el fichero de desbordamiento.

(1.1.4) operaciones relacionadas con uso de ficheros en bases de datos

borrado y recuperación de registros

Algunos de los tipos de ficheros vistos anteriormente no admiten el borrado real de datos, sino que sólo permiten añadir un dato que indica si el registro está borrado o no. Esto es interesante ya que permite anular una operación de borrado. Por ello esta técnica de marcar registros, se utiliza casi siempre en todos los tipos de archivos.

En otros casos los datos antes de ser eliminados del todo pasan a un fichero especial (conocido como **papelera**) en el que se mantienen durante cierto tiempo para su posible recuperación.

fragmentación y compactación de datos

La fragmentación en un archivo hace referencia a la posibilidad de que éste tenga huecos interiores debido a borrado de datos u a otras causas. Causa los siguientes problemas:

- Mayor espacio de almacenamiento
- Lentitud en las operaciones de lectura y escritura del fichero

Por ello se requiere **compactar** los datos. Esta técnica permite eliminar los huecos interiores a un archivo. Las formas de realizarla son:

- **Reescribir el archivo para eliminar los huecos.** Es la mejor, pero lógicamente es la más lenta al requerir releer y reorganizar todo el contenido del fichero.
- **Aprovechar huecos.** De forma que los nuevos registros se inserten en esos huecos. Esta técnica suele requerir un paso previo para reorganizar esos huecos.

compresión de datos

En muchos casos para ahorrar espacio de almacenamiento, se utilizan técnicas de compresión de datos. La ventaja es que los datos ocupan menos espacio y la desventaja es que al manipular los datos hay que descomprimirlos lo que hace que la manipulación de los datos sea lenta.

cifrado de datos

Otra de las opciones habituales sobre ficheros de datos es utilizar técnicas de cifrado para proteger los ficheros en caso de que alguien no autorizado se haga con el fichero. Para leer un fichero de datos, haría falta descifrar el fichero. Para descifrar necesitamos una clave o bien aplicar métodos de descifrado; lógicamente cuanto mejor sea la técnica de cifrado, más difícil será descifrar los datos mediante la fuerza bruta.

(1.1.5) tipos de sistemas de información

En la evolución de los sistemas de información ha habido dos puntos determinantes, que han formado los dos tipos fundamentales de sistemas de información.

sistemas de información orientados al proceso

En estos sistemas de información se crean diversas aplicaciones (software) para gestionar diferentes aspectos del sistema. Cada aplicación realiza unas determinadas operaciones. Los datos de dichas aplicaciones se almacenan en archivos digitales dentro de las unidades de almacenamiento del ordenador (a veces en archivos binarios, o en hojas de cálculo, o incluso en archivos de texto).

Cada programa almacena y utiliza sus propios datos de forma un tanto caótica. La ventaja de este sistema (la única ventaja), es que los procesos son independientes por lo que la modificación de uno no afectaba al resto. Pero tiene grandes inconvenientes:

- **Datos redundantes.** Ya que se repiten continuamente
- **Datos inconsistentes.** Ya que un proceso cambia sus datos y no el resto. Por lo que el mismo dato puede tener valores distintos según qué aplicación acceda a él.
- **Coste de almacenamiento elevado.** Al almacenarse varias veces el mismo dato, se requiere más espacio en los discos. Luego se agotarán antes.
- **Difícil acceso a los datos.** Cada vez que se requiera una consulta no prevista inicialmente, hay que modificar el código de las aplicaciones o incluso crear una nueva aplicación.
- **Dependencia de los datos a nivel físico.** Para poder saber cómo se almacenan los datos, es decir qué estructura se utiliza de los mismos, necesitamos ver el código de la aplicación; es decir el código y los datos no son independientes.
- **Tiempos de procesamiento elevados**. Al no poder optimizar el espacio de almacenamiento.
- **Dificultad para el acceso simultáneo a los datos.** Es casi imposible de conseguir ya que se utilizan archivos que no admiten esta posibilidad. Dos usuarios no pueden acceder a los datos de forma concurrente.
- **Dificultad para administrar la seguridad del sistema.** Ya que cada aplicación se crea independientemente; es por tanto muy difícil establecer criterios de seguridad uniformes.

Ilustración 5, Sistemas de Información orientados al proceso

A estos sistemas se les llama sistemas de gestión de ficheros. Se consideran también así a los sistemas que utilizan programas ofimáticos (como **Word** o **Excel** por ejemplo) para gestionar sus datos (muchas pequeñas empresas utilizan esta forma de administrar sus datos). De hecho estos sistemas producen los mismos (si no más) problemas.

sistemas de información orientados a los datos. bases de datos

En este tipo de sistemas los datos se centralizan en una **base de datos** común a todas las aplicaciones. Estos serán los sistemas que estudiaremos en este curso.

