

Unidad II

Software de cómputo

2.1. Sistemas operativos.

Un **sistema operativo (SO)**, frecuentemente **OS**, del inglés *Operating System*) es un **programa** o conjunto de programas que en un sistema informático gestiona los recursos de **hardware** y provee servicios a los **programas de aplicación**, ejecutándose en modo privilegiado respecto de los restantes y anteriores próximos y viceversa.²

Nótese que es un error común muy extendido denominar al conjunto completo de herramientas sistema operativo, es decir, la inclusión en el mismo término de programas como el **explorador de ficheros**, el **navegador web** y todo tipo de herramientas que permiten la interacción con el sistema operativo, también llamado núcleo o **kernel**. Esta identidad entre *kernel* y sistema operativo es solo cierta si el núcleo es **monolítico**. Otro ejemplo para comprender esta diferencia se encuentra en la plataforma **Amiga**, donde el **entorno gráfico de usuario** se distribuía por separado, de modo que, también podía reemplazarse por otro, como era el caso de **directory Opus** o incluso manejarlo arrancando con una **línea de comandos** y el **sistema gráfico**. De este modo, al arrancar un Amiga, comenzaba a funcionar con el **propio sistema operativo** que llevaba incluido en una **ROM**, por lo que era cuestión del usuario decidir si necesitaba un entorno gráfico para manejar el sistema operativo o simplemente otra aplicación. Uno de los más prominentes ejemplos de esta diferencia, es el **núcleo Linux**, usado en las llamadas **distribuciones Linux**, ya que al estar también basadas en **Unix**, proporcionan un sistema de funcionamiento similar. Este error de precisión, se debe a la modernización de la informática llevada a cabo a finales de los 80, cuando la filosofía de estructura básica de funcionamiento de los grandes computadores³ se rediseñó a fin de llevarla a los hogares y facilitar su uso, cambiando el concepto de computador **multiusuario**, (muchos usuarios al mismo tiempo) por un sistema monousuario (únicamente un usuario al mismo tiempo) más sencillo de gestionar.⁴ (Véase **AmigaOS**, **beOS** o **MacOS** como los pioneros⁵ de dicha modernización, cuando los Amiga fueron bautizados con el sobrenombre de *Video Toasters*⁶ por su capacidad para la **Edición de vídeo** en entorno **multitarea round robin**, con **gestión de miles de colores** e interfaces **intuitivos** para **diseño en 3D**).

Uno de los propósitos del sistema operativo que gestiona el núcleo **intermediario** consiste en gestionar los recursos de localización y protección de acceso del hardware, hecho que alivia a los programadores de aplicaciones de tener que tratar con estos detalles. La mayoría de aparatos electrónicos que utilizan **microprocesadores** para funcionar, llevan incorporado un sistema operativo (teléfonos móviles, **reproductores de DVD**, computadoras, radios, **enrutadores**, etc.). En cuyo caso, son manejados mediante una **Interfaz**

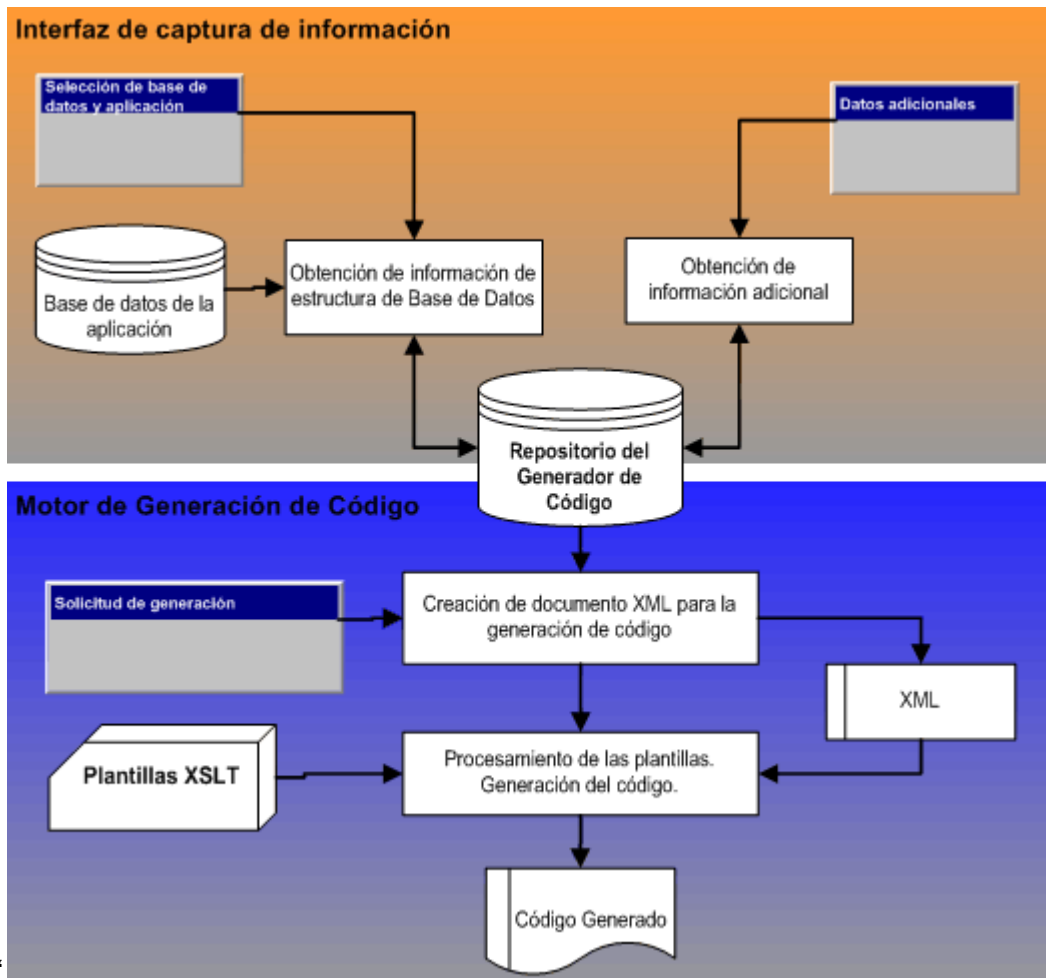
Gráfica de Usuario, un gestor de ventanas o un entorno de escritorio, si es un celular, mediante una consola o control remoto si es un DVD y, mediante una línea de comandos o navegador web si es un enrutador.

