

Introducción

Las bases de datos forman parte de cualquier sistema de información. Junto a estas, se encuentran todas las aplicaciones que permiten su gestión y explotación, además de aquellas que sirven a la mecanización de los procesos de la organización en cuestión. Estas aplicaciones requieren de accesos eficientes al lugar donde se encuentra el gigantesco repositorio de datos. Las tecnologías de mantenimiento usadas son diversas, la arquitectura de almacenamiento varía desde las locales, centralizadas y distribuidas hasta las alojadas en la nube.

Los sistemas gestores de bases de datos son las herramientas que permiten la gestión completa de las bases de datos y, últimamente, su explotación en mecanismos de análisis de datos y de toma de decisiones. El proceso de diseño de una base de datos puede dividirse en cuatro pasos fundamentales: el análisis de requisitos, el diseño conceptual, el diseño lógico y el diseño físico.

El **análisis de requisitos** pretende entender qué datos van a almacenarse y qué aplicaciones deben construirse sobre ella ofreciendo un rendimiento adecuado. Se incluye discusiones con los grupos de usuarios, estudio de los actuales sistemas operacionales y cómo se espera que evolucionen, y examen de cualquier información disponible sobre las aplicaciones existentes que van a ser sustituidas o completadas por la aplicación de bases de datos. Existen multitud de metodologías para realizar esta fase, incluso hay herramientas que lo automatizan.

En el **diseño conceptual** se desarrolla un esquema conceptual atendiendo a la información recogida en el análisis de requisitos.

En el **diseño lógico** se elige el sistema gestor que se va a utilizar y se convierte en un esquema atendiendo a dicho gestor elegido y partiendo del diseño conceptual elaborado en la etapa anterior.

Por último, en el **diseño físico** se diseñan las cargas de trabajo para refinar el diseño con el objetivo de que cumpla los requisitos de rendimiento. Esta fase puede afectar al rediseño de los esquemas obtenidos en fases anteriores.

El diseño físico parte del esquema lógico y del uso esperado de la base de datos, con el objetivo de decidir su configuración física, adaptándose a los dispositivos de almacenamiento masivos y la arquitectura hardware donde se aloja el software para la gestión de la base de datos.

Los mecanismos de explotación de las bases de datos están evolucionando al mismo modo que lo hacen las tecnologías, que posibilitan procesamientos más rápidos y ejecución de algoritmos y metodologías que requieren de sistemas con mayor capacidad de cálculo. Estas técnicas se pueden apoyar en el uso de inteligencia artificial, a través de procesos de la inteligencia de negocios y con sistemas basados en almacenes de datos y Big Data. En esta unidad se hace una presentación de los sistemas transaccionales, los almacenes de datos, la minería de datos y el enfoque de los sistemas Big Data.

1.1. Características de las bases de datos

Las **bases de datos** suponen una colección gigantesca de datos, organizados o no, registrados sistemáticamente en soportes orientados al almacenamiento masivo y con el fin de su gestión posterior. La gestión de datos hace referencia a la creación, manipulación y eliminación de dichos datos. Estas operaciones se denominan también **altas, modificaciones y bajas de datos**.

Se han definido las bases de datos como una ingesta de datos organizados o no. Efectivamente, estos datos pueden encontrarse de forma claramente estructurada, a través del uso de elementos que permiten su organización, como pueden ser las tablas. Pero también pueden encontrarse sin organización. En este tipo de bases de datos, el pilar importante no es la estructuración de los datos, sino el acceso, manejo y administración de estos y los algoritmos que lo permiten. Este tipo de bases de datos, en las que es imposible estructurar la información por su tamaño tan excesivamente grande, se conoce en la actualidad como **soluciones Big Data**.