En esos sistemas los datos se almacenan en una única estructura lógica que es utilizable por las aplicaciones. A través de esa estructura se accede a los datos que son comunes a todas las aplicaciones.

Cuando una aplicación modifica un dato, dicho dato la modificación será visible para el resto de aplicaciones.

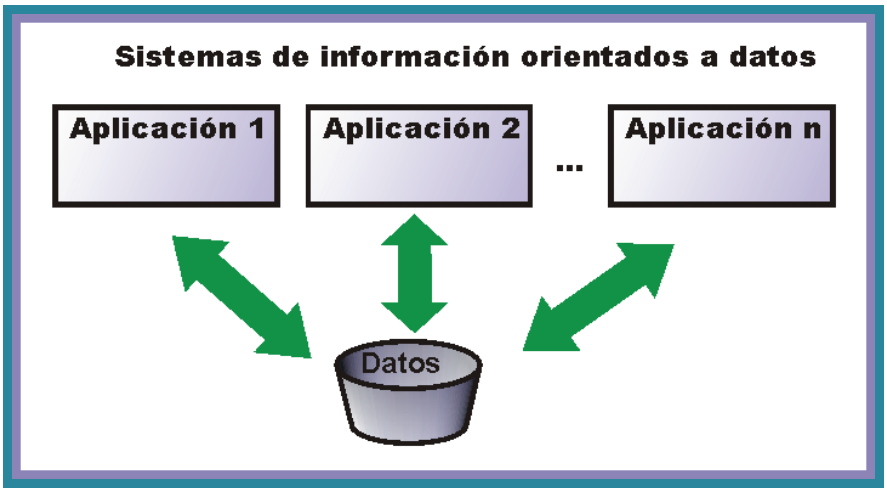

Ilustración 6, Sistemas de información orientados a datos

ventajas

- **Independencia de los datos y los programas y procesos.** Esto permite modificar los datos sin modificar el código de las aplicaciones.
- **Menor redundancia**. No hace falta tanta repetición de datos. Sólo se indica la forma en la que se relacionan los datos.
- **Integridad de los datos**. Mayor dificultad de perder los datos o de realizar incoherencias con ellos.
- **Mayor seguridad en los datos.** Al permitir limitar el acceso a los usuarios. Cada tipo de usuario podrá acceder a unas cosas..
- **Datos más documentados.** Gracias a los **metadatos** que permiten describir la información de la base de datos.
- **Acceso a los datos más eficiente.** La organización de los datos produce un resultado más óptimo en rendimiento.
- **Menor espacio de almacenamiento.** Gracias a una mejor estructuración de los datos.
- **Acceso simultáneo a los datos.** Es más fácil controlar el acceso de usuarios de forma concurrente.

desventajas

- **Instalación costosa.** El control y administración de bases de datos requiere de un software y hardware poderoso
- **Requiere personal cualificado.** Debido a la dificultad de manejo de este tipo de sistemas.
- **Implantación larga y difícil.** Debido a los puntos anteriores. La adaptación del personal es mucho más complicada y lleva bastante tiempo.
- **Ausencia de estándares reales.** Lo cual significa una excesiva dependencia hacia los sistemas comerciales del mercado. Aunque, hoy en día, una buena parte de esta tecnología está aceptada como estándar de hecho.

(1.1.6) utilidad de los sistemas gestores de bases de datos

Un sistema gestor de bases de datos o **SGBD** (aunque se suele utilizar más a menudo las siglas **DBMS** procedentes del inglés, ***Data Base Management System***) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.

En estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos.

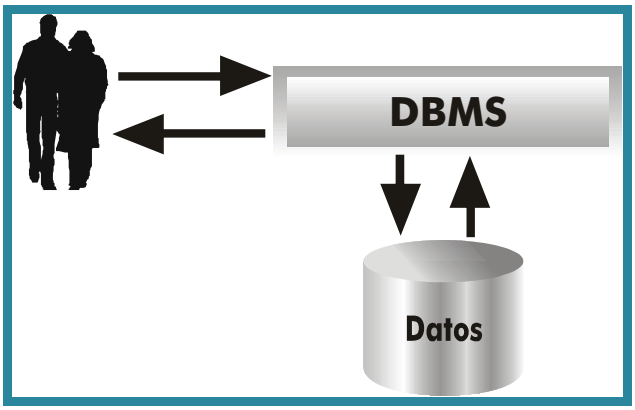

Ilustración 7, Esquema del funcionamiento y utilidad de un sistema gestor de bases de datos

El éxito del SGBD reside en mantener la seguridad e integridad de los datos. Lógicamente tiene que proporcionar herramientas a los distintos usuarios. Entre las herramientas que proporciona están:

- **Herramientas para la creación y especificación de los datos**. Así como la estructura de la base de datos.
- **Herramientas para administrar y crear la estructura física** requerida en las unidades de almacenamiento.
- **Herramientas para la manipulación de los datos** de las bases de datos, para añadir, modificar, suprimir o consultar datos.
- **Herramientas de recuperación** en caso de desastre
- **Herramientas para la creación de copias de seguridad**
- **Herramientas para la gestión de la comunicación** de la base de datos