2.2. Bases de datos.

ARQUITECTURA DE UNA BASE DE DATOS

la arquitectura de base de datos consta de tres niveles , externo, conceptual y interior. Es evidente que la separación de los tres niveles fue una de las principales características de la modelo de base de datos relacional que domina las bases de datos del siglo 21.[1] El nivel externo define cómo los usuarios a comprender la organización de los datos. Una sola base de datos puede tener cualquier número de puntos de vista a nivel externo. El nivel interno define cómo los datos se almacenan físicamente y procesados por el sistema de cómputo . arquitectura interna se refiere a costo, rendimiento , escalabilidad y otras cuestiones operativas. El conceptual es un nivel de direccionamiento indirecto entre lo interno y externo. Proporciona una visión común de la base de datos que es complicado por los detalles de cómo los datos son almacenados o manejados, y que se pueden unificar los diversos puntos de vista externo en un todo coherente.

Victor feria La base de datos son varios datos recopilados para ser utilizados despues como por ejemplo en una tienda la base de datos seria todos los productos que tienen ala venta o vendieron se puede representar como documentos graficas osea q en este caso de la tienda se utilizaria para buscar el articulo ver el precio de x articulo etc.



SISTEMA DE GESTION DE LA BASE DE DATOS

Un sistema de gestión de bases de datos (DBMS) consiste en software que opera bases de datos, el almacenamiento , acceso, seguridad , mecanismos de seguridad y otros . sistemas de gestión de base de datos se pueden clasificar de acuerdo con el modelo de la base de datos que apoyo, como relacional o XML, El tipo (s) del equipo en el que apoyo, como un clúster de servidores o un teléfono móvil, el lenguaje de consulta(S) que acceden a la base de datos, tales como SQL o [X Query?](#), el desempeño del comercio -offs , como la escala máxima o la velocidad máxima o de otros. Algunos DBMS cubrir más de una opción en estas categorías , por ejemplo , soportar múltiples lenguajes de consulta.

Componentes del SGBD

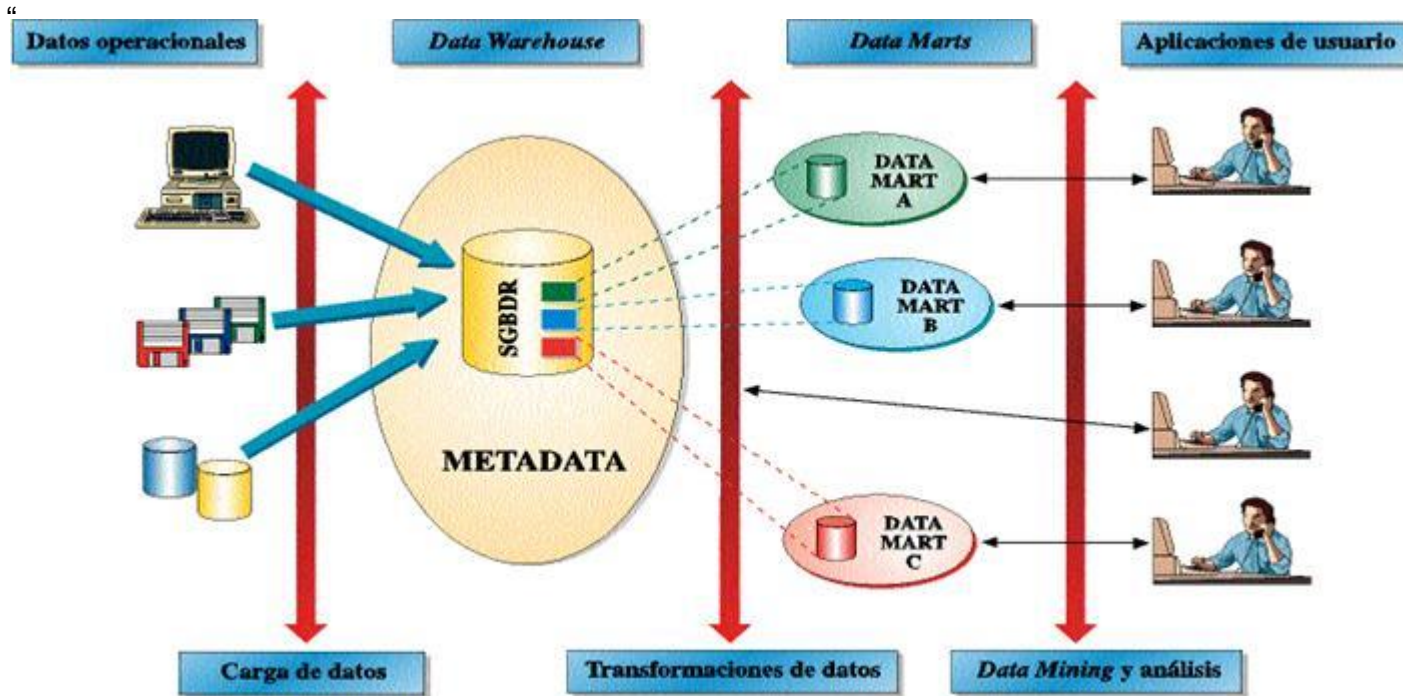
La mayoría de DBMS a partir de 2009 implementar un modelo relacional.[2] Otros sistemas de DBMS, tales como objetos de DBMS, ofrecen características específicas para las necesidades más especializadas. Sus componentes son

