

INTERNA. Tecnologías actuales de ordenadores: de los dispositivos móviles a los superordenadores y arquitecturas escalables y de altas prestaciones. Computación en la nube. Base tecnológica. Componentes, funcionalidades y capacidades.

Tecnologías actuales de ordenadores: de los dispositivos móviles a los superordenadores y arquitecturas escalables (grid, cluster, MPP, SMP, arquitecturas multinúcleo y otros).

Tecnologías actuales de ordenadores

El acrónimo **TIC “Tecnologías de la Información y de la Comunicación”** agrupa elementos y técnicas utilizadas en el tratamiento y transmisión de la información, principalmente de informática, internet y telecomunicaciones.

Podemos decir que las TIC son herramientas informáticas que almacenan, procesan y presentan información de formas muy diferentes. Estas Tecnologías incluyen ordenadores, internet, tecnologías de radiodifusión (radio y televisión) y telefonía.

Dispositivos móviles

Generalmente, los dispositivos móviles los definimos como aquellos micro-ordenadores que son lo suficientemente ligeros como para ser transportados por una persona, y que disponen de la capacidad de batería suficiente como para poder funcionar de forma autónoma.

Es importante destacar que los ordenadores portátiles no se consideran dispositivos móviles debido a que consumen más batería y suelen ser más pesados.

A grandes rasgos se pueden dividir los dispositivos móviles en tres amplios grupos que son: teléfonos, PDAs y consolas.

Teléfonos

Son los más pequeños del grupo, y por tanto los más ligeros y más transportables. Su función primordial era clara históricamente, lo que hace un teléfono cualquiera: recibir y realizar llamadas; aunque desde hace ya tiempo es impensable concebir un teléfono móvil que solamente haga eso. Funcionalidades propias de ordenadores, o de dispositivos de otro tipo, como la grabación y edición de vídeo, realización de fotografías, lectura de documentos, localización en mapas, navegación por internet, y muchas cosas más.

PDAs (Personal Digital Assistant) (Asistente Personal Digital)

También son conocidos como ordenadores de mano u organizadores electrónicos.

Su funcionalidad principal es servir como organizadores, con agenda, calendario, gestión de contactos, y posteriormente han ido creciendo, de forma que actualmente sirven tanto

como aparatos en los que leer un libro como en los que encontrarse en un mapa. La línea que los separa de los teléfonos es cada vez más difusa.

Consolas

En realidad esta categoría debería llamarse “dispositivos orientados a jugar”, porque son más que simples consolas. Dos de los ejemplos en el mercado son la Sony PlayStation Portable (PSP) y la Nintendo DS, que no sólo sirven para jugar, sino que integran algunas de las funcionalidades típicas de una PDA, como reproducción de archivos multimedia, integración con agenda y calendario, o navegador de internet.

Dispositivos fijos. Computadoras.

Estos dispositivos requieren de una ubicación física permanente o prácticamente permanente ya que necesitan de una toma de corriente, a no ser que lleven una batería, como los ordenadores portátiles. Aun así, la autonomía de los portátiles es limitada lo que les convierte en dispositivos fijos. Además de su dependencia de la corriente eléctrica, una computadora tiene una envergadura y peso que les convierte en incómodos de trasladar.

Microordenadores

Ordenador portátil

Llamamos ordenador portátil al ordenador que puede funcionar autónomamente sin necesidad de tenerlo enchufado a la red eléctrica, y el cual puede ser trasladado de un lugar a otro con facilidad.

Se distinguen tres tipos de portátiles: Deskbook, Desktop y Mobile.

- **Deskbook** : Son portátiles de bajo precio. En realidad son utilizados como ordenadores de sobremesa pero que se pueden transportar. Son grandes, pesados y no traen batería.
- **Desktop** : Estos son los portátiles por excelencia. Son iguales que los deskbook pero estos sí que tienen batería incorporada, lo que hace que se puedan transportar con mayor facilidad. Son más ligeros y tienen prácticamente las mismas prestaciones que los de sobremesa.
- **Mobile** : Pertenece a la última generación de ordenadores portátiles. Son los más ligeros, se calientan mucho menos y su batería tienen una autonomía mayor. Todo esto hace que sean más manejables. Además, hacen menos ruido. Tienen el mismo rendimiento y prestaciones que un desktop.

Ordenador de sobremesa

El conocido como PC (Personal Computer), Ordenador personal, etc. Es el computador por excelencia.

Están fabricados para el uso de una persona, son de un tamaño medio... y cumple multitud de funcionalidades.

Fueron concebidos para usuarios domésticos, pero su potencial y sus programas los han implantado en el ámbito laboral y profesional.

Estaciones de Trabajo o Workstation

Son equipos de gran potencia. Son sofisticados y especialmente diseñados para niveles de alto rendimiento. Suelen ser utilizados para ingeniería, cálculos técnicos, diseño, gráfico, diseño de software, ...

Estas computadoras de gama alta están equipadas con funciones adicionales como por ejemplo, procesadores más rápidos, monitores de alta resolución, tarjetas gráficas potentes, y aplicaciones integradas que vienen instaladas por defecto.

Servidor

Es un ordenador que ha sido optimizado para proveer de servicios a otros ordenadores sobre una red local o de internet. Usualmente disponen de procesadores de alta potencia, mucha memoria y varios discos duros de gran tamaño.

Miniorodenadores

Son ordenadores de tamaño medio, con unas capacidades intermedias entre ordenadores personales y los grandes ordenadores.

Pueden ser utilizados por varios usuarios al mismo tiempo y disponen de mayores recursos que los microordenadores.

Cuentan con una mayor capacidad de proceso, mayor memoria, periféricos más sofisticados y posibilidad de conectar más de un puesto de trabajo. Son también conocidos como ordenadores departamentales.

Ordenadores grandes o Mainframes

Son ordenadores de gran capacidad, tanto de procesamiento como de almacenamiento, comunicaciones, etc. Son capaces de gestionar múltiples bases de datos, procesar miles de transacciones al minuto procedentes de miles de terminales a la vez.

Es frecuente encontrar varios procesadores trabajando en paralelo, lo cual requiere sistemas más complejos y equipos especialistas.

SuperOrdenadores

Son ordenadores de gran potencia y elevadísimas prestaciones. Se utilizan principalmente para cálculos científicos que necesitan una gran capacidad de proceso. Es capaz de realizar miles de millones de operaciones por segundo.

Arquitecturas Escalables

Una arquitectura escalable es aquella que tiene la capacidad de incrementar el rendimiento sin que tenga que rediseñarse y simplemente aprovecha el hardware adicional que se le disponga.

Generalmente podemos definir la escalabilidad como la capacidad que tiene un sistema informático de modificar su configuración o su tamaño, para ajustarse a los cambios.

- **Dimensiones** . La escalabilidad de un sistema se puede medir en distintas dimensiones.
- **Escalabilidad de carga** . Esto se hace más fácil mediante un sistema distribuido, podemos ampliar y reducir los recursos con mayor facilidad para adecuar las cargas ya sean pesadas o ligeras según sea necesario.
- **Escalabilidad geográfica** . Un sistema es escalable geográficamente cuando su uso y sus ventajas se conservan sin que afecte la distancia de los usuarios.
- **Escalabilidad administrativa** . Este debe de manejarse con facilidad sin importar las organizaciones que necesiten compartir un solo sistema distribuido.

- **Escalabilidad vertical** . También se dice escala hacia arriba, quiere decir que en un solo nodo del sistema es donde se han agregado más recursos. Ejemplo, añadir memoria a un disco duro de una computadora.
- **Escalabilidad horizontal** . Quiere decir que se agregan más nodos a un sistema. Ejemplo, agregar una nueva computadora a un programa de aplicación para espejo.

MPP (Massive Parallel Processing)

Un computador masivamente paralelo es un sistema de Memoria Distribuida que consiste en muchos nodos individuales, cada uno de los cuales es esencialmente un ordenador independiente en sí mismo, y consiste en al menos un procesador, su propia memoria y un enlace a la red que lo une con el resto de nodos. El término masivo supone cientos o miles de nodos. Los nodos se comunican por paso de mensajes, usando estándares como MPI.

MPI (Message Passing Interface)

Una API que permite a procesos comunicarse con otros enviando y recibiendo mensajes. Es un estándar de facto para programas paralelos ejecutándose en clusters de ordenadores y supercomputadores.

SMP (Symmetric Multiprocessing)

Una arquitectura multiprocesador donde dos o más procesadores idénticos se conectan a una Memoria Principal compartida y controlada por una sola instancia del sistema operativo. Está claro que está hablando de las máquinas PC de hoy día, donde cada chip es de doble o cuádruple núcleo, hay una memoria principal y hay un sistema operativo en ejecución.

Grid

Puede ser visto como un sistema distribuido. Lo que lo distingue de un clúster es que el grid tiende a estar más débilmente acoplado, heterogéneo y geográficamente disperso. Aunque se puede dedicar el grid a una aplicación específica, es más común que un solo grid se use para una variedad de propósitos diferentes. Normalmente los recursos no se administran de forma centralizada, se usan estándares abiertos y se obtiene calidad de servicio.

Clúster

Un grupo de ordenadores enlazados, trabajando juntos, colaborando estrechamente, formando uno solo en muchos aspectos. Normalmente están conectados por redes de área local rápidas. El servicio que se suele dar es mejorar la disponibilidad y el rendimiento. Distinguimos estas categorías:

- Clústers de Alta Disponibilidad. Linux-HA es un proyecto que sirve para esto.
- Clústers de Balanceo de Carga.
- Clústers de computación. Donde empezamos a colisionar con el concepto de grid. Se usan para simulaciones, modelado, predicción del tiempo, etc.

Multinúcleo

Los núcleos pueden o no compartir caché y pueden implementar métodos de comunicación:

- paso de mensajes
- memoria compartida
- comunicación internúcleo

Algunas topologías son:

- bus
- anillo
- malla bidimensional
- crossbar

Ley de Amdahl

Encontrar el máximo nivel de mejora para un sistema completo cuando solo una parte es mejorada. Se usa normalmente en computación paralela para predecir la mejora máxima teórica utilizando múltiples procesadores o núcleos.

Por ejemplo, si un programa necesita 20 horas en un solo núcleo y una porción particular de 1 hora no puede ser paralelizada, pero las 19 horas restantes sí pueden serlo, entonces independientemente de cuantos procesadores utilicemos el mínimo tiempo de ejecución posible no puede ser menor a esa hora.

INTERNA. Conceptos de sistemas operativos: Características, evolución y tendencias. Estructuras, componentes y funciones. Sistemas operativos multiprocesador.

Conceptos de Sistemas Operativos

Introducción

Un Sistema Operativo es un programa que administra el hardware de una computadora. También proporciona las bases para los programas de aplicación, y actúa como intermediario entre el usuario y el hardware. Estas tareas pueden ser llevadas a cabo de varias formas, lo que permite que algunos Sistemas Operativos se diseñen para ser prácticos, otros eficientes y otros para ser ambas cosas.

¿Qué hace un Sistema Operativo?

Un sistema informático puede dividirse en cuatro componentes: el hardware, el Sistema Operativo, los programas de aplicación y los usuario. El Sistema Operativo controla y coordina el uso del hardware entre los diversos programas de aplicación por parte de los distintos usuarios.

También podemos ver un sistema informático como hardware, software y datos. El Sistema Operativo proporciona los medios para hacer un uso adecuado de estos recursos durante el funcionamiento del sistema informático.

Definición Sistema Operativo

Un Sistema Operativo es un programa, o conjunto de programas eficiente y productivo en el uso de un computador (hardware), permitiendo la ejecución de aplicaciones de usuario. Es el intermediario entre las aplicaciones de usuario y el hardware.

Metas:

- Brindar un ambiente de realización y ejecución de aplicaciones.
- Proveer un entorno sin interferencias a cada usuario (interferencia: lo que un usuario modifica en su entorno, no interfiera ni modifique lo de otro usuario).
- Administrar de forma equitativa los recursos (hardware y software).
- Hacerlo de la forma más amigable e intuitiva posible.

Todas las aplicaciones de usuario requieren un conjunto común de operaciones que son incorporadas al Sistema Operativo.

Tareas principales:

- Implementar diferentes entornos para diferentes usos (interfaz gráfica, shells, tipo web, etc).
- Proveer una o más interfaces con el usuario.
- Proveer a las aplicaciones un conjunto de servicios (a través de los “system services”).
- Eficiencia y equidad en la administración de recursos.

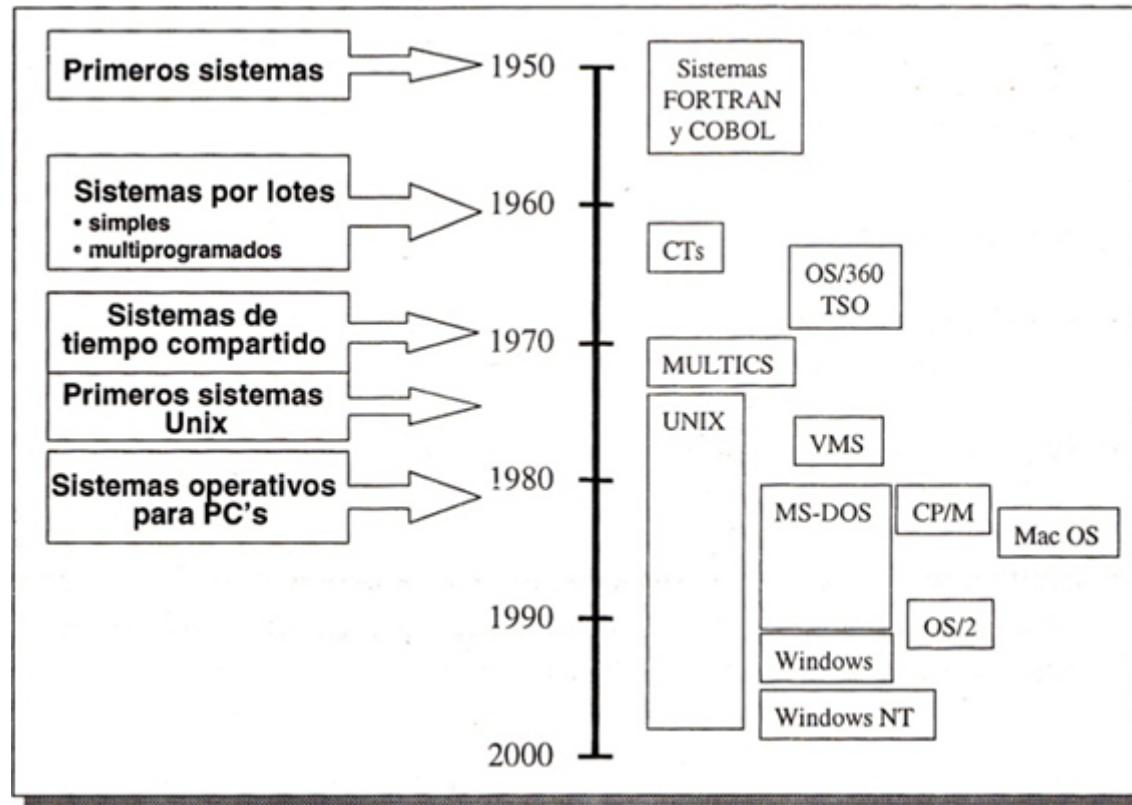
Se puede decir que el Sistema Operativo es un:

- Administrador de recursos. Sus tareas consisten en administrar los recursos disponibles y decidir como asignar estos recursos según los pedidos y asignaciones que tenga.
- Programa de control. Controla la ejecución de los programas para la prevención de errores y mal uso del sistema.

Frecuentemente la porción residente del propio Sistema Operativo se denomina *núcleo del sistema (Kernel)*.

Evolución de los Sistemas Operativos

La informática tal y como se le conoce hoy día, surgió a raíz de la II Guerra Mundial, en la década de los 40. En esos años no existía siquiera el concepto de “Sistema Operativo” y los programadores interactuaban directamente con el hardware de las computadoras trabajando en lenguaje máquina (es decir, en binario, programando únicamente 0s y 1s).



Sistemas Batch o por Lotes (Años 70 y comienzo de los 80)

En las primeras épocas los sistemas eran grandes y costosos. Constaban de una entrada de trabajos y una salida impresa, por lo cual la interacción con el usuario era prácticamente nula. Las principales características eran que el sistema soportaba un único trabajo a la vez, y que las tareas relacionadas se agrupaban en conjuntos o lotes, para su procesamiento más eficiente.

A comienzo de los 80, utilizando las técnicas de Spooling (proceso mediante el cual la computadora introduce trabajos en un buffer, de manera que un dispositivo pueda acceder a ellos cuando esté listo) y multiprogramación (ejecución de múltiples tareas compartiendo recursos) se pudo comenzar a desarrollar técnicas de planificación de despacho.

Esta técnica consistía en seleccionar un lote de trabajos que estaban en memoria secundaria para cargarlos en memoria principal. Luego, el SO seleccionaba uno de ellos para ejecutar, y si este debía esperar por alguna tarea (por ejemplo ejecución de E/S) el sistema elegía otro del lote para utilizar el procesador. Esto incrementó el uso del procesador.

Sistemas de Tiempo Compartido (Finales de los 80)

Estos sistemas eran multiusuarios. Ejecutaban programas de forma concurrente con una elevada tasa de procesos, de forma tal que permitía a los usuarios interactuar directamente con el sistema como si fuera un único usuario.

La necesidad de acceder y actualizar datos de forma concurrente, creó la necesidad de evolucionar el sistema de archivos a uno multiusuario, incorporando técnicas de protección de accesos.

Sistemas de Computadores Personales (Años 80)

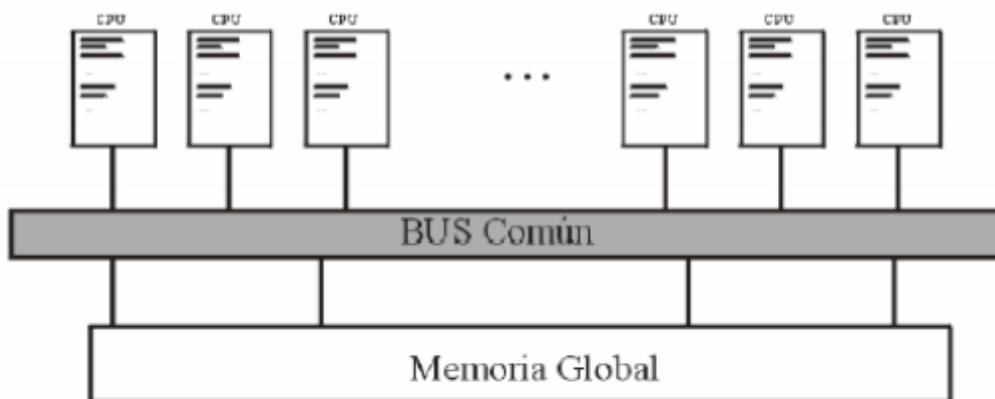
Con costos de hardware decrecientes, fue posible el diseño y uso de computadores personales. Los Sistemas fueron diseñados en base a que serían utilizados por un único usuario, y todo el énfasis en el desarrollo estuvo en mejorar la interacción con el usuario. Se desarrolló la interfaz de ventanas que conocemos hoy.

Sistemas Paralelos (Comienzo de los 90)

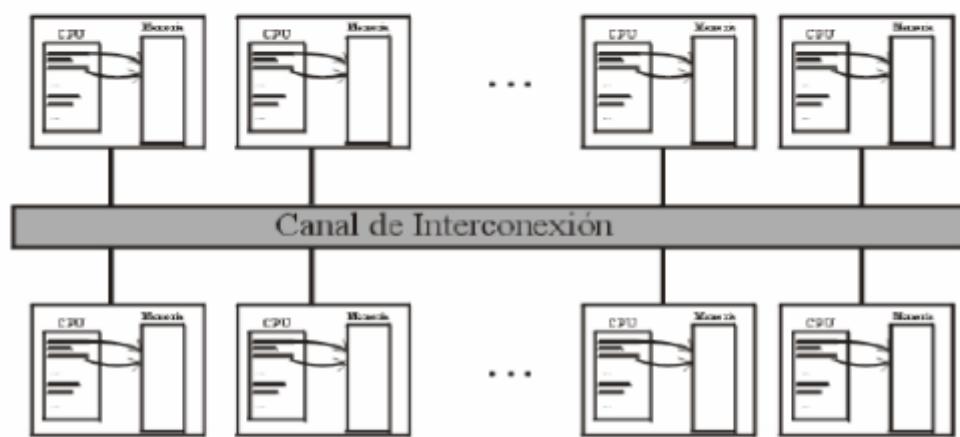
Son sistemas donde se dispone de más de un procesador, permitiendo ejecución simultánea y sincronizada de procesos. Se clasifican en:

- Altamente integrados: “tightly coupled”. Son sistemas en donde los canales de interconexión son de alta velocidad (bus común o memoria compartida).
- Poco integrados: “closely coupled”. Son sistemas en donde los canales de interconexión son de baja velocidad (sistemas en red).