Cuando se habla de bases de datos, se hace referencia a la información almacenada, registrada, organizada. Parece que nadie usa las expresiones **bases de información** o **sistemas gestores de bases de información**. La diferencia entre datos e información parece relevante. La información la procesan el emisor y el receptor a través de unos elementos llamados **mensajes**, que se transfieren entre ellos. Estos agentes son capaces de procesar dicha información con el fin de tomar decisiones. La información no es procesable por máquinas si no se contextualiza y se representa a través de un conjunto ordenado de símbolos y siguiendo unas reglas léxicas y sintácticas, y que se les impregna de significado a través de otras reglas llamadas **reglas semánticas**. Además, estos mensajes se registran en soportes analógicos o digitales. Finalmente, al resultado de cada uno de esos ítems, clasificados, cuantificados, con significados y que se han almacenado en soportes se denominan **datos**.

Desde el punto de vista del repositorio de datos, una **base de datos** es una colección de archivos relacionados entre sí con la finalidad de permitir el manejo de la información. Estos archivos están compuestos por un conjunto ordenado de registros, siendo el registro la unidad mínima de acceso de los datos que almacena un archivo. Los registros, a su vez, están compuestos por un conjunto de campos. Evidentemente, como se ha indicado con anterioridad, no se puede acceder a un campo de un registro, sino que se accede a todo el registro completo para cuando se quiere obtener el valor de un campo. Cada campo permite almacenar información de algún atributo de alguna entidad del mundo real.

Sin embargo, los **sistemas gestores de bases de datos (SGBD)** son herramientas que permiten organizar y manipular volúmenes de datos de gran tamaño. Estas herramientas son las que acceden directamente a los archivos y sus registros para acceder a los datos, aportando una interfaz de abstracción hacia los usuarios que gestionan los datos. Las facilidades que aportan los SGBD a sus usuarios son, principalmente, las siguientes:

- Actualización de datos.
- Recuperación de datos.

- Agregar nuevos registros a los archivos que ya existen.
- Agregar nuevos archivos para poder almacenar nuevos datos.
- Eliminar los registros o archivos.
- Proporcionar medios para controlar el acceso a los datos.

1.1.1. El modelo de datos óptimo

El modelo de datos es el conjunto de herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitaciones, de tal forma que facilita la interpretación del mundo real y su representación en forma de datos en el sistema informático.

El modelo de datos está compuesto por propiedades estáticas y dinámicas. Las estáticas no varían en el tiempo y se especifican a través del esquema. Para ello, se usa el lenguaje de definición de datos. Definido el modelo de datos, pasamos a analizarlo. Este esquema define la estructura a través de los objetos del modelo y de las restricciones que son inherentes a estos. Aunque dependen de cada modelo de datos, los objetos pueden ser entidades, atributos y su dominio, las relaciones y la representación del esquema. En cuanto a las restricciones, hay tres tipos:

- **Restricciones inherentes:** son la que ya vienen impuestas por la propia naturaleza del modelo.
- **Restricciones opcionales o de usuario:** las restricciones impuestas por los diseñadores de las bases de datos atendiendo a los requisitos que deben cumplir. El modelo aporta herramientas para su gestión.
- **Restricciones libres de usuarios:** son responsabilidad del usuario y el modelo de datos ni las reconoce ni las maneja.

Las operaciones que varían con el tiempo definen las propiedades dinámicas de un modelo de datos y corresponderán con las operaciones posibles del lenguaje de manipulación de datos. La evolución de los modelos de datos es el resultado de la propia evolución técnica de los sistemas de tratamiento de datos y de su transmisión. Los modelos más usados, antes y ahora, son:

- **Modelo jerárquico:** la información se estructura a través de un esquema en forma de árbol. XML usa este tipo de modelo.
- **Modelo en red:** en este modelo cada registro puede tener más de un padre, de tal forma que el esquema que se obtiene en cuanto a las relaciones de sus nodos es un grafo, así que para acceder a un registro existen n caminos.
- **Modelo relacional:** este modelo se basa en el uso de estructuras bidimensionales para almacenar los datos denominadas relaciones. Los datos están relacionados a través de claves primarias y claves foráneas con el fin de eliminar cualquier tipo de redundancia. En la actualidad la mayoría de los sistemas gestores de bases de datos soportan este modelo de datos.
- **Modelo multidimensional:** debido a que el volumen de datos puede ser excesivamente grande, los datos y sus relaciones son estructurados en hipercubos de n dimensiones. Está orientado al análisis multidimensional y permite consultas muy avanzadas que relacionan diferentes

dimensiones jerarquizadas. OLAP es un ejemplo de este modelo.