similares, pero no idénticos. RDBMS componentes • Sublenguajes - Relational DBMS (RDBMS) incluyen Data Definition Language (DDL) para definir la estructura de la base de datos , Lenguaje de control de datos (DCL) para la definición de la seguridad / control de acceso , y Lenguaje de Manipulación de Datos (DML) para consultar y actualizar datos . • Interfaz de controladores- Estos controladores son bibliotecas de código que proporcionan métodos para preparar declaraciones, ejecutar instrucciones , ir a buscar los resultados, etc Ejemplos incluyen ODBC, JDBC, MySQL/PHP, [Fire Bird?](#)/Pitón. • SQL del motor- Este componente interpreta y ejecuta el DDL, DCL, Y LMD declaraciones. Incluye tres componentes principales (compilador , optimizador y ejecutor) . • Transacción del motor- Asegura que varias instrucciones SQL o bien tienen éxito o fracasan como grupo, según la aplicación dictados . • motor relacionalobjetos - relacional como tablas, índices y restricciones de integridad referencial se implementan en este componente. • Motor de almacenamiento- Este componente almacena y recupera datos desde el almacenamiento secundario , así como la gestión de transacciones ejecuta o deshace , backup y recuperación, etc

ODBMS componentes

Objeto DBMS (ODBMS) ha transacción y componentes de almacenamiento que son análogas a las de un RDBMS. Algunos ODBMS manejar DDL , DCL y actualizar las tareas de manera diferente. En lugar de utilizar sublenguajes , proporcionan APIs para estos fines. Suelen incluir un sublenguaje y adjunto del motor para procesar consultas con las declaraciones interpretativas análogas a la misma , pero no como SQL . Ejemplo de objetos lenguajes de consulta son OQL, LINQ, JDOQL, JPAQL y otros. El motor de consultas devuelve colecciones de objetos en lugar de filas relacional.

base de datos operacionales



“

Estas bases de datos almacenan datos detallados sobre las operaciones de una organización. Normalmente son organizados por temas, el proceso de volúmenes relativamente altos de las actualizaciones utilizando transacciones. Esencialmente todas las organizaciones principales en la tierra utiliza esas bases de datos. Los ejemplos incluyen bases de datos de clientes que el crédito en contacto con registro, y la información demográfica acerca de un negocio a los clientes, bases de datos que contienen información personal, tales como salario, beneficios, capacidades de datos sobre los empleados, bases de datos de fabricación que los datos de registro sobre los componentes de productos, inventario de piezas y bases de datos financieros que seguir la pista del dinero de la organización, la contabilidad y las transacciones financieras.

Victor feria

la base de datos operacional opera de día en día o sea que cada día se esta renovando el proceso que utiliza es repetitivo su volatilidad es actualizable la perspectiva de esta base de datos es tener los datos actuales

Almacén de los datos

Datos almacenes archivo histórico de datos desde bases de datos operacionales y, a menudo procedentes de fuentes externas tales como empresas de

investigación de mercado. A menudo, los datos operativos sufre la transformación en su camino hacia el almacén , consiguiendo resumen , anónimos , recalificada , etc El almacén se convierte en la fuente central de datos para su uso por los gerentes y otros usuarios finales que no pueden tener acceso a los datos operativos . Por ejemplo , los datos de ventas puede ser agregada a los totales semanales y convertir de códigos de los productos internos para el uso UPC los códigos para que pueda ser comparado con [AC Nielsen?](#) de datos.

base de datos analíticos

Los analistas pueden hacer su trabajo directamente en contra de un almacén de datos , o crear una base de datos analítica independiente para Procesamiento analítico en línea. Por ejemplo , una empresa puede extraer los registros de ventas para analizar la efectividad de la publicidad y las promociones de ventas que a nivel agregado .

base de datos distribuidas

Estas son bases de datos de los trabajadores locales grupos o departamentos en las oficinas regionales, sucursales , fábricas y otros lugares de trabajo . Estas bases de datos puede incluir segmentos de ambas bases de datos de usuario común operacionales y comunes , así como los datos generados y utilizados exclusivamente en el propio sitio de un usuario.

base de datos de usuario final

Estas bases de datos son los datos desarrollados por cada uno de los usuarios finales. Ejemplos de estos son colecciones de documentos en hojas de cálculo, procesadores de texto y los archivos descargados , o incluso la gestión de su colección personal de tarjetas de béisbol.

Base de datos externa

Estas bases de datos contienen datos recogen para su utilización en múltiples organizaciones , ya sea gratuitamente o mediante suscripción . El Internet Movie Database es un ejemplo.

Hipermedia bases de datos

El A todo el mundo web puede ser pensada como una base de datos , aunque una difusión a través de millones de sistemas informáticos independientes. Los

navegadores web “Proceso” estos datos una página a la vez , mientras que rastreadores web y otro software proporcionan el equivalente de los índices de base de datos para apoyar la búsqueda y otras actividades.