Arquitecturas de memoria compartida



Arquitecturas de memoria distribuida



Veamos ahora otra clasificación de los Sistemas Paralelos:

- **Asimétricos** : se designa una CPU (master) para ejecutar el código del núcleo, para no lidiar con la concurrencia, los demás (slaves) ejecutarán lo que éste les designe.
- **Simétricos** : todos los procesadores son considerados iguales, el código del núcleo se dispone en memoria común y es ejecutado por cualquier procesador.

Y otra clasificación más:

- **UMA (Uniform Memory Access)** : cada CPU accede a cualquier lugar de la memoria en el mismo tiempo.
- **NUMA (Non-Uniform Memory Access)** : las CPU tienen áreas de memoria a las que acceden más rápido que el resto.

Veamos ahora una clasificación de Arquitecturas (Taxonomía de Flynn):

- **SISD (Single Instruction, Single Data)** : Arquitectura secuencial, no hay paralelismo, son arquitecturas monoprocesadores.
- **SIMD (Single Instruction, Multiple Data)** : Son sistemas que ejecutan la misma instrucción sobre un conjunto de datos (Arquitectura Vectorial).
- **MISD (Multiple Instruction, Single Data)** : Paralelismo redundante.
- **MIMD (Multiple Instruction, Multiple Data)** : Varios procesadores autónomos que ejecutan en forma simultanea varias instrucciones sobre datos diferentes
 - **memoria compartida** : escalan poco, acceso a memoria es cuello de botella
 - **memoria distribuida** : escalan a miles de procesadores, conectados en una red de alta velocidad.

Como ejemplo de sistemas computacionales que utilizan sistemas paralelos tenemos los clusters. Estos son sistemas en la cual participan varias computadoras. Los clusters brindan alta disponibilidad (mantiene una serie de servicios, a pesar de posibles fallos), alto rendimiento (en cuanto a capacidad de cálculo) y balance de carga (técnica usada para compartir el trabajo a realizar entre varios procesos, ordenadores, etc)

Se clasifican en:

- Simétricos: todos los nodos ejecutan tareas y asumen las de otros ante fallos.
- Asimétricos: nodos primarios ejecutan tareas y nodos secundarios esperan fallos.

Sistemas de Tiempo Real

Son sistemas en los cuales todo resultado debe producirse en un cierto tiempo. De lo contrario se considera que el sistema ha fallado.

Estructura de los Sistemas Operativos

Componentes de un Sistema Operativo

- Gestión de procesos
- Gestión de memoria
- Gestión de Entrada/Salida
- Administración de Almacenamiento Secundario
- Gestión de Archivos
- Sistema de Protección

Gestión de Procesos

Un proceso es un programa en memoria + CPU + acceso a dispositivos + otros recursos. Un proceso necesita de ciertos recursos (CPU, memoria, archivos, dispositivos E/S, etc) para realizar su tarea.

Podemos ver entonces que un proceso es una entidad activa, mientras que un programa es una entidad pasiva.

Sabiendo entonces qué es un proceso, podemos decir entonces que el sistema operativo es el encargado de su administración. Es el encargado de proveer servicios para que cada proceso pueda realizar su tarea. Entre los servicios se encuentran:

- Crear y destruir procesos
- Suspender y reanudar procesos
- Proveer mecanismos para la sincronización y comunicación entre procesos
- Proveer mecanismos para prevenir dead-locks o lograr salir de ellos

Gestión de Memoria

La memoria es un área de almacenamiento común a los procesadores y dispositivos, donde se almacenan programas, datos, etc. El sistema deberá administrar el lugar libre y ocupado, y será el encargado de las siguientes tareas:

- Mantener qué partes de la memoria están siendo usadas, y por quién.
- Decidir qué procesos serán cargados a memoria cuando exista espacio de memoria disponible, pero no suficiente para todos los procesos que deseamos.
- Asignar y quitar espacio de memoria según sea necesario.

Gestión de Entrada/Salida

El sistema operativo deberá ocultar las características específicas de cada dispositivo y ofrecer servicios comunes a todos. Estos servicios serán, entre otros:

- Montaje y desmontaje de dispositivos
- Una interfaz entre el cliente y el sistema operativo para los device drivers
- Técnicas de caché, buffering y spooling
- Device drivers específicos

Administración de Almacenamiento Secundario

Dado que la memoria RAM es volátil y pequeña para todos los datos y programas que se precisan guardar, se utilizan discos para guardar la mayoría de la información. El sistema operativo será el responsable de:

- Administrar el espacio libre
- Asignar la información a un determinado lugar
- Algoritmos de planificación de disco (estos algoritmos deciden quien utiliza un determinado recurso del disco cuando hay competencia por él)

Gestión de Archivos

Proporciona una vista uniforme de todas las formas de almacenamiento, implementando el concepto de archivo como una colección de bytes. El Sistema Operativo deberá proveer métodos para:

- Abrir, cerrar y crear archivos
- Leer y escribir archivos
- Organización de directorios

Sistema de Protección

Por Protección nos referimos a los mecanismos por los que se controla el acceso de los procesos a los recursos.

En un sistema multiusuario donde se ejecutan procesos de forma concurrente se deben tomar medidas que garanticen la ausencia de interferencia entre ellos. Estas medidas deben incorporar la posibilidad de definir reglas de acceso, entre otras cosas.

Servicios del Sistema Operativo

El sistema brindará un entorno de ejecución de programas donde se dispondrá de un conjunto de servicios. Los servicios principales serán:

- **Ejecución de programas** : el SO deberá ser capaz de cargar un programa a memoria y ejecutarlo. El programa deberá poder finalizar, de forma normal o anormal.
- **Operaciones de E/S** : el SO deberá proveer un mecanismo de acceso ya que por eficiencia y protección los usuarios no accederán directamente al dispositivo.
- **Manipulación del Sistema de Archivos** : se deberá tener acceso al sistema de archivos y poder, como mínimo, leer, escribir, borrar y crear.
- **Comunicación entre procesos** : los procesos deberán poder comunicarse, ya sea que estén en el mismo computador o en diferentes.
- **Manipulación de errores** : el sistema deberá tomar decisiones adecuadas ante eventuales errores que ocurran, como fallo de un dispositivo de memoria, fallo en un programa, etc.

Estructura del Sistema

La estructura interna de los sistemas operativos pueden ser muy diferentes, ya que se debe tener en cuenta las metas de los usuarios (fácil, uso, confiable, rápido, etc) y las del sistema (fácil de diseñar, implementar y mantener, eficiente, etc).

Veremos 3 posibles diseños del sistema: sistema monolítico, sistema en capas, sistema con micronúcleo.

Sistema Monolítico

Estos sistemas no tienen una estructura definida, sino que son escritos como una colección de procedimientos donde cualquier procedimiento puede invocar a otro.

Ejemplo de estos sistemas pueden ser MS-DOS. Es importante tener en cuenta que ningún sistema es puramente de un tipo.

Sistema en Capas o Niveles

El diseño se organiza en una jerarquía de capas, donde los servicios que brinda una capa son consumidos solamente por la capa superior. La capa 0 es del Hardware y la N es la de los procesos de Usuario.

Nivel Usuario
Muestra al usuario el proceso que se está ejecutando o el que se quiere ejecutar
Nivel Supervisor
Se encarga de realizar la comunicación de cada proceso entre el sistema y el usuario
Nivel Ejecutivo
Sobre este nivel se realiza la administración de la memoria para almacenar los procesos en páginas
Nivel Núcleo
Se encarga de gestionar qué procesos llegan a la CPU para ser ejecutados

Capas/niveles de un SO

Estos sistemas tienen como ventaja que son modulares y la verificación se puede hacer a cada capa por separado (son más mantenibles). Sin embargo el diseño es muy costoso y es menos eficiente que el sistema monolítico ya que pierde tiempo pasando por cada capa.

Sistema con micronúcleo (microkernels)

La idea consiste en tener un núcleo que brinde los servicios mínimos de manejo de procesos, memoria y que provea la comunicación entre procesos. Todos los restantes servicios se construyen como procesos separados del micronúcleo, que ejecutan en modo usuario.

Estos sistemas tienen como ventaja un diseño simple y funcional, que aumenta la portabilidad y la escalabilidad. Para agregar un nuevo servicio no es necesario modificar el núcleo, y es más seguro ya que los servicios corren en modo usuario.

Cliente/Servidor

Los procesos se diferencian en servidores, que proporcionan ciertos servicios y clientes que disponen de esos servicios.

Máquinas Virtuales

Se ejecuta un monitor de máquinas virtuales que proporciona copias virtuales del hardware al resto de procesos. En cada una de las máquinas se ejecuta un SO.

Exokernels

Un programa se ejecuta en modo kernel, asignando los recursos a las máquinas virtuales.

Híbrido

Implica que el núcleo en cuestión usa conceptos de arquitectura o mecanismos tanto del diseño monolítico como del micronúcleo.

Tipos de Sistemas Operativos

Se pueden clasificar los SO en función de:

- **Nº de usuarios**
 - **Monousuario** : solo 1 usuario puede usar los recursos del sistema simultáneamente.
 - **Multiusuario** : varios usuarios pueden usar los recursos del sistema simultáneamente. Por tanto, aunque haya más de un usuario dado de alta en el sistema, si no pueden trabajar de forma simultánea, el SO no es multiusuario.
- **Nº de procesos o tareas**
 - **Monotarea** : solo puede ejecutar 1 tarea a la vez.
 - **Multitarea o multiprogramación** : puede ejecutar varios programas a la vez.
- **Nº de procesadores**
 - **Monoproceso / monoprocesador** : el SO es capaz de gestionar solo 1 procesador, de manera que si tuviese más sería inútil. En estos SO los procesos irán alternando su ocupación en la CPU.
 - **Multiproceso / multiprocesador** : el SO es capaz de gestionar varios procesadores, de modo que puede usarlos simultáneamente para distribuir su carga de trabajo. Estos sistemas trabajan de dos formas:
 - **Asimétrica** : el SO reparte las tareas, que está realizando, entre los procesadores. Determinados procesos los ejecutará siempre un procesador, y el otro procesador sólo se utilizará para realizar procesos de usuario. En este caso, es posible que un procesador esté siempre trabajando y el otro, en ocasiones, sin actividad.
 - **Símétrica** : los procesos son enviados indistintamente a cualquiera de los procesadores disponibles.
- **Tiempo de respuesta** (tiempo que tarda el usuario en obtener los resultados después de iniciar la ejecución de un programa):
 - **Procesamiento por lotes** : el tiempo de respuesta no es importante y suele ser alto. Los procesos se ejecutan secuencialmente unos tras otro. No existe interacción con el usuario. Ejemplo: copias de seguridad.
 - **Tiempo compartido** : el procesador divide su tiempo entre todos los procesos (usando algoritmos de planificación como Round Robin). Ejemplo: sistemas multiusuarios interactivos (los usuarios interactúan con el sistema).
 - **Tiempo real** : en estos SO, los procesos requieren un tiempo de respuesta muy bajo o inmediato. Ejemplos donde esto es especialmente importante: sistema donde el tiempo de respuesta es crucial como sistemas médicos de monitorización de pacientes, sistemas bancarios, tráfico aéreo...

Administración de Memoria

Memoria Principal

Introducción

En sistemas multiprogramados, para sacarle jugo a la multiprogramación, se necesita tener varios procesos cargados en memoria a la vez.

Recordemos, que con respecto a la administración de memoria, el SO es responsable de:

- Mantener qué partes de la memoria están en uso y por quién.
- Decidir qué procesos cargar cuando haya memoria libre.
- Asignar y quitar espacio de memoria según sea necesario.

Preparación de un programa para ejecutar

Los programas son normalmente escritos en lenguajes de alto nivel, y deben pasar por distintas etapas antes de ser ejecutados:

- Compilación (compile): traducción de código fuente a código objeto.
- Ensamblaje (linker): ensamblar varios códigos objeto en un archivo ejecutable. Surge ante la necesidad de modularizar los programas y reutilizar código.
- Carga (load): asigna el archivo ejecutable a la memoria principal del sistema (crea en memoria el espacio necesario para diferentes áreas y las carga con la información).

El tamaño de un proceso en memoria principal está limitado por la cantidad de memoria física que exista. Para aprovechar mejor la memoria, se puede utilizar la carga dinámica, la cual no cargará en memoria principal una rutina hasta que ésta no sea invocada.

La gran ventaja es que las rutinas que no son utilizadas, no son cargadas a memoria física, y por lo tanto no consumen este recurso.

Direcciones Relativas y Absolutas

La mayoría de los SO permiten que un proceso de usuario resida en cualquier parte de la memoria principal. Es así que, aunque el espacio de direcciones comience en el 0000, la primera dirección de usuario no tiene porqué ser 0000. Esta posibilidad afecta a las direcciones que el programa de usuario puede utilizar. En cada una de las etapas que hemos visto para poder ejecutar un programa, las direcciones pueden representarse de diferentes formas.

Las direcciones de un programa fuente son normalmente simbólicas. Al compilar, el compilador se encarga de reasignar estas direcciones simbólicas a direcciones relativas. El cargador, se encargará, a su vez, de reasignar direcciones relativas a direcciones absolutas.

Asociación de direcciones (address binding)

¿En qué momento el SO reasigna las instrucciones y los datos a direcciones de memoria?:

- Tiempo de Compilación: Si sabemos en el momento de la compilación donde va a residir el proceso en memoria, podemos generar código absoluto (con direcciones absolutas). Ahora, si en algún momento deseamos cambiar su ubicación, deberemos recomilar el código.
- Tiempo de Carga: El compilador deberá generar código reubicable (con direcciones relativas), y en este caso se retarda la reasignación a direcciones absolutas hasta el momento de la carga. Si en algún momento deseamos cambiar su ubicación, deberemos solamente volver a cargarlo.
- Tiempo de Ejecución: Si el proceso puede variar su ubicación en memoria durante su ejecución, entonces es necesario retardar su asignación a direcciones absolutas hasta el momento de ejecución. Para que este esquema pueda funcionar, se requiere soporte de hardware.

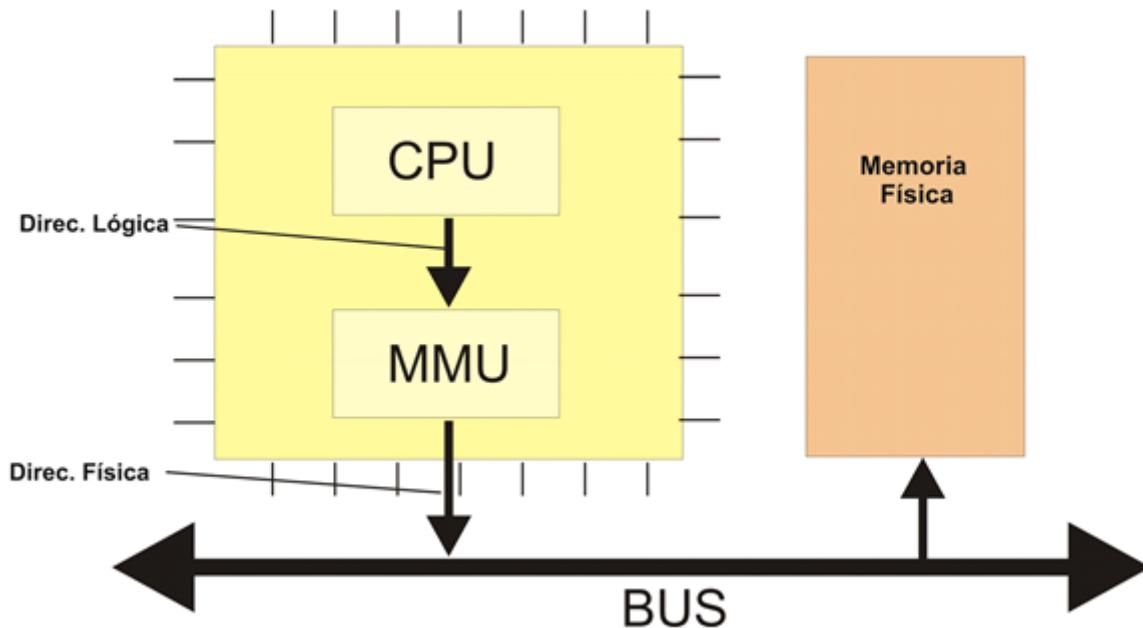
Espacios de direcciones lógico y físico

Una dirección generada por la CPU se denomina normalmente dirección lógica, mientras que una dirección vista por la unidad de memoria se denomina dirección física.

Los métodos de reasignación en tiempo de compilación y de carga generan direcciones físicas y lógicas idénticas; no es el caso para el tiempo de ejecución. En este caso decimos que la dirección lógica es una dirección virtual.

Al conjunto de todas las direcciones lógicas de un programa se le denomina espacio de direcciones lógicas; mientras que al conjunto de todas las direcciones físicas de un programa se le denomina espacio de direcciones físicas.

La correspondencia entre direcciones virtuales y físicas en tiempo de ejecución es establecida por un dispositivo de hardware que se denomina Unidad de Gestión de Memoria (MMU = Memory Management Unit).



Estrategia de asignación o reubicación

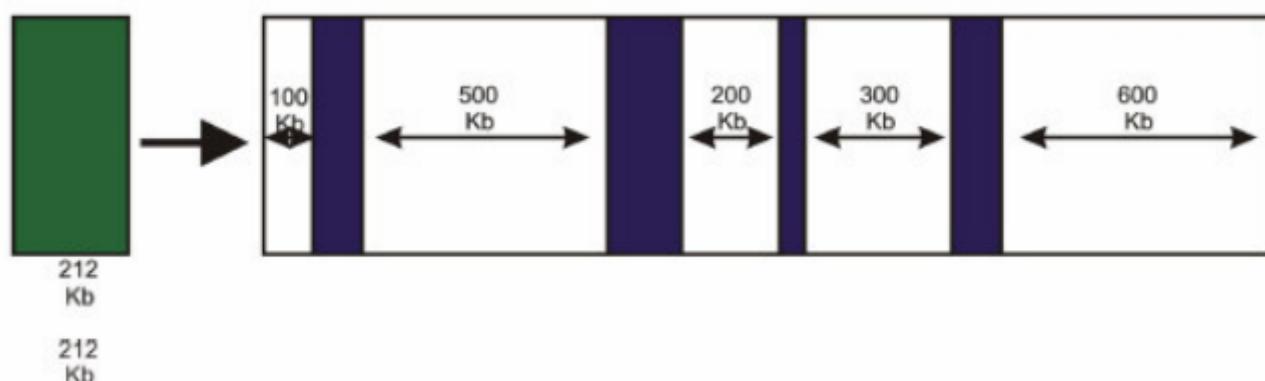
¿Cómo elige el SO en qué porción de memoria colocaremos un proceso? Existen varias estrategias:

- First fit: Asigna el primer “agujero” de memoria libre que satisface la necesidad.
- Best fit: Asigna el mejor “agujero” de memoria libre que exista en la memoria principal.
- Worst fit: Asigna en el “agujero” más grande que exista en la memoria principal.

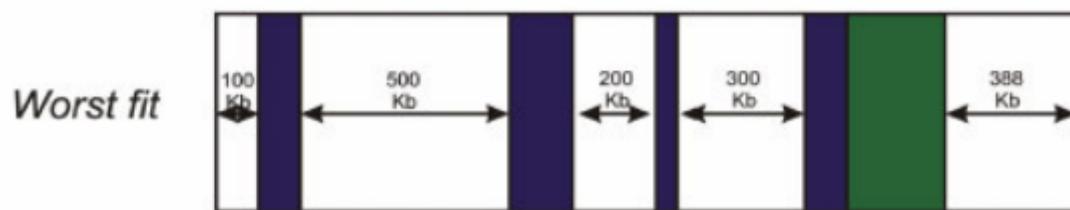
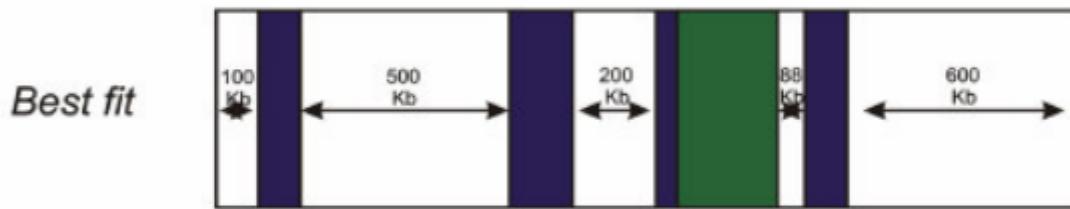
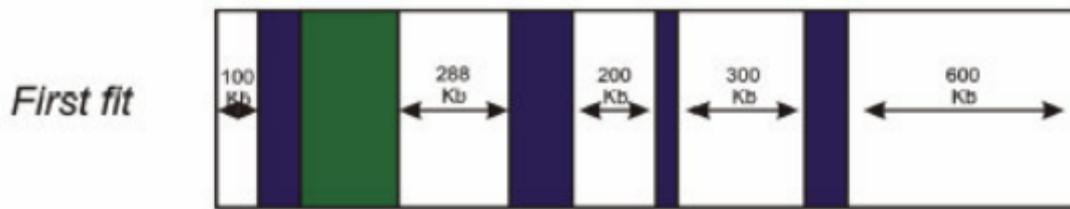
Estudios de simulación han mostrado que *first-fit* y *best-fit* lograron mejores rendimientos en tiempo de asignación y utilización de la memoria que la estrategia *worst-fit*.