- **Modelo orientado a objetos:** usa la filosofía de la programación orientada a objetos con el fin de adaptarse a las estructuras que se originan en programación orientada a objetos. Existe también un modelo relacional-objeto que se adapta a los sistemas gestores relacionales que existen actualmente en el mercado. Un SGBD orientado a este modelo y pionero en su campo es PostgreSQL.
- **Modelo orientado a documentos:** es un modelo en el que los datos están semi estructurados. Los registros y sus datos asociados se almacenan en documentos. Por tanto, cada documento contiene datos semiestructurados y pueden consultarse con herramientas propias para ello. Como ventaja, el esquema puede variar según convenga. MongoDB es un sistema gestor de bases de datos orientado a documentos.

1.1.2. Uso de los sistemas gestores de bases de datos

Un sistema operativo es un conjunto de programas que se comporta como interfaz entre el hardware y las aplicaciones. Cuando una aplicación debe comunicarse con la máquina lo hace a través del sistema operativo, porque es esta interfaz la que conoce cómo es la arquitectura física; es la que interpreta cada uno de los comandos para que esta pueda interpretarla. Sin los sistemas operativos, las aplicaciones informáticas deberían diseñarse e implementarse para cada tipo de máquina. Es decir, habría tantos Words como máquinas diferentes. Teniendo un único Word compatible con un sistema operativo, solo habría que diseñar diferentes Words para cada tipo de sistema operativo. Aunque depende también de otras consideraciones, esto es lo que principalmente hace un sistema operativo.

La necesidad de los sistemas operativos es indudable. Para comprobar la necesidad de los sistemas gestores de bases de datos bastaría con pensar qué ocurre con las aplicaciones si no existe ningún sistema gestor de bases de datos: serían las propias aplicaciones las que tendrían que programar los accesos a discos y capturar los datos que se requieren. Los algoritmos de accesos serían competencia de cada programador: las metodologías de acceso a discos, programas, algoritmos, registros en los archivos, estructuras en archivos, control de acceso por parte de los usuarios, visibilidad de estos con el conjunto de datos, etc. Si en vez de programar todo esto se usará una aplicación o un sistema que permitiera no preocuparse de donde se encuentra el dato físicamente, las aplicaciones que acceden a los datos serían totalmente independientes a estos sistemas. No habría que programar todo este complejo sistema de gestión de los datos físicos. Pues bien, es esto justo lo que hace un sistema gestor de bases de datos: actuar como interfaz entre las aplicaciones que requieren de acceso y los sistemas físicos de almacenamiento de datos. Es el propio sistema gestor de base de datos el que evolucionará en sus metodologías de accesos, organización y seguridad. La aplicación accede a los datos totalmente de forma independiente a esto.

Sin sistemas gestores de bases de datos es necesario la gestión de sistemas de ficheros propios para las aplicaciones que se desarrollan. Esto conllevaba una alta redundancia de los datos, ya que

cada aplicación usaba sus ficheros de almacenamiento de datos, desaprovechando el espacio físico en disco. Además, los tiempos de proceso aumentaban, pues se repetían las mismas acciones implementadas en cada aplicación. La posibilidad de información inconsistente es muy alta en este tipo de sistemas, ya que al repetirse la información en innumerables ficheros existe una alta probabilidad de volverse corruptos. Como cada aplicación está implementada de forma aislada con otras, los datos también lo están, por lo que aumenta enormemente la redundancia.