Modelos

Victor feria Los modelos de bases de datos se dividen en niveles bajos y físicos estos disponen o describen los detalles sobre el almacenamiento de los datos de una computadora y los modelos de datos de representación: disponen de conceptos que pueden entender los usuarios determinados , pero que no diferentes a la forma en que se almacenan los datos en la computadora.

Post- relacionales modelos de base de datos

Productos que ofrece un modelo de datos más general que el modelo relacional a veces son clasificados como post-relacional.[3] términos alternativas son la “ base de datos “ híbrido “, RDBMS de objetos mejorada “y otros. El modelo de datos en dichos productos incorpora relaciones pero no está limitado por E. F. Codd’ s Principio de Información , que exige que toda la información en la base de datos deberán ser emitidos de forma explícita en términos de valores en las relaciones y de ninguna otra manera[4] Algunas de estas extensiones al modelo relacional integrar los conceptos de tecnologías que antes de la fecha en que el modelo relacional. Por ejemplo, permitir la representación de una grafo dirigido con árboles en el nodos. Algunos productos de post- relacional de ampliar los sistemas de relación con las características no relacionales . Otros llegaron en gran parte el mismo lugar por la adición de características relacionales para los sistemas de pre -relacional . Paradójicamente, esto permite que los productos que están a pre -relacionales, como PICK y PAPERAS, Para hacer una afirmación plausible de ser post-relacional .

Objeto modelos de base de datos

En los últimos años, La orientada a objetos paradigma se ha aplicado en las áreas de ingeniería y bases de datos espaciales , las telecomunicaciones y en diversos ámbitos científicos. El conglomerado de programación orientada a objetos y la tecnología de base de datos llevado a este nuevo tipo de base de datos. Estas bases de datos intento de llevar la base de datos mundial y la aplicación de programación del mundo más cerca, en particular, garantizando que la base de datos utiliza el mismo sistema de tipos como el programa de aplicación. Con ello se pretende evitar la sobrecarga (a veces denominado “el impedancia de falta de coincidencia) de convertir la información entre su representación en la base de

datos (por ejemplo, las filas en las tablas) y su representación en el programa de aplicación (típicamente como objetos) . Al mismo tiempo , bases de datos objeto de intento de introducir las ideas clave de la programación de objetos, tales como encapsulación y polimorfismo, en el mundo de bases de datos. Una variedad de estas formas se han probado[¿por quién?] para almacenar objetos en una base de datos. Algunos productos han abordado el problema desde el lado de programación de aplicaciones , haciendo que los objetos manipulados por el programa persistente. Esto también suele requerir la adición de algún tipo de lenguaje de consulta , ya que los lenguajes convencionales de programación no proporcionan la funcionalidad del lenguaje de nivel para encontrar los objetos en función de su contenido de información. Otros[¿cuáles?] han atacado el problema desde el extremo base de datos, mediante la definición de un modelo de datos orientado a objetos para la base de datos , y la definición de una base de datos lenguaje de programación que permite a todas las capacidades de programación, así como las instalaciones tradicionales de consulta.

Instalaciones de almacenamiento

Artículo principal : Base de datos de estructuras de almacenamiento Bases de datos puede almacenar tablas relacionales / índices en la memoria o el disco duro en una de muchas formas: • pedidas o desordenadas archivos planos • ISAM • montones • cubos hash • lógicamente - archivos bloqueados • B + árboles El más comúnmente utilizado[cita requerida] son árboles B + y ISAM. bases de datos de objetos de utilizar una serie de mecanismos de almacenamiento . Algunos utilizan virtual archivos asignados en memoria para que la lengua materna (C + +, Java etc) los objetos persistentes. Esto puede ser muy eficaz pero puede hacer que el acceso en varios idiomas más difíciles. Otros objetos en desmonte fijo y componentes variables de longitud que se agrupan en bloques de tamaño fijo en el disco y volver a montar en el formato apropiado en el cliente o el espacio de direcciones del servidor. Otra técnica popular consiste en almacenar los objetos de tuplas (muy similar a una base de datos relacional) que el servidor de base de datos de volver a montarlo en objetos para el cliente.[cita requerida] Otras técnicas incluyen agrupación por categorías (por ejemplo, agrupando los datos por mes, o la ubicación) , el almacenamiento de resultados de la consulta pre-computadas , conocido como vistas materializadas, la partición de datos por su distribución (por ejemplo , un rango de datos) o hash. topología de administración de memoria y de almacenamiento pueden ser importantes decisiones de diseño para diseñadores de base de datos también. Así como normalización se utiliza para reducir los requerimientos de almacenamiento y mejorar el diseño de base de datos , a la inversa desnormalización se utiliza a menudo para reducir la complejidad y unirse a reducir el tiempo de ejecución de la consulta .[5]

Indexación

Indexación es una técnica para mejorar el rendimiento de base de datos. Los muchos tipos de índice de acciones de la propiedad común que elimina la necesidad de examinar todas las entradas cuando se ejecuta una consulta. En grandes bases de datos, esto puede reducir el tiempo de consulta / costo por órdenes de magnitud. La forma más simple de índice es una lista ordenada de valores que se pueden buscar usando una búsqueda binaria con una referencia adyacentes a la ubicación de la entrada, de forma análoga a la del índice en la parte posterior de un libro. Los mismos datos pueden tener varios índices (base de datos de un empleado puede ser indexados por apellido y fecha de contratación .) Los índices afectan al rendimiento, pero no resultados. Base de datos de los diseñadores pueden agregar o eliminar índices sin cambiar la lógica de aplicación, reduciendo los costes de mantenimiento como la base de datos crece y se desarrolla el uso de base de datos. Dada una consulta en particular, el DBMS " optimizador de consultas es responsable de la elaboración de la estrategia más eficiente para encontrar datos coincidentes. El optimizador decide qué índice o índices de usar, la manera de combinar datos de distintas partes de la base de datos, la manera de presentar los datos en el orden solicitado, etc Los índices pueden acelerar el acceso de datos, pero consumen espacio en la base de datos, y debe ser actualizado cada vez que los datos son alterados. por lo tanto los índices pueden acelerar el acceso de datos, pero el mantenimiento de datos lenta. Estas dos propiedades determinar si un determinado índice vale la pena el costo.