Veamos un ejemplo:

Si quisiéramos asignar a memoria un proceso de 212 kb, y tenemos los siguientes espacios libres (espacios en blanco):



Veamos en qué hueco asigna al proceso cada estrategia:



Problema de asignación de memoria

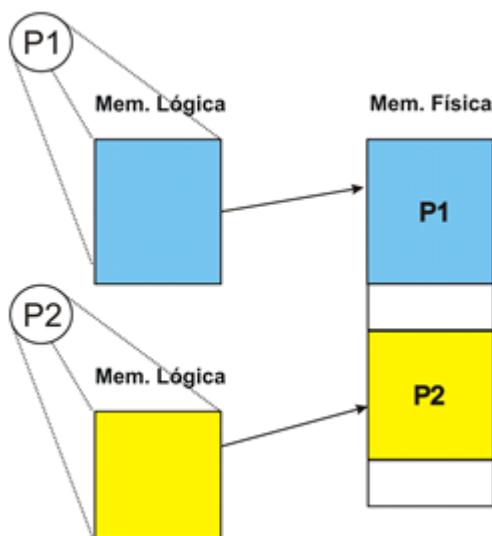
La memoria física puede ser asignada a los diversos procesos en ejecución siguiendo diversas técnicas:

- Asignación contigua
- Asignación dispersa
- **Asignación contigua**

El espacio de direcciones lógicas de un proceso se mapea sobre una única zona de la memoria física: las direcciones de memoria son contiguas.

Métodos:

- Particiones fijas
- Particiones variables

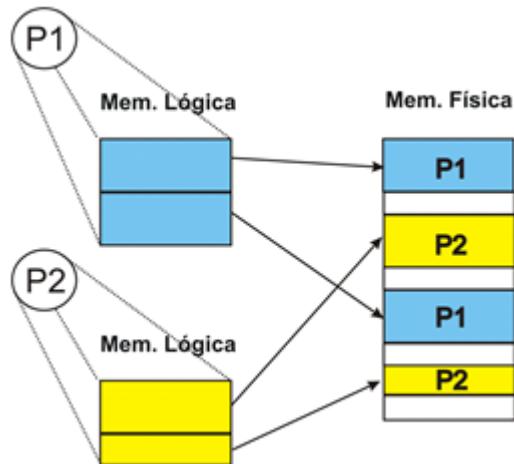


- **Asignación dispersa**

La memoria lógica se divide en fragmentos (páginas o segmentos), que se mapean sobre zonas no contiguas de la memoria física.

Técnicas de asignación dispersa:

- Paganación
- Paganación multinivel
- Segmentación
- Segmentación paginada



Para implementar estas técnicas se necesita el apoyo de la MMU.

Fragmentación

Existen dos tipos de fragmentación:

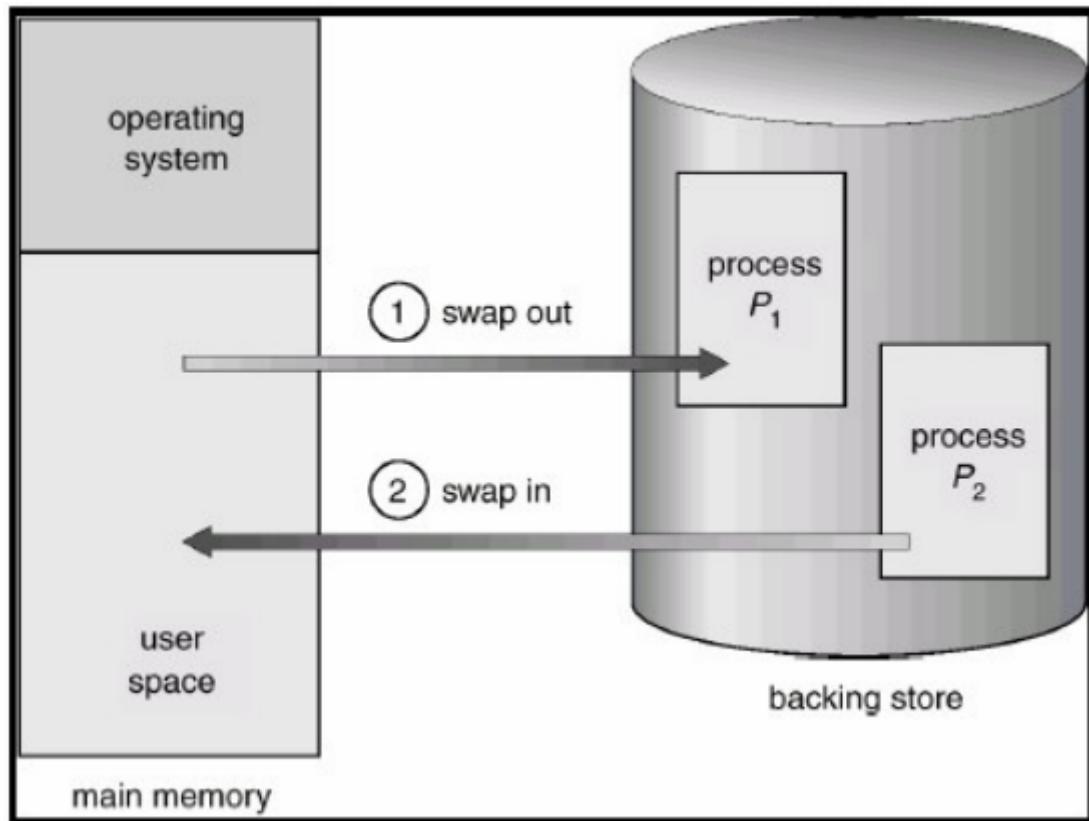
- Fragmentación Interna: Es la pérdida de espacio en disco debido al hecho de que el tamaño de un determinado archivo sea inferior al tamaño del clúster, ya que teóricamente el archivo estaría obligado a ser referenciado como un clúster completo.
- Fragmentación Externa: Se da cuando existe suficiente memoria libre en el sistema para satisfacer un requerimiento de memoria, pero no es posible asignarlo debido a que no es un espacio contiguo.

Dicho lo anterior, vemos que las estrategias presentadas en el ejemplo anterior muestran problemas de fragmentación externa, ya que en la memoria quedan una gran cantidad de espacios pequeños que no son asignados.

Intercambio (Swapping)

Como ya vimos, un proceso debe estar en memoria principal para ser ejecutado. Sin embargo, los procesos pueden ser intercambiados temporalmente, sacándolos de memoria y almacenándolos en el disco, y volviéndolos a llevar a memoria para continuar su ejecución.

Al mecanismo de llevar un proceso desde memoria principal a disco se le denomina *s swap-out*. Al inverso se le denomina *swap-in*. El mayor tiempo consumido en el *swaping* es el tiempo de transferencia.



Memoria Virtual

Introducción

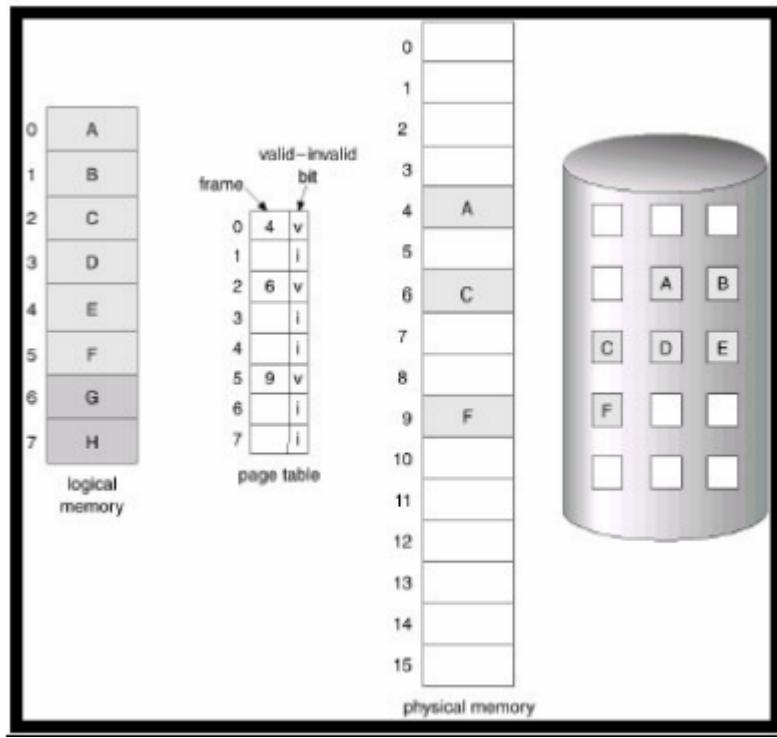
La memoria virtual permite ejecutar procesos que requieren más memoria que la disponible en el sistema, manteniendo en memoria principal solo aquella memoria que el proceso esté utilizando y el resto en el disco. De esta forma el usuario ya no debe preocuparse por las limitaciones de memoria física.

Cada proceso tiene su propio espacio de direccionamiento virtual (o lógico) y la MMU es la encargada de mapear las direcciones virtuales (o lógicas) a físicas.

Implementación

La implementación de memoria virtual es realizada a través de la técnica de paginación bajo demanda. En la paginación bajo demanda los procesos residen en un dispositivo de disco y son puestos en memoria principal cuando es necesario cargarlos para ejecutar. La carga del proceso en memoria no es total, sino que implementa un cargador "perezoso" (lazy swapper), que cargará las páginas según se vayan necesitando.

Utilizar un esquema de este tipo requiere el conocimiento de las páginas que están activas en memoria. Para ello se utiliza el valid-invalid bit, que consiste en agregar a la tabla de páginas un nuevo campo (bit de validez), que indique para cada entrada, si la página se encuentra o no en memoria. Al inicio, la tabla de páginas indicará que ninguna página está en memoria (todos los bits de validez se encontrarán en **i** (invalid)).



En este ejemplo tenemos que el proceso tiene para usar 8 páginas, de las cuales solo usa 6, y de las cuales solo 3 están en memoria principal (A, C, F). Todas las páginas estarán en el disco (incluidas aquellas que también están en memoria principal).

Fallo de página

La memoria cargada en memoria principal se le denomina memoria residente. El acceso a memoria residente por parte de un proceso es tomado como un acceso normal, pero el acceso a memoria no residente genera un fallo de página.

El fallo de página genera un trap a nivel del SO, que activa una rutina de atención que carga la página en memoria principal.

Acceso a Memoria

El acceso a memoria genera la siguiente secuencia de pasos:

- Verificar que el proceso referencia una página correcta dentro de su espacio virtual, ya que no todas las direcciones dentro de su espacio son válidas. Por ejemplo, el acceso fuera de un array puede generar un acceso a una página virtual que no fue asignada al proceso. Si el proceso referencia a una página incorrecta, se genera un error y el proceso termina.
- Si el acceso fue correcto, se busca en la tabla de páginas el frame correspondiente, verificando el bit de validez-invalidez.
- Si el bit es de validez se accede al frame correspondiente y se termina el acceso.
- Si no es válido se genera un trap de page fault, que involucra los siguientes pasos:
 1. Se busca frame libre en memoria principal, si no hay se ejecuta el algoritmo de reemplazo.
 2. Se lee de disco la página a cargar, y se carga en el frame obtenido en el paso anterior.
 3. Se actualiza la tabla de páginas, indicando que la página está disponible en memoria principal.
 4. Se devuelve el control a la instrucción que fue interrumpida por el PF (page fault).

Si se aplica este método se tendrá un sistema puro de paginación bajo demanda. Tener en cuenta que para poder llevarlo a cabo se precisa una tabla de páginas y espacio swap de disco.

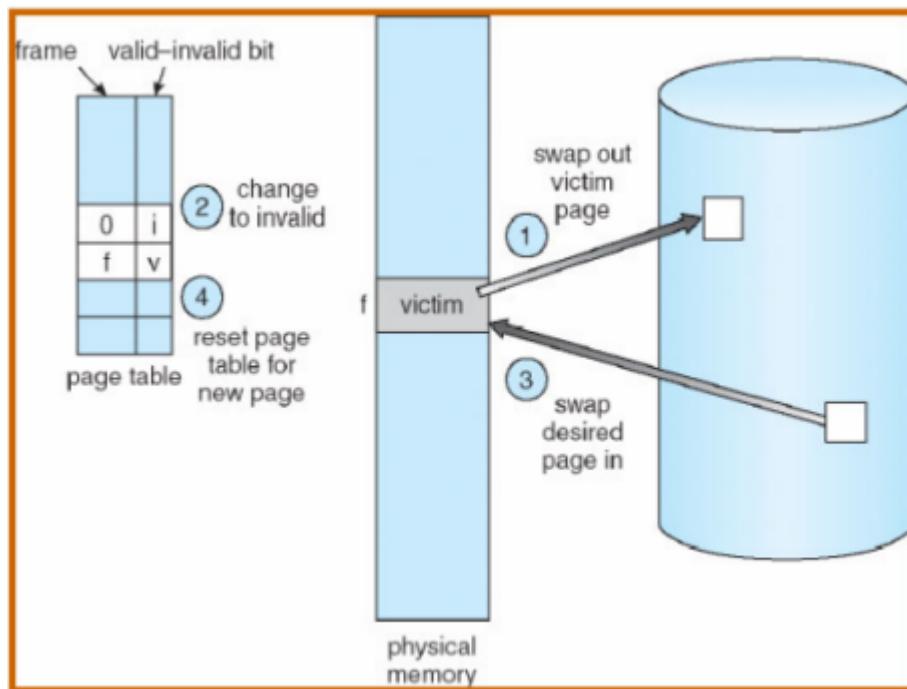
Algoritmos de reemplazo

La necesidad de traer a memoria principal una página en una memoria principal llena, genera la búsqueda de un frame a reemplazar, mediante un algoritmo de reemplazo. Un mal algoritmo de reemplazo puede generar un impacto significativo de degradación del sistema.

Cuando se elige un frame a reemplazar, este será puesto en memoria swap, y ante un eventual uso futuro, volverá a memoria principal a través de un page fault.

Los pasos a seguir cuando reemplazamos frames son los siguientes:

- Elegir el frame mediante algún algoritmo de reemplazo.
- Escribir el frame en memoria swap (swap out) y ajustar la tabla de páginas.
- Cargar la página en el frame correspondiente (swap in).
- Ajustar la tabla de página.



Veamos ahora algunos algoritmos:

- **FIFO (First in First out)**

El algoritmo reemplaza la página que lleva más tiempo en memoria principal. Es un algoritmo fácil de implementar ya que requiere únicamente de una estructura tipo cola, pero reemplaza las páginas sin tener en cuenta las referencias que tuvo.

- **Segunda Oportunidad**

Este algoritmo intenta disminuir la cantidad de fallos de páginas del algoritmo FIFO, teniendo en cuenta las referencias a las páginas.

El algoritmo será igual al anterior, salvo que cada página tendrá un bit que indicará si fue o no referenciada luego de ser cargada a memoria. Al momento del reemplazo, se verifica el bit de referencia; si está encendido, a la página se le da una segunda oportunidad y es puesta al final de la cola. Luego se continúa con la siguiente página que está al principio de la cola. Si el bit está apagado, esta página será seleccionada para ser reemplazada.

Es un tanto ineficiente, pero disminuye la cantidad de fallos de páginas.

- **Óptimo**

En este algoritmo se reemplaza la página que no va a ser usada por el mayor periodo de tiempo. Es imposible de implementar porque requiere conocer a qué páginas accederá el proceso.

- **LRU (Least Recently Used - Recientemente Menos Usada)**

Este algoritmo asocia a cada página el tiempo en que fue referenciada. La página elegida por el algoritmo de reemplazo será la que fue accedida hace más tiempo. Este algoritmo es el que más se aproxima al óptimo y es bastante utilizado por los SO.

- **NRU (No Recientemente Usada)**

En este algoritmo a las páginas se les asigna un bit de referencia y otro de modificación. El bit de referencia se enciende cada vez que se lee o escribe la página, mientras que el de modificación solo se enciende cada vez que se escribe. Cada cierto tiempo el bit de referencia es apagado.

Al ocurrir un fallo de página, los frames son divididos en 4 clases. Se reemplazará un frame al azar de la clase más baja que no esté vacía:

- Clase 0: No referenciada, no modificada
- Clase 1: No referenciada, modificada
- Clase 2: Referenciada, no modificada
- Clase 3: Referenciada, modificada

Al ejecutar el algoritmo de reemplazo, existen dos opciones de páginas a reemplazar:

- Reemplazo global: Un proceso puede reemplazar un frame utilizado por otro. Aunque los PF de un proceso afectan a otros, es el método más usado.
- Reemplazo local: Un proceso reemplaza únicamente los frames que tiene asignado, es por eso que la cantidad de frames de un proceso no varía. La desventaja es que hay marcos que se pueden desperdiciar.

Asignación de frames a procesos e hiperpaginación

Si el SO no implementa una estrategia de asignación de memoria, un proceso que requiera mucha memoria puede hacer colapsar el sistema.

Una forma de asignar frames a procesos podría ser dividir la cantidad de frames del sistema en partes iguales para cada proceso. Este método puede ser ineficiente ya que no todos los procesos consumen la misma cantidad de memoria.

Si un proceso utiliza en forma activa una cantidad mayor de frames de los asignados por el sistema, tendrá un algo porcentaje de fallos de página, dando lugar a que el proceso esté continuamente realizando PF, pasando más tiempo paginando que ejecutando, lo que se conoce como **hiperpaginación**. Se degrada significativamente el rendimiento del sistema.

Procesos

Definición de proceso

Un proceso es un programa en ejecución que necesita estar cargado en memoria y disponer de recursos (CPU, memoria, archivos, dispositivos de E/S) para cumplir su

objetivo. Se trata de una entidad activa. Mientras que los programas son un conjunto de archivos que están almacenados en algún dispositivo de almacenamiento (disco duro, pendrive ...) y cuyo código fuente está escrito en algún lenguaje de programación. Cuando este conjunto de archivos se ejecutan, entonces pasa a ser un proceso.

Procesos en memoria

Un proceso en memoria se constituye de varias secciones:

- **Código (text)** : instrucciones del proceso.
- **Datos (data)** : variables globales del proceso.
- **Memoria dinámica (Heap)** : Memoria dinámica que genera el proceso.
- **Pila (Stack)** : utilizado para preservar el estado en la invocación anidada de procedimientos y funciones.

Estados de los procesos

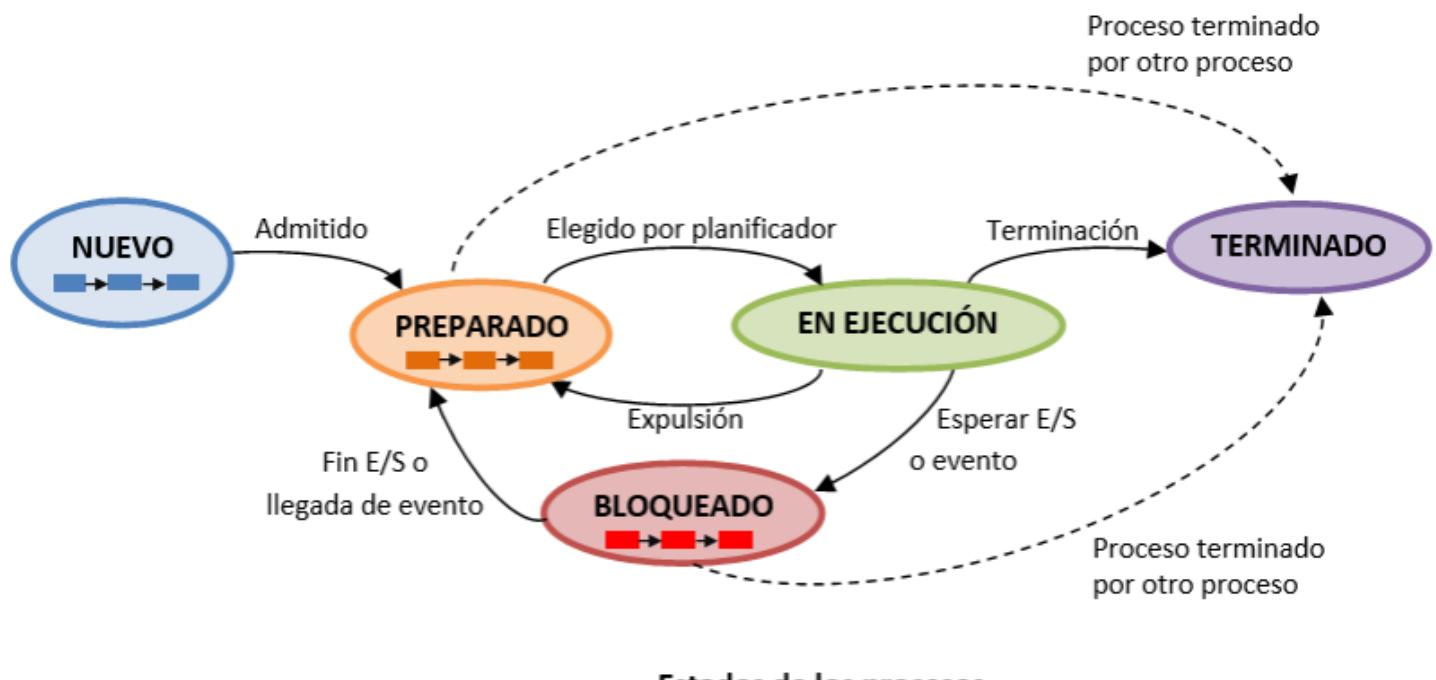
El estado de un proceso se define por su actividad actual, cambiando a medida que se ejecuta. La ejecución de un proceso alterna una serie de ráfagas de CPU y E/S.