Ante estos problemas surge un nuevo enfoque, el uso de colecciones de datos recogidos y administrados por un sistema encargado de su gestión, y serán las aplicaciones las que exploten estos sistemas para acceder a la colección de datos. A esta colección se le llama **base de datos** y al conjunto de herramientas que permiten su gestión, **sistemas gestores de bases de datos (SGBD)**. En definitiva, las principales características de los datos almacenados son su acceso desde diferentes métodos, su almacenaje una sola vez, su organización y su relación.

La capacidad que un sistema tiene para referenciar a los datos que almacena de forma aislada de su evolución y de su explotación se denomina **independencia de los datos**. Para conseguir esa independencia entre las aplicaciones y los datos se construye una representación lógica separada de su representación física, compuesta por tres niveles de abstracción denominados **nivel interno, nivel conceptual y nivel externo**. El nivel conceptual contiene el diseño conceptual de los datos, el nivel interno define la estructura física de los datos y el nivel externo referencia a la visión que tienen de la base de datos los usuarios que la explotan. De este modo, se consigue una independencia lógica de los datos que permite cambiar el esquema conceptual sin cambiar las aplicaciones, y una independencia física que posibilita modificar el esquema interno sin necesidad de hacerlo en el esquema conceptual o en el esquema externo.

Esta arquitectura que provee de las ventajas antes señaladas conlleva también dificultades que versan desde la instalación o la especialización en el personal que la usa al hardware adicional. La implantación de estos sistemas es más costosa y larga y con rentabilidad medible a largo plazo.

1.1.3. Componentes de los sistemas de bases de datos

Las bases de datos principalmente están compuestas por la colección masiva de datos correspondiente a los procesos de gestión de la información. Además de la base principal, existen otros elementos en un entorno de una base de datos, y estos son:

- **Datos:** la colección de datos, tanto los propios del usuario como aquellos que permitan realizar las operaciones de consultas sobre ellos. Los datos pueden estar encriptados para aumentar la seguridad del sistema.
- **Metadatos:** son datos que registran el comportamiento de la información almacenada en la base de datos. Se dice que componen la información de la información. Toda esta información sobre cualquier objeto que existe en la base de datos se almacena en un espacio lógico especial denominado diccionario de datos.
- **Sistemas gestores de bases de datos:** compuestos por el conjunto de aplicaciones que

permiten la gestión completa de los datos almacenados en la colección de datos. Su núcleo se denomina motor y es el encargado de gestionar los accesos a los datos físicos y ofrecer un mecanismo de abstracción a los niveles superiores que requieren de información demandada por los usuarios de la base de datos.

- **Usuarios:** son cada uno de los modos de acceso a los datos. Generalmente un usuario se asocia a la persona que explota la base de datos, pero una persona puede tener diferentes modos de acceder a las bases de datos. Cada uno de ellos es un usuario. Una aplicación informática también puede ser un usuario de la base de datos. Las cuentas de usuario identifican al usuario, y emplean un identificador y una contraseña que lo autentica. Un mismo usuario puede tener diferentes modos de acceso y de explotación y hacer uso de los roles y perfiles para este cometido.

El sistema gestor de bases de datos (SGBD)

El sistema gestor de bases de datos, o SGBD, actúa como interfaz entre los datos y las aplicaciones que requieren esos datos y proporcionar mecanismos para definirlos, almacenarlos de forma óptima, mantenerlos y manipularlos. Además, ofrece potentes herramientas para la administración orientada, principalmente, a la gestión de los usuarios y el almacenamiento de los datos, gestión de instancias, control de transacciones, recuperación y copias de seguridad.