Transacciones

La mayoría de DBMS proporcionar algún tipo de apoyo a transacciones, Que permiten que varios elementos de datos que se actualizará de manera coherente, de manera que las actualizaciones que se parte de una transacción tienen éxito o fracasan al unísono. La llamada ACID normas, que se resumen aquí, caracterizan a este comportamiento:

- Atomicidad: O todos los cambios de datos en una transacción tiene que pasar, o ninguno de ellos. La transacción debe ser completada, o bien debe ser deshecho (parte posterior laminado).
- Consistencia: Cada transacción debe preservar las reglas de consistencia declarado para la base de datos.
- Aislamiento: Dos transacciones simultáneas no pueden interferir unos con otros. Los resultados intermedios dentro de una transacción debe seguir siendo invisibles para otras transacciones. La forma más extrema de aislamiento es seriability, Lo que significa que las transacciones que tienen lugar simultáneamente al contrario, podría llevarse a cabo en algunas series, sin afectar el resultado final.
- Durabilidad: las transacciones realizadas no se puede

interrumpir , o si sus resultados descartados. Se debe persistir a través de (por ejemplo) se reinicia DBMS. En la práctica, muchos DBMS permiten la relajación selectiva de estas normas para equilibrar el comportamiento perfecto con un rendimiento óptimo.

Replicación

Base de datos de replicación implica el mantenimiento de múltiples copias de una base de datos en equipos diferentes, para permitir que más usuarios para acceder a ella , o permitir que un sitio secundario de tomar de inmediato sobre si el sitio primario deja de funcionar. Algunos de replicación cuestan DBMS en la parte superior de sus instalaciones de registro de transacciones , la aplicación de registro de la primaria a la secundaria en tiempo casi real . clustering de base de datos es un concepto relacionado para el manejo de grandes bases de datos y de usuarios mediante el empleo de un racimo de varios equipos para acoger una base de datos única que puede utilizar la replicación como parte de su enfoque Victor feria La replica de una base de datos es para tener seguridad y tener control de todo lo que entra o sale de la base y para recopilar datos pasdos con los presentes

Seguridad

Base de datos de seguridad denota el sistema , los procesos y procedimientos que protegen a una base de datos de actividad no autorizada. DBMS por lo general a través de reforzar la seguridad control de acceso, auditoría, Y cifrado: • El control de acceso que gestiona se puede conectar a la base de datos a través de autenticación y lo que pueden hacer a través de autorización. • Auditoría de los registros de base de datos información sobre la actividad : quién, qué, cuándo, y posiblemente dónde. • El cifrado protege los datos en el nivel más bajo posible mediante el almacenamiento y, posiblemente, la transmisión de datos en un formato ilegible . El DBMS encripta los datos cuando se agrega a la base de datos y descifra al regresar resultados de la consulta . Este proceso puede producirse en el lado cliente de una conexión de red para evitar el acceso no autorizado en el punto de uso.

Confidencialidad

Ley y Reglamento prevé la entrega de información de algunas bases de datos , la protección de la historia clínica , registros de conducir , teléfono troncos, etc En el Reino Unido, La regulación de base de datos corresponde a la privacidad Oficina del Comisionado de la Información. Las organizaciones con sede en el Reino

Unido y almacenamiento de datos personales en formato digital , tales como bases de datos deben inscribirse en la Oficina .[8]

Cierre

Cuando una transacción modifica un recurso, el DBMS paradas que otras transacciones también se modifica , por lo general por cierre mismo. Cierre también proporcionan un método para asegurar que los datos no cambia , mientras que una transacción está leyendo o incluso que no cambiará hasta que una transacción que una vez leído se ha completado .

Granularidad

Las cerraduras pueden ser grueso , con una base de datos , de grano fino , que abarca un conjunto de datos solo punto , o intermedio que abarca una colección de datos, como todas las filas de una tabla de RDBMS.

Bloqueo de tipos

Las cerraduras pueden ser compartido[9] o exclusivo, Y se puede bloquear la lectores y / o escritores. Las cerraduras pueden ser creados implícitamente por el DBMS cuando una transacción realiza una operación , o explícitamente a petición de la transacción. bloqueos compartidos permiten que las transacciones múltiples para bloquear el mismo recurso. El bloqueo persiste hasta que todas las transacciones tan completa . bloqueos exclusivos están en manos de una sola transacción y evitar que otras transacciones de cierre el mismo recurso. bloqueos de lectura tienden a ser compartidas , y evitar que otras transacciones modifiquen los recursos. Escriba cerraduras son exclusivos, y evitar que otras transacciones modificar el recurso. En algunos sistemas, bloqueos de escritura también impiden que otras transacciones de la lectura del recurso. El DBMS implícitamente bloqueos de datos cuando se actualiza, y también puede hacerlo en el momento que se lea. Las transacciones de forma explícita bloquean los datos para asegurarse de que puede completar sin punto muerto u otras complicaciones. cerraduras explícito puede ser útil para algunas tareas administrativas .[10][11] Bloqueo puede afectar significativamente el rendimiento de base de datos , especialmente con las operaciones grandes y complejas en entornos altamente concurrentes.