- **Herramientas para la creación de aplicaciones** que utilicen esquemas externos de los datos
- **Herramientas de instalación** de la base de datos
- **Herramientas para la exportación e importación** de datos

(1.1.7) niveles de abstracción de una base de datos

introducción

En cualquier sistema de información se considera que se pueden observar los datos desde dos puntos de vista:

- **Nivel externo**. Esta es la visión de los datos que poseen los usuarios del Sistema de Información.
- **Nivel físico**. Esta es la forma en la que realmente están almacenados los datos.

Realmente la base de datos es la misma, pero se la puede observar desde estos dos puntos de vista. Al igual que una casa se la pueda observar pensando en los materiales concretos con los que se construye o bien pensando en ella con el plano en papel.

En todo sistema de información digital, los usuarios ven los datos desde las aplicaciones creadas por los programadores. A ese nivel se manejan formularios, informes en pantalla o en papel,...

Pero la realidad física de esos datos, tal cual se almacenan en los discos queda oculta a los usuarios. Esa forma de ver la base de datos está reservada a los administradores. Es el nivel físico el que permite ver la base de datos en función de cómo realmente se están almacenando en el ordenador, en qué carpeta, qué archivos se usan,…

En el caso de los Sistemas de Base de datos, se añade un tercer nivel, un tercer punto de vista, es el **nivel conceptual**. Ese nivel se sitúa entre el físico y el externo.

En cada nivel se manejan esquemas de la base de datos, al igual que al construir una casa, los distintos profesionales manejan distintos tipos de planos (eléctricos, de albañilería, de tuberías de agua,…). Con lo cual una base de datos requiere diseñar al menos tres esquemas (en realidad son más).

esquema físico

Representa la forma en la que están almacenados los datos. Esta visión sólo la requiere el **administrador/a**. El administrador la necesita para poder gestionar más eficientemente la base de datos.

En este esquema se habla de archivos, directorios o carpetas, unidades de disco, servidores,…

esquema conceptual

Se trata de un esquema teórico de los datos en el que figuran organizados en estructuras reconocibles del mundo real y en el que también aparece la forma de relacionarse los datos. Este esquema es el paso que permite modelar un problema real a su forma correspondiente en el ordenador.

Este esquema es la base de datos de todos los demás. Como se verá más adelante, es el primer paso a realizar al crear una base de datos. En definitiva es el plano o modelo general de la base de datos.

El esquema conceptual lo realiza **diseñadores/as** o **analistas**.

esquema externo

En realidad son varios. Se trata de la visión de los datos que poseen los **usuarios y usuarias finales.** Esa visión es la que obtienen a través de las aplicaciones. Las aplicaciones creadas por los desarrolladores abstraen la realidad conceptual de modo que el usuario no conoce las relaciones entre los datos, como tampoco conoce dónde realmente se están almacenando los datos.

Los esquemas externos los realizan las **programadoras/es** según las indicaciones formales de los y las **analistas**.

Realmente cada aplicación produce un esquema externo diferente (aunque algunos pueden coincidir) o **vista de usuario**. El conjunto de todas las vistas de usuario es lo que se denomina **esquema externo global**.

(1.2) componentes de los SGBD

(1.2.1) funciones. lenguajes de los SGBD

Los SGBD tienen que realizar tres tipos de funciones para ser considerados válidos.

función de descripción o definición

Permite al diseñador de la base de datos crear las estructuras apropiadas para integrar adecuadamente los datos. Se dice que esta función es la que permite definir las tres estructuras de la base de datos (relacionadas con los tres niveles de abstracción).

- **Estructura interna**
- **Estructura conceptual**
- **Estructura externa**

Realmente esta función trabajo con **metadatos**. Los metadatos es la información de la base de datos que realmente sirve para describir a los datos. Es decir, ***Sánchez Rodríguez*** y ***Crespo*** son datos; pero ***Primer Apellido*** es un metadato. También son datos decir que la base de datos contiene ***Alumnos*** o que el ***dni*** lo forman 9 caracteres de los cuales los 8 primeros son números y el noveno un carácter en mayúsculas.

La función de definición sirve pues para **crear, eliminar o modificar metadatos**. Para ello permite usar un **lenguaje de descripción de datos** o **DDL**. Mediante ese lenguaje:

- Se definen las estructuras de datos
- Se definen las relaciones entre los datos
- Se definen las reglas que han de cumplir **los** datos

función de manipulación

Permite modificar y utilizar los **datos** de la base de datos. Se realiza mediante un **lenguaje de modificación de datos** o **DML.** Mediante ese lenguaje se puede:

- Añadir datos
- Eliminar datos
- Modificar datos
- Buscar datos