Las funciones principales que destacar de un sistema gestor de bases de datos son:

- **Definición y control centralizado:** descripción de los datos y las interrelaciones entre las estructuras que los almacenan. Mantiene toda la descripción en el diccionario de datos.
- **Manipulación de los datos:** mecanismos para consultar y transformar los datos almacenados y para añadir otros nuevos. La evolución del esquema de la base de datos debe interferir lo menos posible en las aplicaciones que interactúan con la base de datos.
- **Seguridad e integridad:** mecanismos para la gestión de accesos autorizados, el control de vistas externas de usuario y el control de las restricciones, garantizando la integridad y consistencia de los datos.
- **Tareas de administración:** copias de seguridad, recuperación ante fallos, explotación del sistema mediante roles, control de privilegios, etcétera.
- **Control de la concurrencia:** se debe garantizar el acceso simultáneo a los datos y la veracidad de estos.

Usuarios de la base de datos

Dependiendo de la finalidad de uso de la base de datos, las personas usan cuentas de usuario que se clasifican en los siguientes tipos:

Usuarios normales: emplean la base de datos a través de aplicaciones desarrolladas para interactuar con la base de datos.

Usuarios avanzados: interactúan con la base de datos directamente a través de programas especializados para la comunicación directa con el sistema gestor de bases de datos. Estos programas pueden ser de entornos textuales o gráficos. Los primeros consumen menos recursos, pero los segundos ofrecen una interfaz gráfica que hace muy cómodo el trabajo de gestión de los objetos almacenados.

Programadores: son los encargados de desarrollar las aplicaciones que actúan con interfaces entre los usuarios estándares o normales con los datos que éstas gestionan.

Administradores: son los usuarios especiales encargados del control técnico de todo el sistema. Sus funciones principales son:

- Definir el esquema conceptual.
- Definir la estructura de almacenamiento y los métodos de acceso.
- Implementación de la base de datos.
- Modificar el esquema y la organización física de los datos.
- Conceder permisos y privilegios para el acceso a los datos.
- Definir los mecanismos para las copias de seguridad y su recuperación.
- Supervisar el rendimiento del sistema.

Lenguajes de bases de datos

La interacción del usuario con las bases de datos debe efectuarse a través de alguna técnica que haga fácil la comunicación, y que permita al usuario centrarse en el problema que desea solucionar, más que en la forma de expresarlo con las técnicas que se le suministran. La mejor forma de alcanzar este objetivo es darle un lenguaje parecido al lenguaje natural, que le permita expresar de forma sencilla los requerimientos.

Los lenguajes en los sistemas gestores de bases de datos permiten la comunicación a la base de datos a través de comandos. Se persigue que estos comandos posean el mayor nivel de abstracción posible y se diseñan con el fin de que se parezcan al lenguaje natural con los elementos léxicos, sintácticos y semánticos necesarios y suficientes para aportar la máxima flexibilidad posible.

Dependiendo de la finalidad del conjunto de comando que pertenece al lenguaje, se establecen varios tipos:

Lenguaje de definición de datos, Data Definition Language, DDL: son los comandos que usa el administrador de la base de datos para la construcción de los elementos de esta. Ejemplos: CREATE, ALTER y DROP

Lenguaje de manipulación de datos, Data Manipulation Language, DML: son los comandos que usan los usuarios avanzados que permiten efectuar las operaciones sobre las bases de datos. Estas operaciones son el alta de nuevos datos, la supresión de datos registrados, modificación de valores que cambian, las consultas de datos y la gestión de vistas. Estos comandos pueden embeberse en otros lenguajes de programación que los programadores usan para desarrollar las aplicaciones. Ejemplos: SELECT, INSERT, DELETE Y UPDATE.

Lenguaje de control de datos, Data Control Language, DCL: son los comandos que generalmente usan los administradores para controlar el acceso a los datos. Ejemplos: GRANT y REVOKE.

A continuación, se ejemplifican sistemas gestores de bases de datos actuales y sus características más interesantes:

Access de Microsoft: manejo muy sencillo, para bases de datos muy pequeñas; fácil de usar en webs, orientado a usuarios individuales o pequeños equipos.

Apache Derby: tamaño reducido, escrito en Java, multiplataforma, muy portable, embebido o cliente/servidor.