Los estados de un proceso son:

- **Nuevo** : Cuando el proceso es creado.
- **En Ejecución** : El proceso tiene asignado un procesador y está ejecutando sus instrucciones.
- **Bloqueado** : El proceso está esperando por un evento (que se complete un pedido de E/S o una señal).
- **Preparado** : El proceso está listo para ejecutar, solo necesita del recurso procesador.
- **Terminado** : El proceso finalizó su ejecución.

Transiciones entre los estados

Veamos ahora como los procesos pueden cambiar de estados a partir de determinados hechos. A continuación se muestra el diagrama de estados y transiciones de los procesos:



- **Nuevo -> Preparado** : el SO está preparado para admitir un proceso más.
- **Preparado -> Ejecución** : el planificador escoge un proceso para la ejecución.

- **Ejecución -> Preparado** : el proceso en ejecución es interrumpido y expulsado del procesador porque ya ha consumido su tiempo asignado o porque otro proceso de mayor prioridad está esperando.
- **Ejecución -> Bloqueado** : el proceso abandona voluntariamente la CPU y espera a un evento externo.
- **Bloqueado -> Preparado** : finaliza el evento que estaba esperando el proceso y pasa al estado preparado.
- **Ejecución -> Terminado** : el proceso termina su ejecución (terminación normal).
- **Preparado/Bloqueado -> Terminado** : el proceso es eliminado (terminación anormal).

Listas y colas de procesos

Los procesos, según su estado, deberán esperar por determinados eventos, como ya vimos. Puede suceder, que más de un proceso esté esperando por el mismo evento, es por eso que se deben organizar en diferentes colas o listas.

- **Lista de procesos del sistema (job queue)** : Esta será una lista especial, porque los procesos que están en ella no esperan por nada en particular, sino que es la lista de todos los procesos del sistema. Al crearse un nuevo proceso se agrega el PCB a esta lista. Cuando el proceso termina su ejecución es borrado.
- **Cola de procesos listos (ready queue)** : Esta cola se compondrá de los procesos que estén en estado listo. La estructura de esta cola dependerá de la estrategia de planificación utilizada.
- **Cola de espera de dispositivos (device queue)** : Los procesos que esperan por un dispositivo de E/S particular son agrupados en una lista específica al dispositivo. Cada dispositivo de E/S tendrá su cola de espera, por lo que existirán varias device queue.

Bloque de Control de Proceso (PCB)

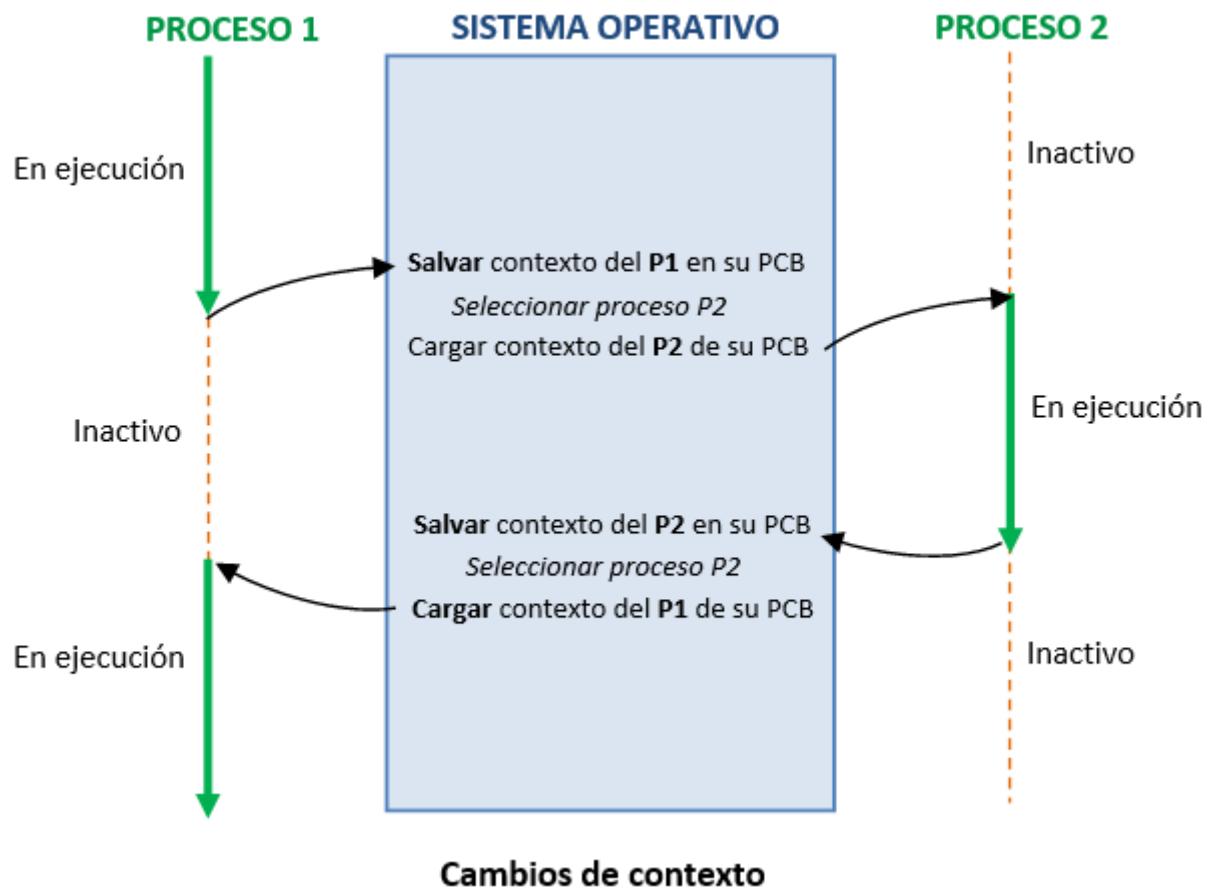
Cuando un proceso se ejecuta, el SO le asigna un espacio de direcciones de memoria (que contiene las instrucciones, los datos y la pila que es una estructura para almacenar y recuperar datos del proceso) y lo añade a la tabla de procesos.

El SO guarda en la tabla de procesos por cada proceso una estructura de datos llamada Bloque de Control de Proceso (PCB) que almacena la siguiente información:

- **Identificación** de proceso: del proceso en sí (PID), del proceso padre (PPID) y de usuario.
- **Información de estado** del proceso: preparado, en ejecución, bloqueado,
- **Prioridad** del proceso.
- **Dirección de memoria** donde se ha cargado el proceso
- **Otros** : recursos utilizados, valores de los registros del procesador, propietarios, permisos.

Cambio de contexto (context switch)

Para dar sensación de ejecución simultánea o multiprogramación, el tipo de CPU debe repartirse entre los procesos. Esto implica **cambios de contexto** que consisten en quitarle la CPU al proceso “en ejecución” y asignársela a otro estado “preparado”. Esta operación la realiza un componente del SO llamado **dispatcher** o **planificador a corto plazo** y en ella se guarda el contexto del proceso en ejecución en su PCB y se restaura el contexto del nuevo proceso a ejecutar mediante su PCB.



Los cambios de contexto pueden suponer una sobrecarga si se utilizan con mucha frecuencia. En general, suelen producirse cuando un proceso finaliza, es expulsado o se suspende.

Comunicación entre procesos

Procesos que se ejecutan concurrentemente pueden ser procesos independientes o cooperativos. Un proceso es cooperativo si puede afectar o verse afectado por los restantes procesos que se ejecuten en el sistema, y es independiente si no.

Evidentemente, cualquier proceso que comparta datos con otro será cooperativo. Veamos algunas razones por las cuales es bueno tener un entorno que permita la cooperación de procesos:

- Compartir información. Dado que varios usuarios pueden estar interesados en la misma información, se debe proveer un acceso concurrente a ella.
- Acelerar cálculos. Si deseamos que una determinada operación se ejecute rápidamente, debemos dividirla en subtareas ejecutándose cada una de ellas en paralelo. Esto se consigue solo si hay múltiples CPU o varios canales de E/S.

El mecanismo que provee esto es IPC (InterProcess Communication), que permite intercambiar datos e información.

Hilos (Thread)

La mayoría de los SO proporcionan características que permiten que un proceso tenga múltiples hilos de control.

¿Qué es un hilo?

Un hilo es una unidad básica de utilización de CPU, la cual contiene un id de hilo, su propio program counter, un conjunto de registros y una pila; se representa a nivel del SO con una estructura llamada TCB (Thread Control Block)

Los hilos comparten con otros hilos que pertenecen al mismo proceso la sección de código, la sección de datos, entre otras cosas. Si un proceso tiene múltiples hilos, puede realizar más de una tarea a la vez (esto es real cuando se posee más de una CPU).

Ventajas de usar hilos

- **Respuesta** : el tiempo de respuesta mejora, ya que el programa puede continuar ejecutándose, aunque parte de él esté bloqueado.
- **Compartir recursos** : los hilos comparten la memoria y los recursos del proceso al que pertenecen, por lo que se puede tener varios hilos de ejecución dentro del mismo espacio de direcciones.
- **Economía** : es más fácil la creación, cambio de contexto y gestión de hilos que de procesos.
- **Utilización múltiples CPUs** : permite que hilos de un mismo proceso ejecuten en diferentes CPUs a la vez. En un proceso mono-hilo, un proceso ejecuta en una única CPU, independientemente de cuantas tenga disponibles.

Hilos a nivel de usuario y de kernel

- **Hilos a nivel de usuario** : son implementados en alguna librería. Estos hilos se gestionan sin soporte del SO, el cual solo reconoce un hilo de ejecución.
- **Hilos a nivel de kernel** : el SO es quien crea, planifica y gestiona los hilos. Se reconocen tantos hilos como se hayan creado.

Los hilos a nivel de usuario tienen como beneficio que su cambio de contexto es más sencillo que el cambio de contexto entre hilos de kernel. Además, se pueden implementar aún si el SO no utiliza hilos a nivel de kernel. Otro de los beneficios consiste en poder planificar diferente a la estrategia del SO.

Los hilos a nivel de kernel tienen como gran beneficio poder aprovechar mejor las arquitecturas multiprocesadores, y que proporcionan un mejor tiempo de respuesta, ya que si un hilo se bloquea, los otros puedes seguir ejecutándose.

Planificación

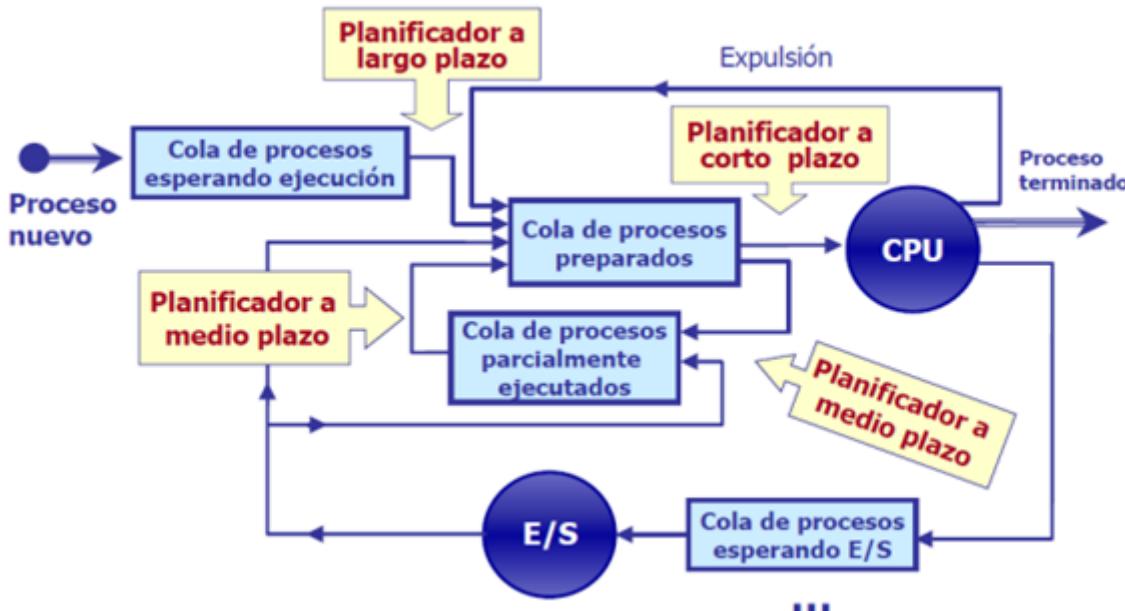
La planificación es la base para lograr la multiprogramación.

En un sistema multiprogramado, generalmente en un determinado instante existirán varios procesos que requieren el procesador a la vez, el componente del SO que realiza la operación de elegir el proceso a utilizar es el planificador.

Principales planificadores de CPU

- **Planificador a largo plazo** :
 - Selecciona procesos de la cola de esperando ejecución y los carga a memoria
 - Controla el grado de multiprogramación. Es importante que elija un conjunto equilibrado de procesos.
 - Se ejecuta con poca frecuencia.
- **Planificador a corto plazo** :
 - Selecciona entre los procesos preparados en memoria y les asigna la CPU.
 - Se ejecuta con mucha frecuencia.

- **Planificador a medio plazo :**
 - Decide qué proceso pasa de la memoria principal a la secundaria (memoria virtual) o viceversa.



El SO enlaza los PCB's de los procesos que están en el mismo estado a las diversas colas que puedan existir.

Esquemas de planificación

Se invoca al planificador cuando:

1. Cuando un proceso cambia de ejecutando a bloqueado.
2. Cuando un proceso finaliza.
3. Cuando un proceso cambia de ejecutando a listo.
4. Cuando un proceso cambia de bloqueado a listo.
5. Cuando se crea un nuevo proceso.

Cuando ocurren los dos primeros casos, el planificador es invocado debido a que el proceso en ejecución libera el procesador.

Los últimos tres casos se dan solamente cuando el planificador es expropiativo, ya que puede quitar el procesador a un proceso que estaba ejecutando para dárselo a otro.

Planificación no apropiativa y apropiativa

- Planificación no apropiativa (non-preemptive):
 - Algoritmos no expulsivos.
 - Los procesos se ejecutan hasta que terminan o se bloquean.
 - Sencillo de implementar.
 - Rendimiento negativo en general.
- Planificación apropiativa (preemptive):
 - Algoritmos expulsivos.
 - Los procesos pueden ser expulsados de la CPU.
 - Mayor coste de implementación. Necesitan soporte hardware adicional (relojes).
 - Mejora el servicio y evita monopolización de la CPU.

Medidas para poder evaluar los algoritmos de planificación

- **Utilización de CPU :** es el porcentaje de uso útil que tiene un procesador.
- **Rendimiento (Throughput) :** número de procesos terminados por unidad de tiempo.

- **Tiempo de retorno** : tiempo desde que un proceso se carga hasta que finaliza su ejecución.
- **Tiempo de espera** : es la suma de los tiempos que un proceso estuvo en la cola de procesos listos.
- **Tiempo de respuesta** : tiempo desde la carga hasta que el proceso da su primera respuesta.

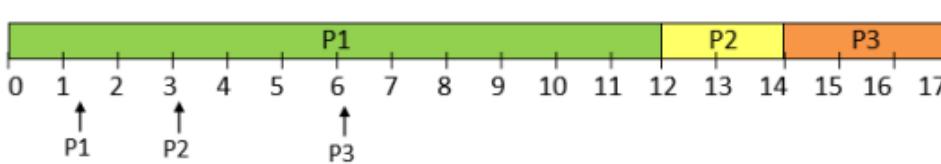
Algoritmos de planificación

- **FCFS (First Come First Served)**

La CPU es asignada a los procesos en el mismo orden que lo solicitan. Es un algoritmo no expulsivo.

- Ventajas
 - Sencillo de implementar (cola FIFO).
- Inconvenientes
 - Mal tiempo de espera
 - Efecto convoy (procesos con largas ráfagas de CPU retrasan a procesos con ráfagas cortas).
 - No válido para procesos interactivos.

Proceso	Instante llegada	Tiempo CPU
P1	0	12
P2	2	2
P3	5	3



$$\text{Tiempo medio de espera} = (0+10+9)/3=6,3$$

- **SJF (Shortest Job First)**

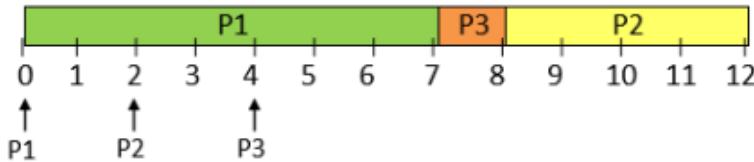
Primero el que menos tiempo total de CPU requiere.

Se escoge el proceso de la cola de preparados con una próxima racha de CPU más corta y se ejecuta hasta que se termine o se suspenda. Si hay varios procesos con rachas de CPU iguales, se puede aplicar FIFO. Algoritmo no expulsivo.

- Ventajas
 - Optimiza el tiempo de espera
 - Favorece los procesos orientados a E/S
- Desventajas
 - Es costoso averiguar cuándo dura la siguiente racha de CPU
 - Inanición de los procesos con rachas de CPU largas

Proceso	Instante llegada	Tiempo CPU
P1	0	7
P2	2	4
P3	4	1

Tiempo medio de espera =
 $(0+6+3)/3 = 3$

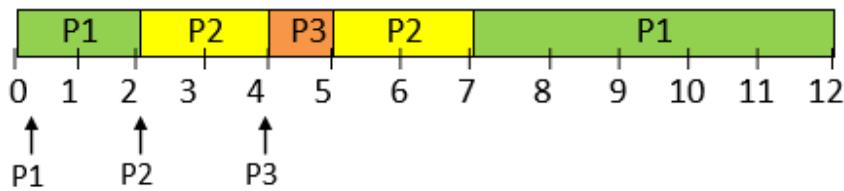


- **SRTF (Shortest Remaining Time First)**

Primero al que menos tiempo de CPU le queda para acabar.

Versión apropiativa de SJF (como el SJF, solo que puede echar a los procesos)

Proceso	Instante llegada	Tiempo CPU
P1	0	7
P2	2	4
P3	4	1



Tiempo medio de espera =
 $(5+1+0)/3 = 2$

- **Planificación por prioridades**

Primero el que tiene más prioridad.

Cada proceso tiene asignada una prioridad. El planificador selecciona el proceso con prioridad más alta (a igual prioridad se selecciona con FCFS).

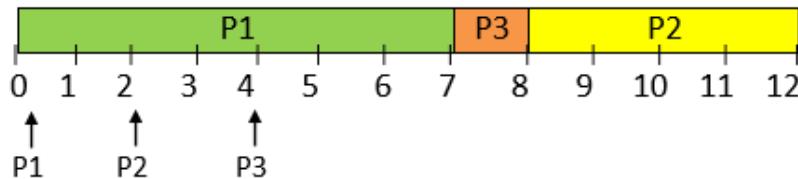
Las prioridades pueden ser dinámicas (cambian con el tiempo) o estáticas (se mantienen).

- Inconvenientes

- Riesgo de inanición de procesos con prioridad baja. Una solución sería aumentar la prioridad con el incremento del tiempo de espera.