Aislamiento

Aislamiento se refiere a la capacidad de una transacción para ver los resultados de otras transacciones. Mayor aislamiento normalmente reduce el rendimiento y / o concurrencia, DBMS líder para ofrecer opciones administrativas para reducir el aislamiento. Por ejemplo, en una base de datos que se analizan las tendencias en lugar de buscar a un bajo nivel de detalle, el aumento de rendimiento puede justificar que se permita a los lectores a ver cambios no confirmados ("lecturas sucias").

Interbloqueos

Interbloqueos se producen cuando dos operaciones cada uno requiere de datos que el otro ya ha bloqueado exclusivamente. Detección de interbloqueo se lleva a cabo por el DBMS, que luego se anula una de las operaciones y permite a la otra para completarse.

2.3. Lenguajes de desarrollo.

Un lenguaje de programación es un lenguaje artificial diseñado para expresar cálculos que se pueden realizar por una máquina, particularmente una computadora. Los lenguajes de programación se pueden utilizar para crear programas que controlan el comportamiento de una máquina, para expresar algoritmos de precisión, o como un modo de comunicación humana.

```
0100000000010100011011000000100101100011
110001011101000100011111111110100000100
00101001011000011010111011010110110010001
01101100000101011001000100001110001001111
0100110010110100110110100111101111011110
00011010011010011010011010011110111101110
0100100110100110100110100111010001110
110001001int main()
010101001{
111001100 printf("Hello World")
00100000111 return 42;
000110100}
01001001101111010111011110000001010001110
11000100100010101100100111011101000101111
01010100111001101010111000101010100011000
1111001100000110111111010100111110001100
0100000111111101010010010011010101110110
```



Los primeros lenguajes de programación son anteriores a la invención de la computadora, y fueron utilizados para dirigir el comportamiento de las máquinas, tales como telares Jacquard y pianolas. Miles de diferentes lenguajes de programación se han creado, sobre todo en el campo de la computación, y muchos más se crean cada año. La mayoría de los lenguajes de programación describen de computación en un imperativo de estilo, es decir, como una secuencia de comandos, aunque algunos idiomas, tales como los que apoyan la programación funcional y programación lógica, el uso de formas alternativas de descripción.



Un lenguaje de programación se suele dividir en los dos componentes de la sintaxis (forma) y la semántica (significado) y muchos lenguajes de programación tienen algún tipo de especificación por escrito de su sintaxis o semántica. Algunos idiomas se definen por un documento de especificación, por ejemplo, el C lenguaje de programación se especifica mediante una ISO estándar, mientras que otros idiomas, tales como Perl, tienen una posición dominante en la aplicación que se utiliza como referencia.



Para que la computadora entienda nuestras instrucciones debe usarse un lenguaje específico conocido como código máquina, el cual la máquina comprende fácilmente, pero que lo hace excesivamente complicado para las personas. De hecho sólo consiste en cadenas extensas de números 0 y 1.

Para facilitar el trabajo, los primeros operadores de computadoras decidieron hacer un traductor para reemplazar los 0 y 1 por palabras o abstracción de palabras y letras provenientes del inglés; éste se conoce como lenguaje ensamblador. Por ejemplo, para sumar se usa la letra A de la palabra inglesa add (sumar). El lenguaje ensamblador sigue la misma estructura del lenguaje máquina, pero las letras y palabras son más fáciles de recordar y entender que los números. La necesidad de recordar secuencias de programación para las acciones usuales llevó a denominarlas con nombres fáciles de memorizar y asociar: ADD (sumar), SUB (restar), MUL (multiplicar), CALL (ejecutar subrutina), etc. A esta secuencia de posiciones se le denominó "instrucciones", y a este conjunto de instrucciones se le llamó lenguaje ensamblador. Posteriormente aparecieron diferentes lenguajes de programación, los cuales reciben su denominación porque tienen una estructura sintáctica similar a los lenguajes escritos por los humanos, denominados también lenguajes de alto nivel.

El primer manual para el lenguaje Fortran apareció en octubre de 1956, con el primer compilador Fortran entregado en abril de 1957. Esto era un compilador optimizado, porque los clientes eran reacios a usar un lenguaje de alto nivel a menos que su compilador pudiera generar código cuyo desempeño fuera comparable al de un código hecho a mano en lenguaje ensamblador.

En 1960, se creó COBOL, uno de los lenguajes usados aún en 2010 en informática de gestión.

A medida que la complejidad de las tareas que realizaban las computadoras aumentaba, se hizo necesario disponer de un método más eficiente para programarlas. Entonces, se crearon los lenguajes de alto nivel, como lo fue BASIC en las versiones introducidas en los microordenadores de la década de 1980. Mientras que una tarea tan sencilla como sumar dos números puede necesitar varias instrucciones en lenguaje ensamblador, en un lenguaje de alto nivel bastará una sola sentencia.

Elementos

Todos los lenguajes de programación tienen algunos elementos de formación primitivos para la descripción de los datos y de los procesos o transformaciones aplicadas a estos datos (tal como la suma de dos números o la selección de un elemento que forma parte de una colección). Estos elementos primitivos son definidos por reglas sintácticas y semánticas que describen su estructura y significado respectivamente.