CouchDB: orientado a documentos, NoSQL, usa JSON, licencia Apache.

DB2 de IBM: multiplataforma, usa XML. Su versión gratuita se llama DB2 Express.

FireBird: relacional, multiplataforma, bajo consumo, concurrencia óptima. Código abierto.

Informix de IBM: sencillo, confiable, relacional y multiplataforma. Usa XML. Consume menos recursos que Oracle.

MariaDB: derivado de MySQL y con licencia GPL. Multiplataforma.

MongoDB: NoSQL, usa BSON con esquema dinámico. Multiplataforma.

MySQL: comercial y libre. Es relacional, multihilo, multiusuario y multiplataforma. Orientado a entornos web...

SQL Server de Microsoft: cliente/servidor, relacional, competidor de Oracle.

Oracle: multiplataforma, confiable y seguro, cliente/servidor, relacional, uso de JSON. Ofrece una versión gratuita Express Edition. Es un producto muy competitivo.

OrientDB: orientado a documentos con licencia Apache.

PostgreSQL: sistema relacional-objetos. Es el SGBD libre más avanzado del mundo. Multiplataforma.

SQLite: Relacional, accesos muy rápidos y multiplataforma. winre

SyBase: Relacional, altamente escalable y de bajo costo.

1.2. Las bases de datos en la producción de software.

Las bases de datos se deben integrar en el ciclo de vida de la ingeniería del software, ya que esta es una parte del conjunto completo de la producción de aplicaciones. El concepto de ciclo de vida se utilizó para modelar el proceso de ingeniería del software y se puede definir como el conjunto de etapas por las que atraviesa el sistema desde el análisis, diseño, desarrollo y explotación. En estas fases se deben definir las actividades que realizar, asegurando consistencia con el resto de los sistemas de información y proporcionando controles para la gestión del proyecto.

1.2.1. Los procesos en la producción de software

La norma 12207 establece diferentes procesos que se pueden sintetizar como sigue:

Proceso de adquisición: el proceso de adquisición supone una etapa inicial en la ingeniería del software, y, aunque es externa de la parte que desarrolla el producto, es vinculante con los inicios. Estas son las actividades que desarrolla el cliente que contrata el producto.

Proceso de suministro, Planning: hace referencia al proceso que el proveedor del producto desarrolla. Contiene los mecanismos para dar respuestas óptimas, planificando los procedimientos, detallando los recursos que gestionar con el fin de garantizar el éxito del producto.

Proceso de análisis, Analyze: hace referencia a la detección de los requisitos de todas las áreas vinculantes con la producción del producto software. El objetivo final es la recopilación de todos los requisitos y sirve de entrada al proceso de diseño.

Proceso de diseño, Design: contiene todas las etapas de diseño conceptual y lógico del producto que se va a elaborar. Es uno de los procesos más importantes, ya que el resto depende de la correcta elaboración de los mecanismos de diseño.

Proceso de desarrollo, Development: contiene las actividades de análisis de requisitos, diseño, codificación, integración, pruebas e instalación y aceptación. Una vez que se ha diseñado el producto, se comienza con las fases relacionadas con la implementación del producto y cada uno de los mecanismos para su programación.

Proceso de validación y verificación, Validation, Verification y Testing: el producto comienza a usarse como si estuviera en posproducción. Se testean todos los posibles itinerarios de ejecución.

Proceso de explotación y mantenimiento: el producto entra en su ejecución real, haciéndose necesario revisiones constantes y detección de necesidades de mejoras.

1.2.2. Ciclo de vida en las bases de datos

El ciclo de vida en las bases de datos está totalmente integrado en la ingeniería del software y sus

fases están relacionadas con ellas. Principalmente se distinguen las siguientes fases:

Planificación: qué se ha de realizar, con qué recursos se cuenta y qué técnicas se usan para la recolección de datos, su formato y documentación, y cómo se llevará el diseño y la implementación.