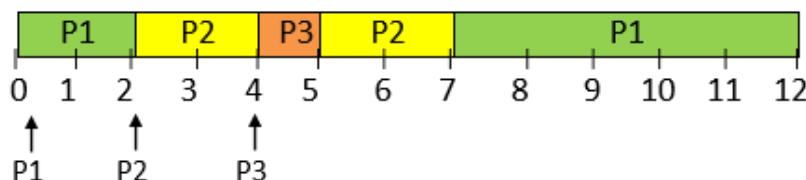
Proceso	Instante llegada	Tiempo CPU	Prioridad
P1	0	7	5
P2	2	4	10
P3	4	1	15

Prioridades sin expulsión:



Tiempo medio de espera:
 $(0+6+3)/3=3$

Prioridades con expulsión:



Tiempo medio de espera =
 $(5+1+0)/3 = 2$

• Planificación Circular (Round Robin)

Todos el mismo tiempo por turnos.

A cada proceso se le asigna una cantidad de tiempo de CPU llamada "quantum". Si el proceso tiene un intervalo de CPU mayor que el quantum es expulsado de la CPU.

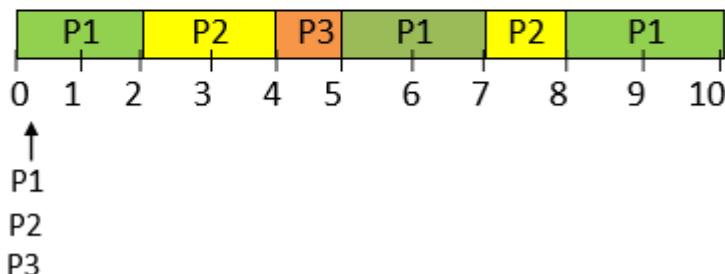
La cola de preparados se gestiona con una política FIFO.

Si el valor del quantum es grande el algoritmo degenera en FCFS. Si es pequeño se generará sobrecarga debido a cambios de contexto.

Es Equitativo.

Ej. (Quantum = 2):

Proceso	Instante llegada	Tiempo CPU
P1	0	6
P2	0	3
P3	0	1



Tiempo medio de espera=
 $(4+5+4) / 3 = 4,3$

• Multilevel Queue

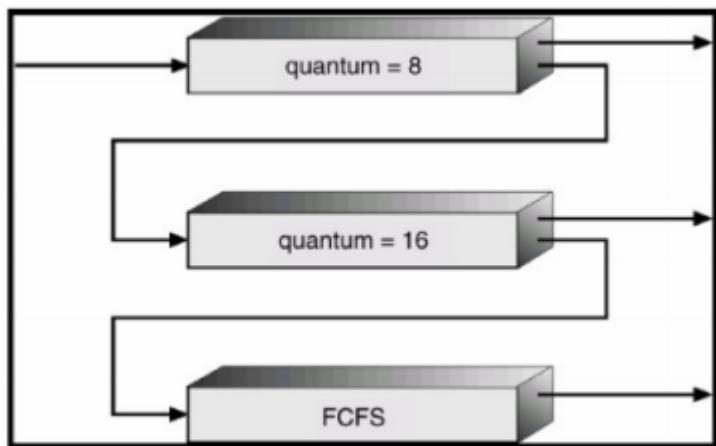
Este algoritmo propone dividir la lista de procesos listos en varias colas, una para cada tipo de proceso. Cabe destacar que los procesos no podrán cambiar de cola, que cada cola tendrá su propio algoritmo de planificación, y que existirá un algoritmo de planificación entre colas.

- **Multilevel Feedback Queue**

Este algoritmo se diferencia con el anterior en que los procesos si pueden cambiar de nivel, dependiendo del uso del CPU que tengan. La cola de más alta prioridad corresponderá a los I/O bound, la más baja a los CPU-bound.

Un algoritmo así se define por:

- Cantidad de colas
- Algoritmo de cada cola
- Criterio para subir de nivel un proceso
- Criterio para bajar de nivel un proceso
- Criterio para asignar un proceso nuevo a una de las colas



- **Sistemas Multiprocesador**

En un sistema simétrico, cualquier procesador ejecuta procesos de usuario. Se puede asignar una cola de listos a cada CPU, lo cual es conveniente para el uso de la caché. Pueden haber desbalances de trabajo entre procesadores, por lo cual se pueden migrar procesos de cola para balancear la carga nuevamente.

Programación Concurrente

¿Qué es la programación concurrente?

Se conoce por programación concurrente a la rama de la informática que trata de las técnicas de programación que se usan para expresar el paralelismo entre tareas y para resolver los problemas de comunicación y sincronización entre procesos.

El principal problema de la programación concurrente corresponde a no saber en qué orden se ejecutan los programas (en especial los programas que se comunican). Se debe tener especial cuidado en que este orden no afecte el resultado de los programas.

Sección crítica y exclusión mutua

El método más sencillo de comunicación entre procesos de un programa concurrente es el uso común de unas variables de datos. Esta forma tan sencilla de comunicación puede llevar, no obstante, a errores en el programa ya que el acceso concurrente puede hacer que la acción de un proceso interfiera en las acciones de otro de una forma no adecuada.

Para evitar este tipo de errores se pueden identificar aquellas regiones de los procesos que acceden a variables compartidas y dotarlas de la posibilidad de ejecución como si fueran una única instrucción.

Se denomina **Sección Crítica** a aquellas partes de los procesos concurrentes que no pueden ejecutarse de forma concurrente o, también, que desde otro proceso se ven como si fueran una única instrucción. Esto quiere decir que si un proceso entra a ejecutar una sección crítica en la que se accede a unas variables compartidas, entonces otro proceso no puede entrar a ejecutar una región crítica en la que acceda a variables compartidas con el anterior.

Las secciones críticas se pueden **excluir mutuamente**. Para conseguir dicha exclusión se deben implementar protocolos software que impidan el acceso a una sección crítica mientras está siendo utilizada por un proceso.

Semáforos

Dijkstra dio en 1968 una solución al problema de la exclusión mutua con la introducción del concepto de semáforo binario. Esta técnica permite resolver la mayoría de los problemas de sincronización entre procesos y forma parte del diseño de muchos SO.

Un semáforo binario es un indicador (S) de condición que registra si un recurso está disponible o no. Un semáforo binario solo puede tomar dos valores: 0 y 1. Si, para un semáforo binario $S=1$ entonces el recurso está disponible y la tarea lo puede utilizar; si $S=0$ el recurso no está disponible y el proceso debe esperar.

Los semáforos se implementan con una cola de tareas a la cual se añaden los procesos que están en espera del recurso.

Un semáforo binario se puede definir como un tipo de datos especial que sólo puede tomar los valores 0 y 1, con una cola de tareas asociada y con sólo tres operaciones para actuar sobre él:

- INIT (S, val): inicializa al semáforo en el valor val (0 ó 1)
- P(S):
 if S = 1 **then**
 S := 0
 else
 Suspender la tarea que hace la llamada y ponerla en la cola de tareas
- V(S):
 if la cola de tareas está vacía **then**
 S := 1
 else
 Reanudar la primera tarea de la cola de tareas

1. La operación **INIT** se debe sellar a cabo antes de que comience la ejecución concurrente de los procesos ya que su función exclusiva es dar una valor inicial al semáforo.
2. Un proceso que corre la operación **P** y encuentra el semáforo a 1, lo pone a 0 y prosigue su ejecución. Si el semáforo está a 0 el proceso queda en estado de *bloqueado* hasta que el semáforo se libera.
3. Cuando se ejecuta la operación **V** puede haber varios procesos en la lista o cola. El proceso que la dejará para pasar al estado listo dependerá del esquema de gestión de la Cola. Si no hay ningún proceso en espera el semáforo se deja libre para el primero que lo requiera.

El semáforo binario resulta adecuado cuando hay que proteger un recurso que pueden compartir varios procesos, pero cuando lo que hay que proteger es un conjunto de recursos similares, se puede usar una versión más general de semáforo que lleve la cuenta del número de recursos disponibles. En este caso el semáforo se inicializa con el número total de recursos disponibles (N) y las operaciones P y V se diseñan de modo que se impida el acceso al recurso protegido por el semáforo cuando el valor de éste es menor o igual que cero. Cada vez que se solicita y obtiene un recurso, el semáforo se decremente y se incrementa cuando uno de ellos se libera.

Las operaciones que tenemos son las mismas, con algunas diferencias en su semántica:

- **INIT (S, val):** inicializa al semáforo en el valor val (puede ser cualquier valor)
- **P(S):**
if S > 0 then
 S = S -1;
else
 Suspender la tarea que hace la llamada y ponerla en la cola de tareas
- **V(S):**
if la cola de tareas está vacía then
 S = S +1;
else
 Reanudar la primera tarea de la cola de tareas

Semáforos: mutua exclusión

La exclusión mutua se realiza fácilmente utilizando semáforos. La operación P se usará como procedimiento de bloqueo antes de acceder a una sección crítica y la operación V como procedimiento de desbloqueo. Se utilizarán tantos semáforos como clases de secciones críticas se establezcan.

Gestión de Entrada/Salida

La **gestión de entrada/salida** es una de las funciones más importantes del SO, ya que el SO debe ser capaz de manejar los diferentes periféricos existentes.

Para ello **debe** :

- Enviar órdenes a los dispositivos de E/S
- Determinar el dispositivo que necesita la atención del procesador
- Detectar las interrupciones
- Controlar los errores
- Proporcionar una interfaz entre los dispositivos y el resto del sistema. Esta interfaz debe ser:
 - Sencilla y fácil de usar
 - Debe ser la misma para todos los dispositivos

El SO tiene varias maneras de llevar a cabo la E/S:

- **E/S programada** : el procesador ejecuta un programa que controla las operaciones de E/S. El problema es que el procesador se tiene que quedar esperando (parado) a recibir respuesta.
- **E/S controlada por interrupciones** : los dispositivos envían una señal de interrupción para llamar la atención del sistema.
- **E/S mediante el uso de DMA (acceso directo a memoria)** : un chip se encarga de la transferencia y accede a la memoria para leer o escribir datos que recibe y envía el dispositivo sin pasar por el procesador.

Actualmente los discos duros, unidades de CD, DVD, Blueray, admiten DMA y la tienen activada por defecto.

Dado que la velocidad del procesador es muy superior a la de los dispositivos de E/S, se utilizan **técnicas de almacenamiento intermedio** para mejorar el rendimiento del sistema:

- **Caching** : consiste en almacenar una caché temporal, de rápido acceso, los datos que se usan con más frecuencia.
- **Buffering** : consiste en utilizar un área de memoria como buffer, simulando un dispositivo o un periférico lógico, que hará de dispositivo intermedio entre el periférico real y el procesador. El buffer es independiente del dispositivo de entrada y/o salida, por lo que permite que el procesador comience a trabajar leyendo o almacenando en el buffer mientras la información del periférico se va almacenando o extrayendo del buffer. Esto evita que un periférico lento afecte al rendimiento del equipo informático.
- **Spooling** : técnica en la cual la computadora introduce trabajos en un buffer (un área especial en memoria o en un disco), de manera que un dispositivo pueda acceder a ellos cuando esté listo. El spooling es útil en caso de dispositivos que acceden a los datos a distintas velocidades. El buffer proporciona un lugar de espera donde los datos pueden estar hasta que el dispositivo (generalmente más lento) los procesa. Esto permite que la CPU pueda trabajar en otras tareas mientras que espera que el dispositivo más lento acabe de procesar el trabajo.

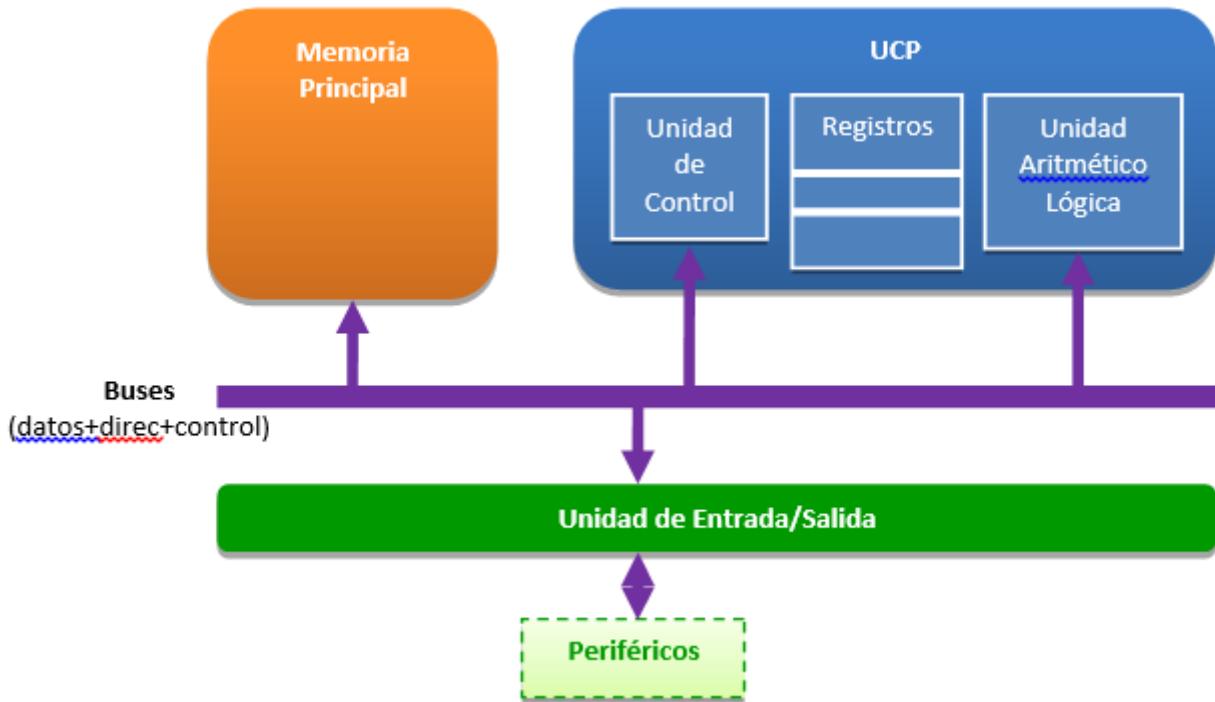
La aplicación más común del spooling es la impresión. En este caso, los documentos son cargados en un área de un disco, y la impresora los saca de éste a su propia velocidad. El usuario puede entonces realizar otras operaciones en el ordenador mientras la impresión tiene lugar en segundo plano. El spooling permite también que los usuarios coloquen varios trabajos de impresión en una cola de una vez, en lugar de esperar a que cada uno acabe para enviar el siguiente.

Unidad de Entrada/Salida

La Unidad de Entrada/Salida (chipset) permite la comunicación de la CPU y la Memoria Principal con el exterior: impresoras, monitor, teclado, etc.

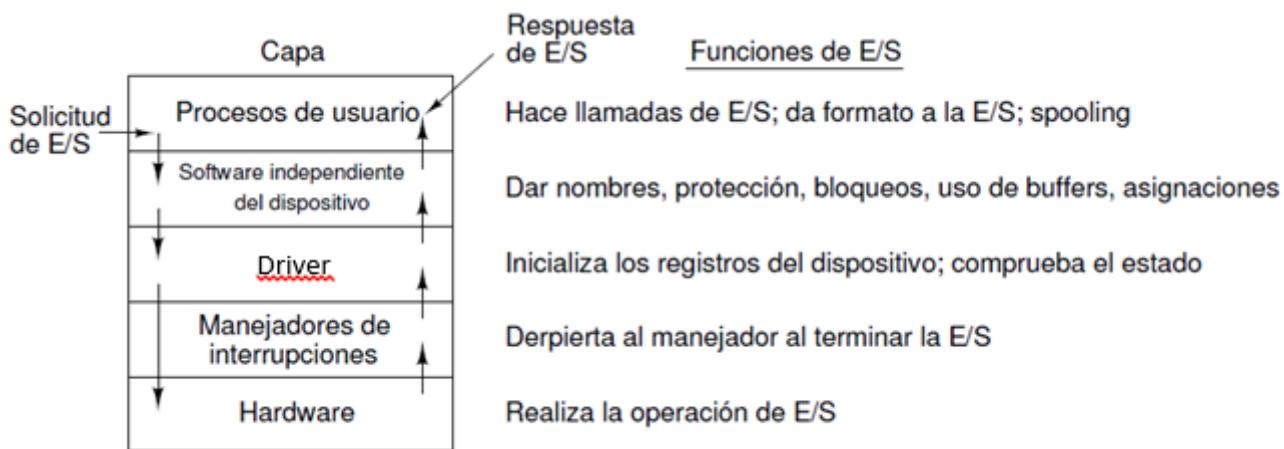
Para que se pueda llevar a cabo el intercambio de información se deben realizar las siguientes tareas:

- **Direccionamiento** : selección del dispositivo de E/S implicado en una transferencia determinada.
- **Sincronización** de CPU y periféricos: es necesario coordinar la actividad de la CPU con los periféricos, ya que sus velocidades de trabajo son distintas.
- **Transferencia** de datos desde o hacia el dispositivo seleccionado.



Software de Entrada/Salida

El software de E/S se organiza en niveles. Los del nivel inferior ocultan las particularidades del hardware a los del nivel superior que presentan una interfaz simple y uniforme al usuario.



Hardware de Entrada/Salida

En general, las unidades de E/S constan de:

- Un componente electrónico denominado controladora
- Un componente mecánico, que es el dispositivo mismo
- **Controladoras**

Las principales funciones de las controladoras son:

- Comunicación en el periférico, intercambio de órdenes, información del estado.
- Detección de errores.
- Comunicación con el procesador, descodificadores de órdenes, datos, información de estado, reconocimiento de dirección.

Al código específico que el SO utiliza para programar una controladora se le conoce como manejador o driver de la controladora. De este modo, para llevar a cabo las tareas de E/S,

el SO, usando el driver, se comunica con la controladora a través de una serie de registros específicos que cada controladora tiene. Cuando la orden ha sido cumplida, la controladora produce una interrupción con el fin de permitir que la CPU atienda al SO para comprobar los resultados de la operación de E/S. Para dicha comprobación se utilizan los valores de los registros de la controladora que informan sobre el estado final.

- **Dispositivos de E/S**

Los periféricos se pueden clasificar en función de si gestionan la información por bloques o caracteres.

Gestión de Ficheros

Introducción

- Fichero o archivo: Conjunto de información de un determinado tipo que está almacenada en un dispositivo de almacenamiento. Ejemplo: documento de texto, sonido, imagen, ...
- Carpeta o directorio: Tipo especial de fichero que se utiliza para organizar ficheros (u otras carpetas).
- Sistema de ficheros: Parte del SO que permite “administrar” la información almacenada de los dispositivos de E/S en forma de ficheros.
 - Objetivos:
 - Crear, modificar o borrar ficheros (o carpetas)
 - Controlar el acceso a los ficheros (mediante permisos)
 - Permitir intercambio de datos entre ficheros
 - Permitir realizar copias de seguridad de los ficheros
 - Permitir el acceso a los ficheros mediante nombres simbólicos

Ficheros

-> Nombre y extensión de los ficheros

Los archivos generalmente se componen de:

- Nombre: La mayoría de SO permiten usar nombres de hasta 255 caracteres y algunos SO, como Linux, distinguen entre mayúsculas y minúsculas.
- Extensión: Sirve para saber el programa que permite ejecutar o abrir un fichero. Algunos SO como Linux no necesitan el uso de extensiones.

-> Tipos de ficheros:

- Ficheros normales o regulares: Aquellos ficheros que contienen datos (información).
- Directorios: Fichero que se utiliza para organizar los ficheros (u otras carpetas).
- Ficheros especiales de dispositivos: representan a dispositivos de E/S.

-> Información que contiene un fichero:

- Nombre
- Tamaño
- Fechas: de creación, modificación, ...
- Propietario
- Permisos (lectura, escritura, ejecución, ...)
- Ubicación
- Enlaces: puntos desde los que se puede acceder al fichero

-> Operaciones que se puedes hacer sobre un fichero:

- Crear

- Abrir
- Escribir
- Cerrar
- Borrar

Directorio

-> Operaciones que se pueden hacer sobre un directorio:

- Crear
- Entrar
- Salir
- Leer su contenido
- Añadir o Eliminar en él archivos o directorios
- Borrar

La mayoría de los SO tienen un sistema de archivos de estructura jerárquica, en el que los directorios parten de uno llamado directorio raíz, y del que cuelgan todos los demás en forma de árbol, de ahí que se utilicen términos como árbol de subdirectorios.

-> Directorios especiales:

- Existen dos tipos:
 - . directorio actual
 - .. directorio padre

-> Rutas:

Concatenación de directorios y subdirectorios para llamar a un archivo en una estructura de directorios.

- Tipos:
 - Absolutas: se llama al archivo desde el directorio raíz hasta el archivo. Ejemplo: c:\web\imagenes\logo.gif
 - Relativas: se llama al archivo desde el directorio actual en el que estemos. Ejemplo: si estamos en la carpeta "web" la ruta hasta llegar al archivo "logo.gif" sería: imagenes\logo.gif

Métodos de asignación

Métodos para asignar espacio a cada fichero dentro del disco.