Sintaxis

A la forma visible de un lenguaje de programación se le conoce como sintaxis. La mayoría de los lenguajes de programación son puramente textuales, es decir, utilizan secuencias de texto que incluyen palabras, números y puntuación, de manera similar a los lenguajes naturales escritos. Por otra parte, hay algunos lenguajes de programación que son más gráficos en su naturaleza, utilizando relaciones visuales entre símbolos para especificar un programa.

La sintaxis de un lenguaje de programación describe las combinaciones posibles de los símbolos que forman un programa sintácticamente correcto. El significado que se le da a una combinación de símbolos es manejado por su semántica (ya sea formal o como parte del código duro de la referencia de implementación).

Dado que la mayoría de los lenguajes son textuales, este artículo trata de la sintaxis textual.

La sintaxis de los lenguajes de programación es definida generalmente utilizando una combinación de expresiones regulares (para la estructura léxica) y la Notación de Backus-Naur (para la estructura gramática). Este es un ejemplo de una gramática simple, tomada de Lisp:

No todos los programas sintácticamente correctos son semánticamente correctos. Muchos programas sintácticamente correctos tienen inconsistencias con las reglas del lenguaje; y pueden (dependiendo de la especificación del lenguaje y la solidez de la implementación) resultar en un error de traducción o ejecución. En algunos casos, tales programas pueden exhibir un comportamiento indefinido. Además, incluso cuando un programa está bien definido dentro de un lenguaje, todavía puede tener un significado que no es el que la persona que lo escribió estaba tratando de construir.

Usando el lenguaje natural, por ejemplo, puede no ser posible asignarle significado a una oración gramaticalmente válida o la oración puede ser falsa:

“Las ideas verdes y descoloridas duermen furiosamente” es una oración bien formada gramaticalmente pero no tiene significado comúnmente aceptado.

“Juan es un soltero casado” también está bien formada gramaticalmente pero expresa un significado que no puede ser verdadero.

http://upload.wikimedia.org/wikipedia/commons/e/e1/Python_add5_syntax.svg

Paradigmas

Los programas se pueden clasificar por el paradigma del lenguaje que se use para producirlos. Los principales paradigmas son: imperativos, declarativos y orientación a objetos.

Los programas que usan un lenguaje imperativo especifican un algoritmo, usan declaraciones, expresiones y sentencias.³ Una declaración asocia un nombre de variable con un tipo de dato, por ejemplo: `var x: integer;`. Una expresión contiene un valor, por ejemplo: `2 + 2` contiene el valor 4. Finalmente, una sentencia debe asignar una expresión a una variable o usar el valor de una variable para alterar el flujo de un programa, por ejemplo: `x := 2 + 2;` `if x == 4 then haz_algo();`. Una crítica común en los lenguajes imperativos es el efecto de las sentencias de asignación sobre una clase de variables llamadas “no locales”.⁴

Implementación

A continuación, encontrará una breve lista de los lenguajes de programación actuales:

Lenguaje Principal área de aplicación Compilado/interpretado ADA Tiempo real
Lenguaje compilado BASIC Programación para fines educativos Lenguaje
interpretado C Programación de sistema Lenguaje compilado C++ Programación
de sistema orientado a objeto Lenguaje compilado Cobol Administración Lenguaje
compilado Fortran Cálculo Lenguaje compilado Java Programación orientada a
Internet Lenguaje intermediario MATLAB Cálculos matemáticos Lenguaje
interpretado Cálculos matemáticos Cálculos matemáticos Lenguaje interpretado
LISP Inteligencia artificial Lenguaje intermediario Pascal Educación Lenguaje
compilado PHP Desarrollo de sitios web dinámicos Lenguaje interpretado
Inteligencia artificial Inteligencia artificial Lenguaje interpretado Perl Procesamiento
de cadenas de caracteres Lenguaje interpretado

La implementación de un lenguaje es la que provee una manera de que se ejecute un programa para una determinada combinación de software y hardware. Existen básicamente dos maneras de implementar un lenguaje: compilación e interpretación.

Compilación: es el proceso que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz interpretar. Los programas traductores que pueden realizar esta operación se llaman compiladores. Éstos, como los programas ensambladores avanzados, pueden generar muchas líneas de código de máquina por cada proposición del programa fuente. Interpretación: es una asignación de significados a las fórmulas bien formadas de un lenguaje formal. Como los lenguajes formales pueden definirse en términos puramente sintácticos, sus fórmulas bien formadas pueden no ser más que cadenas de símbolos sin ningún significado. Una interpretación otorga significado a esas fórmulas.

Se puede también utilizar una alternativa para traducir lenguajes de alto nivel. En lugar de traducir el programa fuente y grabar en forma permanente el código objeto que se produce durante la compilación para utilizarlo en una ejecución futura, el programador sólo carga el programa fuente en la computadora junto con los datos que se van a procesar. A continuación, un programa intérprete, almacenado en el sistema operativo del disco, o incluido de manera permanente dentro de la máquina, convierte cada proposición del programa fuente en lenguaje de máquina conforme vaya siendo necesario durante el procesamiento de los datos.

La siguiente vez que se utilice una instrucción, se la deberá interpretar otra vez y traducir a lenguaje máquina. Por ejemplo, durante el procesamiento repetitivo de

los pasos de un ciclo o bucle, cada instrucción del bucle tendrá que volver a ser interpretada en cada ejecución repetida del ciclo, lo cual hace que el programa sea más lento en tiempo de ejecución (porque se va revisando el código en tiempo de ejecución) pero más rápido en tiempo de diseño (porque no se tiene que estar compilando a cada momento el código completo). El intérprete elimina la necesidad de realizar una compilación después de cada modificación del programa cuando se quiere agregar funciones o corregir errores; pero es obvio que un programa objeto compilado con antelación deberá ejecutarse con mucha mayor rapidez que uno que se debe interpretar a cada paso durante una ejecución del código.