Definición: se especifica el ámbito y los límites de la aplicación de bases de datos, así como los sistemas con los que interactúa. Se determinan los usuarios y las áreas de aplicación.

Recogida y análisis de los requisitos: se recogen y analizan los requerimientos de los usuarios y de las áreas de aplicación a través de entrevistas, observación del funcionamiento de la organización, análisis de documentos e instrumentos de recogida de información, etcétera.

Diseño de la base de datos: está dividido en tres fases importantes:

Diseño conceptual: se produce un esquema conceptual independiente del nivel físico del sistema.

Diseño lógico: tras refinar el esquema conceptual se obtiene el esquema lógico, en el que se ha eliminado toda construcción que no se puede explotar con el modelo de datos que se haya escogido. Ejemplos de modelos de datos son el relacional, el orientado a objetos, el relacional-objeto, el documental, etcétera.

Diseño físico: atendiendo al SGBD elegido que más se adecua al sistema de información, el esquema lógico se traduce en un esquema físico. En este, se tiene en cuenta las estructuras de almacenamiento y los métodos de acceso.

Diseño de la aplicación: diseño de los programas de aplicación que interactúan con la base de datos, y en muchas ocasiones es paralela al diseño de esta. Se debe asegurar que los requisitos de usuario se encuentran en el diseño de la aplicación. También se puede diseñar las interfaces de usuario.

Prototipado: un prototipo es un modelo de trabajo de las aplicaciones del sistema y es útil para que los usuarios puedan usar el sistema e identificar qué aspectos son o no adecuados. El prototipado ayuda al equipo de diseño e implementación a detectar incorrectas interpretaciones de los requisitos de los usuarios.

Implementación: haciendo uso de los comandos del lenguaje de definición de datos se crea el esquema de la base de datos, el sistema de almacenamiento en ficheros y el esquema de los usuarios. Se implementan los procesos que permiten comunicarse con la base de datos y trabajar con los datos. También se usan los comandos de manejo de datos que son llamados desde scripts o desde aplicaciones desarrolladas con otros lenguajes de programación.

Conversión y carga de datos: cuando se actualizan sistemas y se debe reemplazar los datos antiguos por nuevos incluidos en diferentes estructuras.

Prueba y mantenimiento: se prueba y valida los requisitos de los usuarios, se diseñan test con datos reales y se toman medidas sobre la fiabilidad y calidad del software desarrollado.

1.3. Ubicación e integración de las bases de datos

La arquitectura más tradicional para el tratamiento de datos consiste en centralizar todos los datos en una máquina encargada de facilitar los accesos a los archivos almacenados en sus sistemas de almacenamiento masivos. Los diferentes clientes se conectan a esta máquina central cada vez que requiere datos que esta almacena. La arquitectura física de esta organización generalmente está soportada por una serie de equipos conectados a través de una red de área local. La máquina que soporta la base de datos también está conectada a la red, con configuraciones físicas y lógicas que permiten acelerar los accesos y asegurarlos.

A la máquina que provee este servicio se le llama **servidor** y a las máquinas que usan el servicio, **clientes**. Esta es la **arquitectura cliente/servidor** tradicional que se ha estado usando durante mucho tiempo. Los programas informáticos que las organizaciones usan y que requieren altas comunicaciones con los almacenes de datos están compuestos por cuatro componentes: las bases de datos, la lógica de transacciones, la lógica de negocios y de la aplicación de gestión de datos y la interfaz de usuarios. Atendiendo al modo en el que estos componentes se desarrollan en el lado del cliente y en el lado del servidor se obtienen diferentes modelos.

En la **arquitectura de dos niveles**, el nivel del cliente y el nivel del servidor, el primero se encarga de la presentación de los datos al usuario, es decir, la interfaz de usuario y la lógica de negocio, y el segundo de soportar los datos a través de herramientas destinadas para esta función, los sistemas gestores de bases de datos. Estos sistemas pueden ser SGBD de diferentes modelos, relacionales, documentales, orientados a objetos, objeto- relacionales o también sistemas propietarios de gestión de datos.