- Asignación contigua: los bloques de un fichero se encuentran de forma contigua en el disco.
- Asignación enlazada: en cada bloque está parte de los datos del fichero y una pequeña parte para indicar el siguiente bloque que contiene los datos del fichero.

Características técnicas y funcionales de los sistemas operativos: Windows, Linux, Unix y otros. Sistemas operativos para dispositivos móviles.

Sistemas Windows

Los SO Windows han ido evolucionando desde 1981, en que empezó a comercializarse MS-DOS. Era un SO monotarea, monousuario y monoprocesador. El interfaz gráfico de usuario (GUI) se empezó a probar con Windows 1.0 en 1985 y se lanzaba desde MS-DOS. En 1990 aparece Windows 3.0, evolucionando hasta el famoso Windows 3.1.

Windows NT aparece en 1993 y representa un SO de 32 bits para el Intel 60386. La versión NT 4.0 (1996) incluía el GUI de Windows 95 en que los componentes gráficos, anteriormente en modo usuario, pasaron a modo núcleo (NT Executive). Windows 2000 heredó la arquitectura NT incluyendo el servicio de directorio (AD, Active Directory). Case en paralelo, Windows 95 evolucionó a Windows 98 y Windows Me, que mantenían el código de 16 bits lo que no les hacía tan eficientes en procesadores 386 y posteriores. No soportan NTFS. La tecnología NT y el interfaz mejorado de Windows 95 se fusionan en Windows XP (2001), evolucionando a Windows Vista (2006), Windows 7, Windows 8 y Windows 10.

Las características de la tecnología NT se aplican a los SO profesionales de Microsoft, destacando:

- SO de 32 bits: No compatible hacia atrás. Por tanto supone una ruptura con Windows Me. Las versiones Windows 2000 y XP son la evolución del código original de Windows NT.
- Independencia de memoria separada: La ejecución de un programa se hace en regiones de memoria distintas, lo que evita que las inestabilidades afecten al resto. El SO gestiona el uso de la memoria, evitando perder el control de la máquina.
- Multitarea apropiativa (preemptive): O ejecución simultánea de aplicaciones. El SO asigna los recursos evitando inanición. Se opone a la multitarea colaborativa (no apropiativa) tipo Windows 95.
- Multiusuario y multiprocesador: Se gestiona la concurrencia al sistema de distintos usuarios en red y se pueden usar varios procesadores en la misma máquina, asignándoles distintas tareas.
- Portabilidad: Se refiere a la independencia del hardware, implementado con una capa HAL (Hardware Abstraction Layer). El resto de código es común a cualquier sistema.
- Seguridad de dominio: Consiste en la inclusión de autenticación de usuarios para el acceso a recursos de red. Los controladores de dominio se encargan de validar usuarios de alta en la BBDD llamada SAM (Security Account Manager). A partir de Windows 2000 Server, los dominios se integran en el servicio de directorio Microsoft, el AD.
- NTFS: La nueva tecnología de sistema de ficheros (NTFS) incluye seguridad y se basa en la creación de listas de control de acceso (ACL) para cada archivo o directorio. NT también incluye soporte para FAT y HPFS (OS/2).
- Tolerancia a fallos: Se incluyen mecanismos de continuidad en presencia de fallos y soporte RAID.

Usuarios, Grupos y Dominios

Cada usuario necesita ser identificado, es decir, disponer de una cuenta para iniciar sesión en el sistema. La identificación de una cuenta de usuario se realiza con un nombre y se asocia una contraseña. La cuenta de máximo privilegio es la de Administrador, similar a root en entornos Unix.

Para la gestión de cuentas, usuarios, se agrupan en Grupos, en general con criterios de privilegios o perfiles comunes. Los tipos de grupos que se distinguen en entornos Windows son:

- globales: pueden contener usuarios de un mismo dominio
- locales: con usuario y grupos globales de distintos dominios
- universales: que incluyen usuarios, grupos globales y universales de distintos dominios. No soportado en Windows NT.

NTFS incorpora seguridad a nivel de archivo y carpeta. Con las ACL se definen permisos de usuario o grupo independientemente. Los permisos de Windows son más completos que los nativos de Unix, que también permite el uso de ACL.

Cuando los usuarios de una red son de escala, se hace necesaria una gestión centralizada. Es la razón de ser de la idea de Dominio. Los entornos servidores Windows permiten implementar la arquitectura cliente-servidor, en lo que se da en llamar Dominio. Un dominio es entonces un conjunto de equipos que comparten un servicio de directorio (BBDD de usuarios, recursos y permisos). El directorio se soporta con controladores de dominio (DC, Domain Controller) y lo forman cuentas de usuario y directivas de seguridad.

Cuando un usuario inicia sesión en un equipo cliente, debe indicar nombre, contraseña y dominio. Un controlador del dominio verificará las credenciales, permitiendo, o no, el acceso. Un servidor en un dominio puede desempeñar 3 roles:

- Controlador de dominio principal (PDC): Servidor SAM. Todo dominio NT tiene que tener un controlador principal de dominio.
- Controlador de reserva (BDC): Es una copia de seguridad del SAM. No es obligatoria su presencia, pero sí muy recomendable. Pueden existir varios controladores de reserva, que se sincronizan con el PDC.
- Servidor miembro: Es un servidor específico. No contiene copias del SAM. Participa en el dominio para ofrecer recursos y servicios.

Si un servidor no pertenece a un dominio entonces es independiente, como pueda ser un servidor web público. El controlador de reserva puede cambiar su función a PDC si éste cae. En Windows 2003 Server no se distingue entre PDC y BDC, se denominan controladores de dominio. Para configurar un servidor comp PDC se ejecuta el comando DCPROMO (Asistente de instalación del directorio activo). Los nombre de dominio siguen la sintaxis del sistema DNS.

AD es una estructura basada en el servicio de directorio LDAP (Lightweight Directory Access Protocol) que almacena información sobre recursos facilitando su acceso. Los componentes del directorio se llaman objetos y se almacenan en contenedores, siguiendo una estructura jerárquica. La información del AD se replica en los controladores de dominio para mantener consistencia. Requiere un servidor DNS dinámico.

Otras estructuras del modelo de dominio Windows son:

- Sitio: Lugar físico de un controlador de dominio. Los clientes tratan de iniciar sesión en controladores de dominio de su mismo sitio para acelerar y optimizar el uso de la red.
- Árbol: Conjunto de dominios en una misma jerarquía DNS.
- Bosque: Conjunto de árboles con distintas jerarquías DNS.

- OU (Unidad Organizativa): Conjunto de recursos agrupados para facilitar su administración.

Por fin, para que los usuarios de un dominio accedan a recursos de otro dominio hay que definir una relación de confianza entre dominios. Un sistema tipo Unix puede participar en una red Microsoft, actuando incluso como controlador de dominio. Para ello hay que instalar y configurar samba en un sistema tipo Unix.

Administración

La administración de servidores Windows se realiza con consolas administrativas del Menú Inicio/Programas.

Destacan las siguientes:

- Usuarios y equipos de AD: Menú que permite gestionar cuentas de usuarios en el dominio.
- Administración de equipos: Para administrar equipos locales o remotos. Permite administrar servicios iniciados, dispositivos, visor de sucesos y recursos compartidos.
- Administrador de servicios de Internet (IIS): Para configurar servidores web, FTP, SMTP y NNTP de Windows.
- DHCP, DNS y WINS: Son las herramientas de administración del servidor DHCP, DNS y el servicio WINS, de resolución de nombres Windows.
- Directivas de seguridad: Herramienta de gestión de los privilegios de seguridad a nivel de dominio, de controlador de dominio y local.
- Dominios y confianzas de AD: En particular, para definir las relaciones de confianza entre dominios.
- Enrutamiento y acceso remoto: Permite definir opciones de enrutamiento, acceso remoto (RAS) y redes privadas virtuales (VPN).
- Rendimiento y visor de sucesos: Es la herramienta que ofrece información gráfica del rendimiento de los componentes del sistema. Permite definir alertas de rendimiento. El visor de sucesos registra la actividad del sistema de ficheros de registro (logs).
- Servicios: Permite iniciar y detener servicios.
- Sistema de archivos distribuidos (DFS): Configura DFS para acceder a recursos compartidos.
- Sitios y servicios de AD: Define los servidores que integran cada sitio y la comunicación entre ellos.

El registro de Windows

El registro de Windows es una BBDD jerárquica centralizada dispuesta para almacenar la información de configuración del sistema para usuarios, aplicaciones y dispositivos hardware. La información del registro es la referencia que usa el SO Windows continuamente, como puedan ser perfiles de usuario, aplicaciones instaladas, tipos de documentos que gestiona cada aplicación, elementos hardware del sistema, etc.

El registro reemplaza a la mayoría de archivos .ini basados en texto usados en versiones anteriores de Windows como autoexec.bat y config.sys. Aunque es común a distintas versiones de Windows, existen diferencias.

Una sección del registro es un grupo de claves, subclaves y valores que cuentan con archivos auxiliares con copias de seguridad de sus datos. Los archivos auxiliares de cada sección excepto HKEY_CURRENT_USER suelen estar en la carpeta %SystemRoot%\System32\Config.

Para la clave HKEY_CURRENT_USER suelen disponerse en %SystemRoot%\Profiles\nombreDeUsuario.

Las extensiones de los archivos de estas carpetas indican el tipo de datos que contienen. A veces, la falta de extensión también puede indicar el tipo de datos que contienen. Los tipos de valores más importantes usados en el registro para almacenar la información se resumen en la siguiente tabla.

Valor	Tipo de valor	Significado
Binario	REG_BINARY	Datos binarios sin formato. La mayoría de información de componentes hardware se almacena en forma de datos binarios y se muestra en formato hexadecimal en el Editor de Registro.
Cadena	REG_SZ	Cadena de texto de longitud fija.
Cadena Múltiple	REG_MULTI_SZ	Valores de listas o valores múltiples. Es el formato de lectura más sencilla. Las entradas se separan por espacios, comas u otros signos.
Cadena Expandible	REG_EXPAND_SZ	Este tipo incluye variables que se resuelven cuando un programa o servicio usa los datos.
DWORD	REG_DWORD	Datos representados por un número de 4B (valor entero de 32 bits). Muchos parámetros de controladores de dispositivos y servicios son de este tipo. Se muestran en formato binario, hexadecimal o decimal.

La siguiente tabla enumera las claves predefinidas que usa el sistema. El tamaño máximo del nombre de una clave es de 255 caracteres.

Carpeta / clave predefinida	Descripción
HKEY_CURRENT_USER	Contiene la raíz de la información de configuración del usuario que inicia sesión como carpetas de usuario, colores de pantalla y configuración del Panel de Control. Esta información se asocia al perfil del usuario. A veces se abrevia como HKCU.
HKEY_USERS	Contiene los perfiles de usuario cargados en el equipo. HKEY_CURRENT_USER es una subclave de HKEY_USERS, que suele abreviarse como HKU.
HKEY_LOCAL_MACHINE	Información de configuración específica del equipo (para cualquier usuario). Se abrevia como HKLM.
HKEY_CLASSES_ROOT	<p>Subclave de HKLM\Software. Su información garantiza que al abrir un archivo se haga con el programa correcto. Se abrevia como HKCR.</p> <p>La clave HKLM\Software\Classes contiene la configuración predeterminada a aplicar a todos los usuarios del equipo local.</p> <p>La clave HKCU\Software\Classes contiene la configuración que se aplica sólo al usuario interactivo.</p> <p>La clave HKCR combina la información de estos dos orígenes. Proporciona una vista combinada de los programas de versiones anteriores de Windows.</p> <p>Para cambiar la configuración de un usuario, se hace en HKCU\Software\Classes en lugar de en HKCR. Para cambiar la configuración predeterminada, se cambia HKLM\Software\Classes. Si se escriben valores en HKCR, el sistema almacena la información en HKLM\Software\Classes.</p> <p>Si se cambian valores de clave en HKCR y la clave ya existe en HKCU\Software\Classes, el sistema almacenará la información ahí, en lugar de en HKLM\Software\Classes.</p>
HKEY_CURRENT_CONFIG	Información del perfil de hardware del equipo local cuando se inicia el sistema.

Comandos DOS

A continuación se muestra un resumen de los comandos de administración DOS habituales.

Comandos DOS	
arp	Muestra o modifica la tabla ARP
ipconfig	Configuración IP en modo texto
ftp / tftp	Cliente FTP / TFTP
hostname	Muestra el nombre del host
finger	Informa de los usuarios que ejecutan el servicio finger
lpq	Muestra el estado de una cola de impresión remota (lpd)
lpr	Envía a imprimir a impresora remota
nbtstat	Muestra estadísticas NetBIOS
nslookup	Consultas a un servicio DNS
ping	Eco ICMP a host remoto
rcp	Copia archivos del equipo local a uno remoto que ejecute el servicio RCP
rexec / rsh	Ejecuta un comando en un host remoto con el servicio REXEC (RSH) habilitado

Comandos NET (Comandos DOS para red)

netstat	Muestra estadísticas de red, conexiones y puertos abiertos
net computer	Agrega o elimina máquinas a un dominio
net start	Inicia un servicio. Sin parámetros, muestra los servicios en ejecución
net accounts	Gestiona parámetros de cuentas como la longitud mínima de la contraseña
net config	Gestiona la configuración de servicios
net print	Gestiona trabajos y colas de impresión
net pause	Suspende la ejecución de un servicio
net view	Muestra el listado de recursos que se están compartiendo en un servidor
net continue	Reanuda un servicio suspendido con net pause
net time	Sincroniza el reloj del sistema con el de un host remoto o dominio
net group	Agrega, muestra o modifica grupos globales
net localgroup	Agrega, muestra o modifica grupos locales
net name	Añade o elimina un alias
tracert	Realiza una traza a un host remoto
net send	Envía mensajes a otros usuarios
net stop	Detiene un servicio
net session	Muestra o desconecta sesiones del equipo. Sin parámetros, muestra las sesiones establecidas actualmente
net file	Desbloquea un archivo para otros usuarios. Sin parámetros, muestra los archivos compartidos abiertos
net use	Conexión a recurso remoto. El recurso remoto se mapea en una unidad de red. Ejemplo: net use X: \\\servidor\recurso
net user	Crea y modifica cuentas de usuario. Sin parámetros, muestra las cuentas del sistema
net share	Gestiona recursos compartidos
net statistics	Estadísticas del servicio local

Sistemas Unix y Linux

Los sistemas informáticos, en origen sólo permitían el proceso por lotes. Por tanto, se hizo necesaria la evolución (década de los 60) a sistema de proceso de tiempo compartido. Así, 21

se permitía la interacción con la máquina. El primer sistema de tiempo compartido fue CTSS (Compatible Time-Sharing System), desarrollado en el MIT.

El MIT, Bell Laboratories y General Electrics diseñaron el SO MULTICS (Multiplexed Information and Computing Service), programado en PL/1. MULTICS sirvió de base a Ken Thompson, para desarrollar otro SO en lenguaje ensamblador, para una máquina PDP-7: UNICS, por oposición a MULTICS, que acabó por llamarse "UNIX" (1970).

Dennis Ritchie y Ken Thompson reescribieron Unix en lenguaje B, para otra máquina, la PDP-11. Así, se evitaba reescribir el código fuente cuando el sistema se migraba, consiguiendo portabilidad. El lenguaje B fue mejorado y evolucionó al lenguaje C, reescribiendo Unix en C, que sigue empleándose y evolucionando.

Unix fue proporcionado por AT&T a universidades, con su código fuente. Esto generó mejoras en el código y su amplia difusión. La universidad de Berkeley incluyó memoria virtual, mejoró el sistema de archivos, incorporó el editor de texto vi, el shell csh y la pila de protocolos TCP/IP.

Sus SO se llamaron BSD. AT&T terminó comercializando Unix, en particular su versión más conocida, Unix System V. Las distintas versiones de Unix generaron problemas de compatibilidad de programas, lo que originó su estandarización. El IEEE abrió el proyecto POSIX (Portable Operating System IX) para dicha tarea, lo que materializó en el estándar IEEE 1003.

Linux es un SO de código libre basado en Unix. Fue desarrollado por Linus Torvalds. Actualmente su desarrollo lo coordina la FSF (Free Software Foundation) y su proyecto GNU. El código libre usa licencias GPL (General Public License), que básicamente obliga a proporcionar el código fuente modificado y mantener la licencia GPL para el software desarrollado a partir de otro software protegido con GPL.

Las versiones de Linux se denominan distribuciones. Sus variaciones se refieren a aspectos como el GUI, la instalación, etc. El núcleo del sistema es común. Ejemplos son Ubuntu, Red Hat, Fedora, Suse, etc.

El proyecto GNU define software libre como el que dispone de libertad de ejecución, de modificación, de distribución y mejora. Si se incluye código GPL en un proyecto, todo el código pasará a ser libre. Esta condición evita que el software comercial use código GPL. La aplicación estricta de esta licencia generaría problemas con las bibliotecas del compilador, para programas comerciales. Para evitarlo, se dispone la licencia GNU LGPL (Lesser GPL), menos restrictiva que GPL, que permite integrar partes LGPL sin que todo el código pase a ser software libre.

Los entornos Unix/Linux permiten al usuario elegir el intérprete de comandos (shell). Las Shell difieren en la definición de instrucciones y la programación de scripts. Destacan entre las comunes bsh (Bourne Shell, /bin/sh), csh (C-Shell, /bin/csh, basada en C), ksh (Korn Shell, /bin/ksh), bash (Bourne Again Shell, /bin/bash, mejora csh y ksh) o tcsh (Tab C-Shell, /bin/tcsh, una mejora de csh).

Sistema de archivos

En entornos Unix, el sistema de archivos sigue el estándar de jerarquía de ficheros (FHS, Filesystem Hierarchy Standard, 1993). FHS define la estructura de directorios y sus contenidos. Comenzó en 1994 con el FSSTND (Filesystem Standard), que ha sufrido varias revisiones hasta el actual FHS (1996). FHS es mantenido por el Free Standards Group, una organización constituida por compañías como Hewlett Packard, Dell, IBM o Red Hat. La mayoría de las distribuciones Linux, no lo aplican de forma estricta.

La estructura de directorios es tal que todos los ficheros y directorios aparecen bajo el directorio raíz (/), aun si están almacenados en dispositivos físicos diferentes.

La estructura es la siguiente:

Directorio	Descripción
/bin	Comandos binarios esenciales (cp, mv, ls, rm, etc)
/boot	Ficheros de arranque del sistema (núcleo y discos RAM)
/dev	Dispositivos esenciales
/etc	Ficheros de configuración del sistema, específicos del anfitrión
/home	(opcional) Directorios de inicio de los usuarios
/lib	Bibliotecas compartidas esenciales para los binarios de /bin/, /sbin/ y el kernel
/mnt	Sistemas de ficheros montados temporalmente
/media	Puntos de montaje para dispositivos de medios como unidades lectores de CD
/opt	Paquetes de aplicaciones estáticas
/proc	Sistema de ficheros virtual que documenta sucesos y estados del núcleo
/root	(opcional) Directorio de inicio del usuario root (superUsuario)
/sbin	Binarios de administración de sistema
/tmp	Ficheros temporales
/srv	Datos específicos de sitio servidos por el sistema
/usr	Jerarquía secundaria para datos compartidos de solo lectura. Debe poder ser compartido para múltiples anfitriones y no debe contener datos específicos del anfitrión que los comparte

Para instalar un sistema Linux, suele ser habitual la recomendación de usar al menos tres particiones: /, /boot y swap (a la que no se asigna punto de montaje).

Archivos, permisos e inodos

Los archivos de un sistema de archivos Unix distinguen los tipos ordinarios (datos o programas), directorios (lista de archivos con punteros a sus inodos), especiales (dispositivos tales como puertos y discos) y tuberías con nombre (named pipes, comunican dos procesos).

Los nombres de archivos pueden tener hasta 255 caracteres. Se pueden crear archivos ocultos con un punto como primer carácter del nombre. Un enlace o vínculo (link) permite que un mismo archivo pueda llamarse desde varios directorios).

Sólo existirá una copia del archivo, aunque podrá accederse desde varios directorios. Si se borra el archivo en un directorio sólo se borra el enlace. El archivo sólo se borra si se borra en todos los directorios en que posee enlace. En general los enlaces se refieren a enlaces duros (físicos o hard link), en oposición a los enlaces simbólicos, archivos que apuntan a otro (similar a un acceso directo en Windows).

Todos los directorios y subdirectorios se tratan como archivos. El directorio actual se nota con un punto y el directorio padre con dos.