La mayoría de lenguajes de alto nivel permiten la programación multipropósito, sin embargo, muchos de ellos fueron diseñados para permitir programación dedicada, como lo fue el Pascal con las matemáticas en su comienzo. También se han implementado lenguajes educativos infantiles como Logo que mediante una serie de simples instrucciones. En el ámbito de infraestructura de Internet, cabe destacar a Perl con un poderoso sistema de procesamiento de texto y una enorme colección de módulos.

2.4. Herramientas productivas (ofimática).

La Ofimática es un conjunto de técnicas informáticas las cuales nos permiten ahorrar tiempo, espacio y mejorar las tareas de en general una oficina o lugar donde se hagan ese tipo de trabajos.

Esta denominación llegó en los años setentas con la sustitución de la máquina de escribir por la computadora y le siguió el reemplazo de una capturista de datos por la simple y eficiente copiadora.

La ofimática es tan común pudiéndose decir que nacimos con ella y por ello no nos ponemos a pensar lo difícil que sería la vida sin sus privilegios, ahora cualquier estudiante y hasta un ama de casa goza de sus frutos y avances continuos pues esta presente hasta en un simple fax que ayudo a las oficinas y hogares a ahorrar tiempos de transportación por un simple clic que transmite datos a grandes distancias, pero también tu e-mail cumple con esa función.

Si de este tema hablamos sería una grosería dejar de lado la tan conocida y útil paquetería de Microsoft Office, presente en la mayoría de los equipos de cómputo tanto de un hogar como oficina así como de un empresario o de un simple estudiante de primaria.

Office cuenta con variadas herramientas, entre las más conocidas están: Microsoft Word, Excel, [Power Point?](#) y Publisher, la primera es la mas popular y necesaria, es una procesador de texto del cual somos tan dependientes que cualquier cosa que redactemos en su 99% esta hecho en esta herramienta, Excel por su parte es la herramienta líder en cálculos, utilizada desde una pequeña tienda para realizar los gastos y ganancias del día o semana hasta en una inmensa corporación la cual ocupa tener listo de inmediato el reporte de ventas de el año completo, Power Point no se queda atrás en el sector pues después de tantas herramientas que lo quieren derribar de la cima como líder en presentaciones multimedia se mantiene vigente y es común que hasta los alumnos de primaria ya la usen para enriquecer la clase y por ultimo Publisher una la cual es menos usada que las mencionadas anteriormente pero que es de gran utilidad a la hora de querer crear publicidad de manera rápida o instantánea ya que cuenta con una gran cantidad de plantillas las cuales solo ocupan que les modifiques la información textual y quedan listas.

La Ofimática es de las herramientas mas útiles y es necesario saber que día a día estos instrumentos van mejorando y que debemos esmerarnos en conocer la mayoría de ellas pues su variedad de servicios son amplias y no sabemos cuando podremos necesitarlas.

2.5. Software propietario y libre.

Software propietario, también conocido como software privativo o privado, se refiere a cualquier tipo de software, o programa informático, que prohíbe o pone restricciones al usuario en su uso, redistribución o modificaciones. El dueño del software, generalmente el creador de dicho software, posee los derechos de autor, lo que le permite restringir su uso, la inspección de su código fuente, modificaciones del código fuente y su redistribución, sea el software original o modificado. La Fundación para el Software Libre, fundada por Richard M. Stallman, es una organización cuyo objetivo es crear y promover el uso del software libre. Ésta emplea los términos de software no libre y software privativo, para referirse al software que no es libre según su clasificación. De acuerdo a Richard Stallman, es inmoral instalar y utilizar software propietario, ya que se crea un dilema ético al querer compartirlo, y se puede ser perseguido por la ley. Una persona que utiliza software propietario, al no tener acceso al código fuente, no puede saber si existe alguna amenaza contra su información o equipo, y el al investigarlo o intentar modificarlo también puede ser perseguido por la ley.

Software libre, es el término con el que se clasifica a todo tipo de software que otorgue al usuario el permiso para utilizar, copiar y distribuir, con o sin modificaciones, de manera gratuita o con precio. El término libre, del inglés “free”, no se refiere al precio, sino a la libertad. Para poder tener todas esas libertades, el código fuente del software debe ser accesible para el usuario, por lo que es una condición necesaria para poder ser software libre.

De acuerdo a la Fundación para el Software Libre, un software se categoriza como libre cuando garantiza las cuatro libertades, numeradas de 0 a 3 como en el contexto de la programación:

0. El usuario tiene la libertad de utilizar el programa para cualquier propósito.
1. El usuario es libre de estudiar el funcionamiento del programa y modificarlo de acuerdo a sus necesidades.
2. El usuario es libre de distribuir copias del programa.
3. El usuario tiene la libertad de mejorar el programa y hacer esas mejoras públicas para los demás, de manera que toda la comunidad pueda beneficiarse.

Las únicas condiciones o restricciones que el software libre puede tener es la regla conocida como copyleft, que se refiere a que una copia o modificación de un software mantenga los derechos del autor originales, es decir, si se hace una modificación a un software libre, uno no puede aplicarle restricciones que nieguen las libertades del usuario, porque entonces ya no sería software libre.