En la **arquitectura de tres niveles** se ofrece mayor escalabilidad al aportar un nivel más que separa el nivel de interfaz de usuario con el nivel de lógica de negocio y procesamiento de datos. En el lado del cliente se elaboran los aspectos asociados a la interfaz de usuario y que ejecuta el usuario final. En el nivel intermedio se emplea un servidor de aplicaciones para procesar los datos y llevar a cabo los aspectos asociados con la lógica de negocio. En el nivel del servidor, se implantan sistemas gestores de bases de datos que proveen mecanismos para la entrada y salida de datos que requiere el nivel intermedio. Los servidores de aplicaciones gestionan la lógica de negocios y el acceso a los datos disminuyendo la complejidad del desarrollo de aplicaciones. La conexión física de las máquinas de esta arquitectura puede estar provista a través de redes locales o redes extensas.

La **seguridad** de las bases de datos es un tema que aumenta en importancia, ya que las redes que soportan las nuevas arquitecturas están provistas de posibles mecanismos inseguros y que permiten

vulnerabilidades que pueden provocar efectos no deseados en la explotación de estas. Las bases de datos deben estar altamente protegidas en las arquitecturas locales, a través de dispositivos cortafuegos, servidores proxy y redes desmilitarizadas. Los mecanismos de protección tanto lógicos como físicos deben aumentar, al hacerlo las posibles amenazas y sus consecuentes pérdidas de disponibilidad, de confidencialidad, de privacidad o de integridad.

La mayoría de los sistemas gestores de bases de datos proveen de mecanismos de **control de acceso y autorización**. El control de acceso discrecional y el obligatorio son ejemplos de estos. Las vistas también son un mecanismo que aumenta la seguridad de las bases de datos, ya que ocultan al usuario aquella parte a la que no debe tener acceso. Por ejemplo, los efectivos de los procesos de ventas en una organización no deben acceder a los datos contables, financieros o de cualquier otra naturaleza que no sean los propios del proceso de venta.

Las **copias de seguridad y de restauración** son tareas muy importantes en el lado del servidor. El administrador y cualquier usuario con funciones destinadas a tales efectos deben establecer protocolos exhaustivos sobre el mantenimiento de los datos, la recuperación ante desastres y operatividad con datos que no intervienen en la operatividad presente de la organización. La **alta disponibilidad** también es una tarea importante, a través de mecanismos de balanceo de carga, duplicidad lógica o física, sistemas físicos de redundancia de datos, etcétera.

El **cifrado de la información** también juega un papel muy importante para proteger el activo de cualquier organización: los datos. Estos pueden cifrarse a través de diferentes metodologías y algoritmos, explotación de certificados digitales y uso de protocolos seguros. Los datos almacenados en los soportes pueden estar cifrados para evitar su conocimiento en caso de cualquier intrusismo. Implantar **servidores de autenticación**, como por ejemplo Kerberos, también mejora la seguridad de los sistemas gestores de bases de datos.

Las **bases de datos distribuidas** también son soluciones actuales. Los datos están distribuidos en diferentes máquinas que ofrecen datos y el SGBD distribuido permite gestionarlos haciendo que esta distribución sea transparente para los usuarios. En este tipo de arquitectura los datos están repartidos o replicados, llamados **fragmentos**, y trabajan bajo redes de comunicaciones locales y no locales, a través de intranets o extranets y que usan tanto aplicaciones locales (aquellas que los datos que requieren están en comunicación directa) como aplicaciones globales (que no lo están y requieren de otras máquinas).

La **integración web** con los sistemas gestores de bases de datos conlleva nuevos requisitos y protocolos que desarrollar. No solo para la explotación local en una organización a través de su intranet o extranet, sino también para las nuevas arquitecturas basadas en la nube.