En entornos Unix se definen 3 tipos de permisos básicos:

- lectura (r)
- escritura (w)
- ejecución (x)

Se definen tres perfiles de usuario:

- el propietario del archivo (user, u)
- usuario del grupo del propietario (group, g)
- usuario que no pertenece al grupo del usuario (other, o)

Los permisos se suelen representar con 10 bits: -rwxrwxrwx. El primer carácter corresponde al tipo de fichero ('-', fichero ordinario; 'd', directorio; 'c', fichero especial tipo carácter; 'b', fichero especial tipo bloque; ...). Dependiendo del tipo de Unix hay otras opciones ('l', 's', '='). El resto de caracteres, en bloques de tres, especifican qué tipo de usuario puede realizar qué operación.

Por ejemplo, la respuesta -rwxr-x-r- indica un fichero ordinario, los tres bits del propietario (rwx), le dan permiso de lectura, escritura y ejecución, los tres segundos (r-x), del grupo, le permiten leer y ejecutar el fichero y los tres últimos (r-) permiten al resto, solo lectura.

A parte de los anteriores bits de permisos, hay un cuarto tipo más especial, en el que se engloban setuid, setgid y sticky bit. Setuid o modo "s", significa que la identidad efectiva de usuario con la que se ejecuta el programa es la del propietario. Este permiso no tiene sentido en ficheros no ejecutables.

Por ejemplo, la salida -rwsr-x— 1 usuario 1499 Jun 6 10:17 fichero, indica un fichero modo s. Setgid, o modo s del grupo es similar al anterior, referido al grupo. Ejemplo, -r-xr-sr-x 1 usuario grupo 9984 Jul 16 1994 fichero. Por fin, el sticky bit, o modo "t", cuando está activado indica que el fichero nunca se elimina del área de swap. Suele ser útil para programas ejecutados a menudo y por diferentes usuarios. Sobre un directorio el comportamiento es distinto, permitiendo que sólo el propietario del fichero, el propietario del directorio o el superusuario "root", puedan renombrar o borrar los ficheros contenidos en él. Es útil para áreas compartidas. Por ejemplo, en el directorio /tmp al hacer ls -ald, se puede obtener la salida drwxrwxrwt 5 root 309 Jun 7 11: 41 ./.

Los permisos suelen indicarse también mediante un código octal de 3 números. Cada número codifica una terna. Así, la terna rw- se codificaría como 110 en binario, que en octal es 6.

En entornos Unix los archivos se gestionan con nodos índice o inodos, estructuras de 64B con información del tipo de archivo y permisos, número de referencias del archivo en directorios (enlaces), identificador del propietario y su grupo, tamaño del archivo en Bytes, fecha de último acceso y modificación del archivo e inodo y dirección, formada por 39B divididos en 13 punteros de 3B.

En la dirección, los 10B primeros son directos. Contienen direcciones de bloques de datos. Los 3B siguientes son punteros indirectos: indirecto simple (puntero a bloque de 256 punteros directos, 256 bloques), indirecto doble (puntero a bloque de 256 punteros indirectos simples, $2e8 \cdot 2e8 = 2e16$ bloques) o indirecto simple (puntero a bloque de 256 punteros indirectos dobles, 16 millones de bloques).

La asignación de bloques a un archivo es dinámica, según necesidad. Por eso, los bloques pueden no asignarse secuencialmente, generando fragmentación, lo que perjudica el rendimiento. En Unix System V se usan bloques de 1KB (en FAT pueden llegar a 32KB). Los bloques pequeños evitan desaprovechar espacio en disco. Los punteros de dirección de un archivo se van usando a medida que se necesitan, comenzando por los punteros directos (más rápidos).

Suponiendo bloques de 1KB las capacidades que ofrece cada sistema es, para direccionamiento directo 10KB (10 bloques); indirecto simple 256KB (256 bloques); indirecto doble 65MB (65536 bloques); indirecto triple 16GB ($2e8 \cdot 2e16 = 2e24$ bloques). Así, el tamaño máximo teórico de un archivo sería la suma de la capacidad de un inodo total (los 16GB aprox.). Las ventajas de los inodos es que al ser pequeños pueden mantenerse en memoria, ofreciendo un rápido acceso y aprovechan mejor el espacio en disco.

En los entornos Unix tradicionales, la organización de un disco cuenta con un Bloque de arranque (bloque 0), con información para el arranque del SO; un Superbloque (bloque 1), con información de la organización del sistema de archivos; Inodos, tabla de inodos con la estructura expuesta dividida en Inodo 1 (reservado para gestión de bloques defectuosos) e Inodo 2 (gestionado por el directorio raíz) y Bloque de datos, que guarda el contenido de los archivos.

Gestores de arranque, entornos de escritorio y editores de texto

Un gestor de arranque es un programa que se instala en el sector de arranque del disco duro (MBR, Master Boot Record) y al encender la máquina, permite elegir el SO que ejecutar. Los más populares en entornos Unix son LILO (LInux LOader) y GRUB (GRand Unified Bootloader). El de Microsoft se llama NT Loader (NTLDR) instalado en el sector de arranque de la partición primaria de windows. Los dos tipos pueden convivir ya que se instalan en lugares distintos.

En cuanto a las GUI, en entornos Unix existen algunas típicas, como KDE, que usa Konqueror como gestor de archivos y navegador web; GNOME, que usa Nautilus como gestor de archivos, pudiéndose usar otro programa como navegador web y Xfce, que usa Thunar como gestor de archivos, típico en distribuciones de BSD y Solaris.

Debido a su importancia, a la hora de editar archivos de texto para la configuración, los entornos Unix suelen incluir editores de texto típicos como vi, emacs (de más fácil manejo), vim o xemacs.

Cuentas de usuario

Para acceder a un sistema tipo Unix, un usuario se modela con una cuenta del sistema. Un usuario tendrá además un perfil, que definirá, entre otras cosas, sus privilegios. Los entornos Unix definen un perfil de usuario de máximo privilegio llamado root o superusuario, y por añadidura, su cuenta se denomina cuenta de root. Sus privilegios le permiten realizar cualquier tarea administrativa.

Una cuenta de usuario se define con un nombre de usuario o login, un identificador (UID, User IDentifier), un identificador de su grupo (GID), una contraseña (password), su Shell (el CLI que ejecutará por defecto), su directorio particular (conocido como /home) y comentarios.

La información de una cuenta de usuario se almacena básicamente en tres archivos: passwd, shadow y group del directorio /etc. El primero posee el listado completo de usuarios con información para cada uno de su login, contraseña, uid, gid, comentario, directorio home y shell. El campo contraseña suele aparecer con "x", lo que evita que se pueda ver directamente, porque se encripta en el archivo shadow.

El archivo shadow contiene las contraseñas encriptadas de los usuarios. Para cada usuario se almacena su login, contraseña encriptada, fecha de última modificación, mínimo y máximo de días entre modificaciones, días de aviso de expiración, máximo de días con la cuenta inactiva y fecha de expiración.

El archivo group lista los grupos de usuarios. Para cada grupo se tiene la información del nombre, contraseña, gid y lista de usuarios. La contraseña no se usa. Suele presentarse un asterisco o un espacio en blanco. La lista de usuario sindica los UID de los miembros secundarios del grupo. Como un usuario puede pertenecer a varios grupos, su grupo principal se indica en el archivo passwd y los secundarios se indican incluyendo su UID en la lista de usuario de cada grupo en el archivo group.

Comandos UNIX

A continuación se muestra un resumen de los comandos de administración UNIX habituales.

Comandos UNIX. Gestión Archivos

ls	Muestra los archivos de un directorio
cd	Cambia el directorio actual
chmod	Cambia los permisos de un archivo
chown	Cambia el propietario de un archivo
find	Busca archivos con un cierto nombre
pwd	Muestra el directorio actual
cat	Muestra contenido de un archivo de texto
unmask	Establece la máscara de permisos
mount	Monta una unidad en un directorio
tar	Empaquetá un árbol de directorios en un archivo

Comandos UNIX. Gestión de Procesos

jobs	Muestra todos los trabajos
fg	Pone un trabajo como trabajo principal
stop	Detiene (no finaliza) la ejecución de un trabajo. Puede reanudarse con fg o bg
top	Muestra el estado de los procesos de forma dinámica, ordenados en cada instante por uso de CPU
kill	Finaliza un trabajo o proceso
bg	Lleva un trabajo a segundo plano
ps	Muestra el listado de procesos en ejecución
nice / renice	Cambian la prioridad de un proceso

Comandos UNIX. Gestión de Red

rlogin	Inicio de sesión remoto
rcp	Copia remota
rsh	Shell remoto
ftp / tftp	Servicio FTP / TFTP
ping	Eco ICMP
netstat	Estadísticas de red
ipconfig	Configuración de red
rwho	Usuarios conectados a máquinas de red

Comandos UNIX. Herramientas

grep	Busca un patrón en uno o más archivos
sort	Ordena las líneas de un archivo
uniq	Elimina líneas repetidas
de / be	Calculadoras
od	Muestra archivo byte a byte incluyendo caracteres no imprimibles
cmp / diff	Comparan 2 archivos
tee	Descompone la entrada en 2 flujos de salida, una estándar y otra, un archivo
date	Muestra o cambia la fecha y hora del sistema

Comandos UNIX. Administración de Sistema	
su	Cambia de usuario temporalmente
passwd	Cambio de contraseña
useradd	Añade un usuario
usermod	Modifica un usuario
userdel	Elimina un usuario
groupadd	Añade un grupo
groupmod	Modifica un grupo
groupdel	Elimina un grupo
logout / exit	Cierre de sesión
shutdown / halt / reboot	Apaga el equipo

Sistemas Operativos para dispositivos móviles

Un SO móvil es un conjunto de programas de bajo nivel que permite la abstracción de las peculiaridades del hardware específico del teléfono móvil y provee servicios a las aplicaciones móviles, que se ejecutan sobre él.

Capas de un Sistema Operativo móvil

- **Kernel** . El núcleo o kernel proporciona el acceso a los distintos elementos del hardware del dispositivo. Ofrece distintos servicios a las capas superiores como son los controladores o drivers para el hardware, la gestión de procesos, el sistema de archivos y el acceso y gestión de la memoria.
- **Middleware** . El middleware es un conjunto de módulos que hacen posible la propia existencia de aplicaciones para móviles. Es totalmente transparente para el usuario y ofrece servicios claves como el motor de mensajería y comunicaciones, codecs multimedia, intérpretes de páginas web, gestión del dispositivo y seguridad.
- **Entorno de ejecución de aplicaciones** . El entorno de ejecución de aplicaciones consiste en un gestor de aplicaciones y un conjunto de interfaces programables abiertas y programables por parte de los desarrolladores para la creación de software.
- **Interfaz de usuario** . Las interfaces de usuario facilitan la interacción con el usuario y el diseño de la presentación visual de la aplicación. Los servicios que incluye son el de componentes gráficos (botones, pantallas, listas, ...) y el marco de interacción.

Aparte de estas capas también existe una familia de aplicaciones nativas del teléfono que suelen incluir los menús, el marcador de números de teléfono, ...

Sistemas Operativos móviles

Android

Android Inc. es la empresa que creó el SO móvil. Se fundó en 2003 y fue comprada por Google en 2005 y en 2007 fue lanzado al mercado. Su nombre se debe a su inventor, Andy Rubin. Originalmente era un sistema pensado para las cámaras digitales.

Android está basado en Linux, disponiendo de un Kernel en este sistema y utilizando una máquina virtual sobre este Kernel que es la responsable de convertir el código escrito en Java a código capaz de comprender el Kernel.

Una de las grandes cualidades o características de este SO es su carácter abierto. Android se distribuye bajo dos tipos de licencias, una que abarca todo el código del Kernel y que es GNU GPLv2 (implica que su código se debe poner al alcance de todos y que todos podremos hacer con este código lo que nos parezca oportuno, modificarlo, ampliarlo,

recortarlo, pero siempre estaremos en la obligación de volver a licenciarlo con la misma licencia). Google también tiene otra licencia para el resto de componentes del sistema que se licencia bajo APACHE v2, una licencia libre y de código abierto (implica que este código se pueda distribuir para ser modificado y usado a antojo del que lo utilice, pero a diferencia del primer caso, las modificaciones y el código resultante no es obligatorio licenciarlo bajo las mismas condiciones en las que se encontraba).

La estructura del SO Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina Virtual Dalvik con compilación en tiempo de ejecución hasta la versión 5.0, luego cambió al entorno Android Runtime (ART).

Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic.

Las características y especificaciones son:

Características	
Diseño de dispositivo	La plataforma es adaptable a pantallas de mayor resolución, VG, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales
Almacenamiento	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos
Conectividad	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WIMAX, GPRS, UMTS y HSDPA+
Mensajería	SMS y MMS son formas de mensajería, incluyendo mensajería de texto, además del servicio de Firebase Cloud Messaging (FCM) siendo la nueva versión de Google Cloud Messaging (GCM) bajo la marca Firebase con los nuevos SDK para realizar el desarrollo de mensajería en la nube mucho más sencillo
Navegador web	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor Javascript V8 de Google Chrome. El navegador por defecto de Ice Cream Sandwich obtiene una puntuación de 100/100 en el test Acid3
Soporte de Java	Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y se ejecuta en la Máquina Virtual Dalvik, Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. A partir de la versión 5.0, se utiliza el Android Runtime (ART). El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner
Soporte multimedia	Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP
Soporte para streaming	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player
Soporte para hardware adicional	Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D
Entorno de desarrollo	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. Inicialmente el entorno de desarrollo integrado (IDE) utilizado era Eclipse con el plugin de Herramientas de Desarrollo de Android (ADT). Ahora se considera como entorno oficial Android Studio, descargable desde la página oficial de desarrolladores de Android
Google Play	Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC

Multi-táctil	Android tiene soporte nativo para pantallas capacitivas con soporte multitáctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). Más tarde, Google publicó una actualización que activa el soporte multitáctil de forma nativa
Bluetooth	El soporte para A2DP y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.
Videollamada	Android soporta videollamada a través de Hangouts (antiguo Google Talk) desde su versión HoneyComb
Multitarea	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj
Características basadas en voz	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema
Tethering	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo PdaNet). Para permitir a un PC usar la conexión de datos del móvil Android se podría requerir la instalación de software adicional

Información General

Información General	
Parte de la familia	Linux
Desarrollador	Google (Open Handset Alliance)
Modelo de desarrollo	Código Abierto
Lanzamiento inicial	23 de septiembre de 2008
Tipo de mercado	<ul style="list-style-type: none"> • Teléfonos inteligentes • Tabletas • Android TV • Android Auto • Android Wear
Escrito en	Java (UI), C (núcleo), C++
Núcleo	Núcleo de Linux modificado
Tipo de núcleo	Monolítico
Interfaz gráfica predeterminada	Material Design
Plataformas soportadas	32 y 64 bits ARM, x64, x86, MIPS y MIPS64
Sistema de gestión de paquetes	Google Play, APK y alternativas como F-Droid
Método de actualización	<ul style="list-style-type: none"> • OTA • Play Store
Licencia	Apache 2.0 y GNU GPL 2
Estado actual	En permanente desarrollo
Idiomas	Multilingüe

Arquitectura

Los componentes principales del SO Android son:

- **Aplicaciones** : las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

- **Marco de trabajo de aplicaciones** : los desarrolladores tienen acceso completo a las mismas API del entorno de trabajo usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas** : Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android. Algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android** : Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecutaba hasta la versión 5.0 archivos en el formato de ejecutable Dalvik (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato .dex por la herramienta incluida dx. Desde la versión 5.0 utiliza el ART, que compila totalmente al momento de la instalación de la aplicación.
- **Núcleo Linux** : Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

Versiones

Letra	Nombre	Versión
A	Apple Pie	1.0
B	Banana Bread	1.1
C	Cupcake	1.5
D	Donut	1.6
E	Éclair	2.0 / 2.1
F	Froyo	2.2
G	Gingerbread	2.3
H	Honeycomb	3.0 / 3.1 / 3.2
I	Ice Cream Sandwich	4.0
J	Jelly Bean	4.1 / 4.2 / 4.3
K	KitKat	4.4
L	Lollipop	5.0 / 5.1
M	Marshmallow	6.0 / 6.0.1
N	Nougat	7.0 / 7.1 / 7.1.2
O	Oreo	8.0

Honeycomb fue la primera actualización exclusiva para TV y Tablet, no era apta para móviles.

iOS

iOS es un SO que da vida a dispositivos como el iPhone, el iPad, el iPod Touch o el Apple TV.

Anteriormente denominado iPhone OS creado por Apple originalmente para iPhone, siendo después usado en el iPod Touch e iPad. Es un derivado de Mac OS X que a su vez está basado en Darwin BSD y por lo tanto es un SO tipo Unix, se lanzó en el año 2007.

iOS cuenta con cuatro capas de abstracción:

1. la capa del núcleo del SO
2. la capa de "Servicios Principales"
3. la capa de "Medios"
4. la capa de "Cocoa Touch"

Información General

Información General	
Parte de la familia	BSD
Desarrollador	Apple Inc.
Modelo de desarrollo	Software propietario
Lanzamiento inicial	29 de junio de 2007
Escrito en	C, C++, Objective-C, Swift
Núcleo	XNU
Tipo de núcleo	Núcleo híbrido (XNU)
Interfaz gráfica predeterminada	Cocoa Touch (Multitáctil, GUI)
Plataformas soportadas	ARM (iPad, iPhone y iPod Touch)
Método de actualización	Mediante iTunes. A partir de iOS 6 se puede actualizar desde OTA
Licencia	APSL Apple EULA
Estado actual	En permanente desarrollo
Idiomas	Multilingüe

Actualmente va por la versión iOS 11.

Windows Phone

Windows Phone (abreviado WP) es un SO móvil desarrollado por Microsoft, como sucesor de Windows Mobile. Con Windows Phone; Microsoft ofrece una nueva interfaz de usuario que integra varios de sus servicios propios como OneDrive, Skype y Xbox Live en el SO. Microsoft pasa a enfocarse en un único sistema denominado Windows 10 Mobile, disponible para todo tipo de plataformas (teléfonos inteligentes, tabletas y computadoras). Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente.

Información General

Información General	
Parte de la familia	Windows
Desarrollador	Microsoft
Modelo de desarrollo	Software privativo
Lanzamiento inicial	8 de Noviembre de 2010
Escrito en	C, C++
Plataformas soportadas	Teléfonos inteligentes
Método de actualización	Computadora (WP7), teléfono (WP8)
Licencia	Microsoft (EULA)
Estado actual	Descontinuado
Idiomas	Multilingüe
Predecesor	Windows Mobile, Zune, Kin
Sucesor	Windows 10 Mobile

Versiones

- Windows Phone 7
- Windows Phone 8
- Windows Phone 8.1

BlackBerry 6

El BlackBerry OS es un SO móvil de código cerrado desarrollado por BlackBerry, antigua Research In Motion (RIM).

El sistema permite multitarea y tiene soporte para diferentes métodos de entrada adoptados por RIM para su uso en computadoras de mano. Su desarrollo se remonta a la aparición de los primeros handleds en 1999.

Información General

Información General	
Parte de la familia	SO para teléfonos inteligentes
Desarrollador	BlackBerry
Modelo de desarrollo	Código cerrado
Lanzamiento inicial	1999
Escrito en	Java, C++
Tipo de núcleo	Máquina Virtual Java
Interfaz gráfica predeterminada	GUI
Plataformas soportadas	Teléfonos inteligentes de BlackBerry
Sistema de gestión de paquetes	BlackBerry Desktop Manager
Método de actualización	Desde el teléfono y/o computadora
Licencia	Propietaria
Estado actual	Descontinuado
Idiomas	Multilingüe

Symbian

Fue producto de la alianza de varias empresas de telefonía móvil, entre la que se encuentra Nokia como la más importante.

El SO Symbian es una colección compacta de código ejecutable y varios archivos, la mayoría de ellos son bibliotecas vinculadas dinámicamente (DLL) y otros datos

requeridos, incluyendo archivos de configuración, de imágenes y de tipografía, entre otros recursos residentes. Symbian se almacena, generalmente, en un circuito flash dentro del dispositivo móvil.

Información General

Información General	
Desarrollador	Accenture previamente por Symbian Ltd y Fundación Symbian
Modelo de desarrollo	Software propietario
Lanzamiento inicial	1997
Tipo de mercado	Teléfonos móviles
Núcleo	EKA2
Tipo de núcleo	Micronúcleo
Interfaz gráfica predeterminada	S60
Plataformas soportadas	ARM, x86
Sistema de gestión de paquetes	Symbian Nokia Packet Service
Método de actualización	Ninguno
Licencia	EPL
Estado actual	Descontinuado
Idiomas	Multilingüe

Firefox OS

Firefox OS es un SO móvil, basado en HTML5 con núcleo Linux, para smartphones y tabletas. Es desarrollado por Mozilla Corporation bajo el apoyo de empresas y voluntarios de todo el mundo. Este SO está enfocado especialmente en los dispositivos móviles. Está diseñado para permitir a las aplicaciones HTML5 comunicarse directamente con el hardware del dispositivo usando Javascript y Open Web APIs.

Información General

Información General	
Parte de la familia	Linux
Desarrollador	Mozilla Corporation
Modelo de desarrollo	Código abierto
Lanzamiento inicial	23 de abril de 2013
Tipo de mercado	Genérico (inicialmente smartphones y portátiles)
Escrito en	HTML, CSS, JavaScript, C++
Núcleo	Linux
Interfaz gráfica predeterminada	Gaia (IGU)
Plataformas soportadas	ARM (smartphones)
Sistema de gestión de paquetes	Firefox Marketplace (usando manifiestos)
Método de actualización	OTA o imagen ROM por separado
Licencia	MPL y otras
Estado actual	Descontinuado
Idiomas	Multilingüe

Ubuntu Touch

Ubuntu Touch es un SO móvil basado en Linux. Se caracteriza por ser un sistema diseñado para plataformas móviles.

Información general

Información General	
Parte de la familia	Linux
Desarrollador	Canonical Ltd.
Modelo de desarrollo	FOSS
Lanzamiento inicial	02 de enero de 2013
Tipo de mercado	Virtual o físico
Escrito en	HTML5, C, C++, QML
Núcleo	Linux
Tipo de núcleo	Monolítico
Interfaz gráfica predeterminada	Unity 8
Plataformas soportadas	Smartphones y Tablets
Licencia	GPL
Estado actual	Versión estable
Idiomas	Multilingüe

Listado Sistemas Operativos Móviles

Android
Bada
BlackBerry
Firefox OS
iOS
Linux móvil
Maemo
MeeGo
Nokia Asha
Palm OS
QNX
Replicant
Sailfish OS
Series 40
Symbian
Tizen
Ubuntu Edge
Ubuntu Touch
Uhuru Mobile
WebOS
Windows CE
Windows Mobile
Windows Phone

INTERNA. Características técnicas de los lenguajes y paradigmas actuales de programación.

Lenguajes y paradigmas actuales de programación.

Categorías de Lenguajes de Programación

Los lenguajes de programación se pueden clasificar atendiendo a varios criterios:

- Según el nivel de abstracción
- Según el paradigma de programación que posee cada uno de ellos
- Según su campo de aplicación
- Según su traducción

Según su nivel de abstracción

Lenguajes Máquina

Están escritos en lenguajes directamente legibles por la máquina (computadora), ya que sus instrucciones son cadenas binarias (0 y 1). Da la posibilidad de cargar (transferir un programa a la memoria) sin necesidad de traducción posterior lo que supone una velocidad de ejecución superior, solo que con poca fiabilidad y dificultad de verificar y poner a punto los programas.

Lenguajes de bajo nivel

Los lenguajes de bajo nivel son lenguajes de programación que se acercan al funcionamiento de una computadora. El lenguaje por excelencia es el lenguaje ensamblador, éste trabaja con los registros de memoria de la computadora de forma directa.

La principal utilización de este tipo de lenguajes es para programar los microprocesadores, utilizando el lenguaje ensamblador correspondiente a dicho procesador.

Lenguajes de medio nivel

Hay lenguajes de programación que son considerados como lenguajes de medio nivel (como es el caso del C) al tener ciertas características que los acercan a los lenguajes de bajo nivel pero teniendo, al mismo tiempo, ciertas cualidades que lo hacen un lenguaje más cercano al humano y, por tanto, de alto nivel.

Lenguajes de alto nivel

Los lenguajes de alto nivel son normalmente fáciles de aprender porque están formados por elementos de lenguajes naturales, como el inglés. Esta forma de trabajar puede dar la sensación de que las computadoras parecen comprender un lenguaje natural; en realidad lo hacen de una forma rígida y sistemática, sin que haya cabida, por ejemplo, para ambigüedades o dobles sentidos.

Lenguajes utilizados Pascal, Basic, ...

Según el paradigma de programación

Un **paradigma de programación** indica un método de realizar cómputos y la manera en que se deben estructurar y organizar las tareas que debe llevar a cabo un programa.

Un paradigma es un modelo que, a su vez, es una representación abstracta de la realidad.

Un paradigma de programación es un modelo de programación que representa un estilo o forma de programar o construir programas para realizar ciertas tareas o actividades. Cada modelo tiene sus propias estructuras y reglas de construcción. El modelo de programación por emplear depende del problema que se desee solucionar.

Los paradigmas fundamentales están asociados a determinados modelos de cómputo. También se asocian a un determinado estilo de programación. Los lenguajes de programación suelen interpretar, a menudo de forma parcial, varios paradigmas.

Existen muchos paradigmas de programación diferentes, cada uno de ellos tiene sus **propias características** y tratan de solucionar los problemas clásicos del desarrollo de software desde diferentes perspectivas y filosofías.

Los paradigmas de programación solo son **propuestas tecnológicas** adoptadas por la Comunidad de desarrolladores que se enfocan a resolver uno o varios problemas definidos y delimitados. Existen muchos paradigmas de programación diferentes, posiblemente el más ampliamente utilizado hoy en día sea el de la **programación orientada a objetos**.

Algunos lenguajes de programación pueden soportar **múltiples paradigmas** de programación. Por ejemplo, C++ puede ser empleado para desarrollar software utilizando para ello un modelo de programación puramente orientado a objetos o bien puramente estructurado.

Otros lenguajes han sido diseñados para soportar un **único paradigma** de programación, ese es el caso de **Smalltalk** que soporta únicamente la programación orientada a objetos o **Haskell** que solo soporta la **programación funcional**.

Es común el diseño de lenguajes que soporten múltiples paradigmas de programación. Estos lenguajes son aquellos que soportan al menos dos paradigmas:

- **Scala** : Imperativo, orientado a objetos, funcional, genérico y concurrente
- **Erlang** : Funcional, orientado a objetos y funcional
- **Perl** : Imperativo, orientado a objetos y funcional
- **PHP** : Imperativo, orientado a objetos, funcional y reflexivo
- **JavaScript** : Imperativo, orientado a objetos (prototipos) y funcional
- **Java** : Imperativo, orientado a objetos, reflexivo y genérico
- **Python y Ruby** : Imperativo, orientado a objetos, reflexivo y funcional
- **C++** : Imperativo, orientado a objetos, funcional y genérico
- **C#** : Imperativo, orientado a objetos, funcional (lambda), reflexivo y genérico
- **Lisp** : Orientado a objetos, funcional y declarativo
- **Prolog** : Lógico y declarativo

Estos son algunos ejemplos, existen lenguajes como **Oz** que soporta nueve paradigmas de programación.

Paradigmas de programación

Un paradigma define un conjunto de reglas, patrones y estilos de programación que son usados por un grupo de lenguajes de programación.

Cada lenguaje tiene sintaxis y semántica:

- La sintaxis de un lenguaje de programación está relacionada con la forma de los programas, por ejemplo, las expresiones, comandos, declaraciones, etc. son puestos juntos en un programa.
- La semántica de un lenguaje de programación está relacionada con el significado de los programas, por ejemplo, cómo se comportarán cuando se ejecutan en una computadora.

La sintaxis de un lenguaje influye en cómo los programas son escritos por el programador, leídos por otro programador y traducidos por el computador. La semántica de un lenguaje determina como los programas son compuestos por el programador, entendidos por otros programadores e interpretados por el computador.

Tipos de Paradigmas:

- Paradigma imperativo (por procedimientos)
- Paradigma declarativo
 - Programación funcional
 - Programación lógica
 - Programación Reactiva (Dataflow)
- Paradigma orientado a objetos

Paradigma imperativo (por procedimientos)

En el paradigma por procedimientos, los programas se desarrollan a través de procedimientos. Pascal, C y BASIC son tres de los lenguajes imperativos más importantes. El paradigma se inició a principios de los años 50 cuando los diseñadores reconocieron que las variables y los comandos o instrucciones de asignación constituyan una simple pero útil abstracción del acceso a memoria y actualización del conjunto de instrucciones máquina.

- Describe cómo debe realizarse el cálculo, no el porqué
- Un cómputo consiste en una serie de sentencias, ejecutadas según un control de flujo explícito, que modifican el estado del programa
- Las variables son celdas de memoria que contienen datos (o referencias), pueden ser modificadas y representan el estado del programa
- La sentencia principal es la asignación
- Definición de procedimientos
- Definición de tipos de datos
- Chequeo de tipos en tiempo de compilación
- Cambio de estado de variables
- Pasos de ejecución de un proceso
- Asociados al paradigma imperativo se encuentran los **paradigmas procedural , modular** y la **programación estructurada**
- Lenguajes: FORTRAN-77, COBOL, BASIC, PASCAL, C, ADA, ...
- También lo implementan: Java, C++, C#, Eiffel, Python, ...

La programación imperativa es una forma de escribir programas secuenciales; es decir, que tienes que ir indicando en el programa los pasos o tareas que debe realizar según las siguientes reglas:

1. El programa tiene un diseño modular.

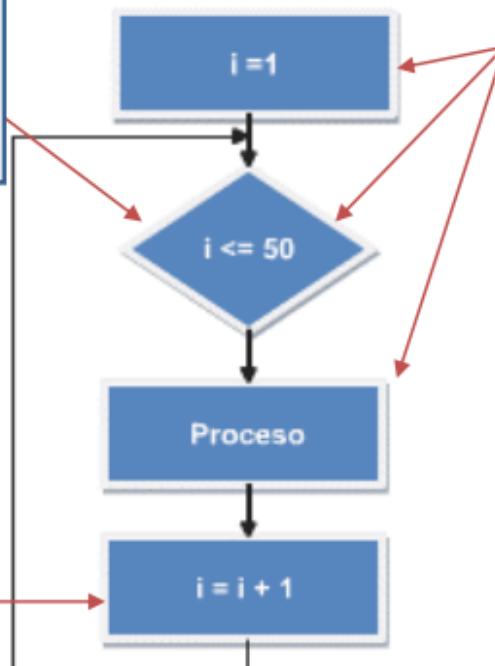
2. Los módulos son diseñados de manera que un problema complejo se divide en problemas más simples.

3. Cada módulo se codifica utilizando las tres estructuras de control básicas: secuencia, selección y repetición.

Estructura de selección. Consiste en una decisión que tiene dos alternativas: verdadero o falso.

Estructura secuencial. Es una instrucción que va seguida de otra.

Estructura de repetición. Es un ciclo en donde se ejecuta una o más instrucciones dependiendo de una condición.



Paradigma declarativo

El paradigma declarativo o paradigma de programación lógica se basa en el hecho de que un programa implementa una relación antes que una correspondencia. Debido a que las relaciones son mas generales que las correspondencias (identificador - dirección de memoria), la programación lógica es potencialmente de más alto nivel que la programación funcional o la imperativa. El lenguaje más popular es el lenguaje PROLOG.

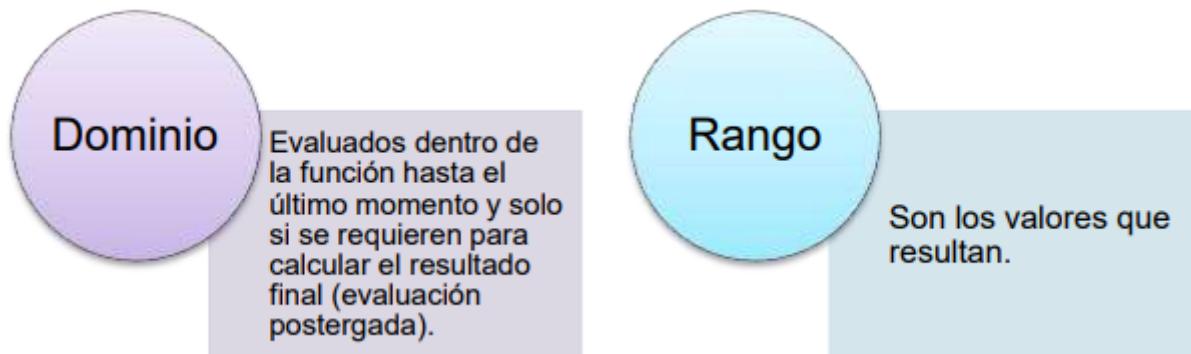
- Describe qué se debe calcular, sin explicar el cómo
- No existe un orden de evaluación prefijado
- Las variables son nombres asociados a definiciones, y una vez instanciadas son inmutables
- No existe sentencia de asignación
- El control de flujo suele estar asociado a la composición funcional, la recursividad y/o técnicas de reescritura y unificación
- Existen distintos grados de pureza en las variantes del paradigma
- Las principales variantes son los paradigmas **funcional**, **lógico**, la **programación reactiva** y los **lenguajes descriptivos**

Programación funcional

La programación funcional se caracteriza por el uso de expresiones y funciones. Un programa dentro del paradigma funcional, es una función o un grupo de funciones compuestas por funciones más simples estableciéndose que una función puede llamar a otra, o el resultado de una función puede ser usado como argumento de otra función. El lenguaje por excelencia es el LISP.

- Basado en los modelos de cómputo **cálculo lambda** (Lisp, Scheme) y **lógica combinatoria** (familia ML, Haskell)
- Las funciones son elementos de primer orden
- Evaluación por reducción funcional. Técnicas:
 - recursividad
 - parámetros acumuladores
 - CPS
 - Mónadas
- Familia LISP (Common-Lisp, Scheme):
 - Basados en s-expresiones
 - Tipado débil
 - Meta-programación
- Familia ML (Miranda, Haskell, Scala):
 - Sistema estricto de tipos (tipado algebraico)
 - Concordancia de patrones
 - Transparencia referencial
 - Evaluación perezosa (estructuras de datos infinitas)
- La computación se realiza mediante la evaluación de expresiones
- Definición de funciones
- Funciones como datos primitivos
- Valores sin efectos laterales, no existe la asignación
- Programación declarativa
- Lenguajes: LISP, Scheme, Haskell, Scala, ...

Las funciones matemáticas son una correspondencia entre un dominio y un rango.



Una definición de función especifica el dominio y el rango, de manera implícita o explícita, junto con una expresión que describe la correspondencia.

Las funciones son aplicadas a un elemento del dominio y devuelven uno del rango.

Programación lógica

- Basado en la **lógica de predicados** de primer orden
- Los programas se componen de hechos, predicados y relaciones
- Evaluación basada en resolución SLD: unificación + backtracking
- La ejecución consiste en la resolución de un problema de decisión, los resultados se obtienen mediante la instanciación de las variables libres
- Definición de reglas
- Unificación como elemento de computación
- Programación declarativa

- Lenguajes: Prolog, Mercury, Oz, ...

Programación Reactiva (Dataflow)

- Basado en la **teoría de grafos**
- Un programa consiste en la especificación del flujo de datos entre operaciones
- Las variables se encuentran ligadas a las operaciones que proporcionan sus valores. Un cambio de valor de una variable se propaga a todas las operaciones en que participa.
- Las hojas de cálculo se basan en este modelo
- Lenguajes representativos: Simulink, Oz, Clojure, ...

Paradigma orientado a objetos

El paradigma orientado a objetos, se basa en los conceptos de objetos y clases de objetos. Un objeto es una variable equipada con un conjunto de operaciones que le pertenecen o están definidas para ellos.

Los objetos pueden usarse una y otra vez para construir múltiples objetos con las mismas propiedades o modificarse para construir nuevos objetos con propiedades similares pero no exactamente iguales.

Los objetos y clases son conceptos fundamentales. Una clase es un conjunto de objetos que comparten las mismas operaciones.

Objetos (o al menos referencia a objetos) deben ser valores de la clase base. Así, cualquier operación puede tomar un objeto como un argumento y puede devolver un objeto como resultado. De esta manera el concepto de clase de objetos está relacionado con el concepto de tipo de dato.

Herencia es también vista como un concepto clave dentro del mundo de los objetos. En este contexto, la herencia es la habilidad para organizar las clases de objetos en una jerarquía de subclases y superclases y las operaciones dadas para una clase se pueden aplicar a los objetos de la subclase.

- Definición de clases y herencia
- Objetos como abstracción de datos y procedimientos
- Polimorfismo y chequeo de tipos en tiempo de ejecución
- Lenguajes: Smalltalk, Java, ...

Las características más importantes de la programación orientada a objetos son las siguientes:



- **Abstracción:** Denota las características esenciales de un objeto, donde se captura su comportamiento. Cada objeto en el sistema sirve como modelo de un “agente” abstracto que puede realizar trabajo, informar y cambiar su estado, y “comunicarse” con otros objetos en el sistema sin revelar “cómo” se implementan estas características. Los procesos, las funciones o los métodos, pueden también ser abstraídos, y cuando sucede esto, una variedad de técnicas son requeridas para ampliar una abstracción.
- **Encapsulamiento:** Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.
- **Principio de ocultación :** Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una “interfaz” a otros objetos, que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.
- **Polimorfismo :** Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. Dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en “tiempo de ejecución”, esta última característica se llama “asignación tardía” o “asignación dinámica”. Algunos lenguajes proporcionan medios más estáticos (en “tiempo de compilación”) de polimorfismo, tales como las plantillas y la [[sobrecarga | sobrecarga de operadores]] de C++.
- **Herencia :** Las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el

comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento, permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo.

- **Recolección de basura:** La recolección de basura o Garbage Collection es la técnica por la cual el ambiente de Objetos se encarga de destruir automáticamente, y por tanto, desasignar de la memoria los Objetos que hayan quedado sin ninguna referencia a ellos. Esto significa que el programador no debe preocuparse por la asignación o liberación de memoria, ya que el entorno la asignará al crear un nuevo Objeto y la liberará cuando nadie lo esté usando. En la mayoría de los lenguajes híbridos que se extendieron para soportar el Paradigma de Programación Orientada a Objetos como C++ u Object Pascal, esta característica no existe y la memoria debe desasignarse manualmente.

Según su campo de aplicación

Aplicaciones científicas

En este tipo de aplicaciones predominan las operaciones numéricas o matriciales propias de algoritmos matemáticos. Lenguajes adecuados son FORTRAN y PASCAL.

Aplicaciones en procesamiento de datos

En estas aplicaciones son frecuentes las operaciones de creación, mantenimiento y consulta sobre ficheros y bases de datos. Dentro de este campo estarían aplicaciones de gestión empresarial, como programas de nóminas, contabilidad, facturación, etc. Lenguajes adecuados son COBOL y SQL.

Aplicaciones de tratamiento de textos

Estas aplicaciones están asociadas al manejo de textos en lenguaje natural. Un lenguaje muy adecuado para este tipo de aplicaciones es el C.

Aplicaciones en inteligencia artificial

Dentro de este campo, destacan las aplicaciones en sistemas expertos, juegos, visión artificial, robótica. Los lenguajes más populares son LISP y PROLOG.

Aplicaciones de programación de sistemas

En este campo se incluirían la programación de software de interfaz entre el usuario y el hardware, como son los módulos de un SO y los traductores. Tradicionalmente para estas aplicaciones se utilizaba Ensamblador, en la actualidad ADA, MODULA-2 y C.

Según su Traducción

Intérpretados

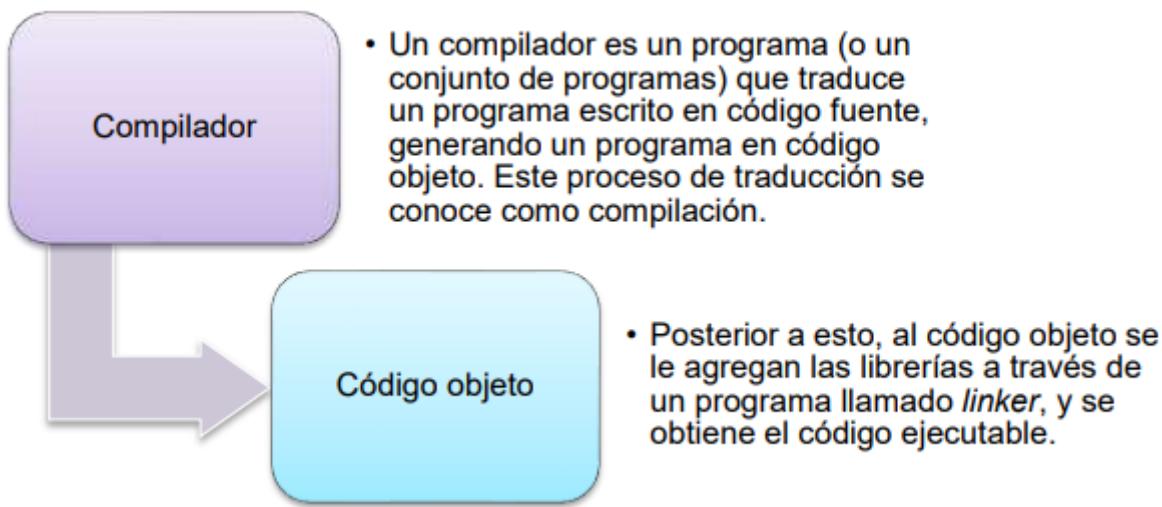
Un intérprete es un programa que analiza y ejecuta un código fuente, toma un código, lo traduce y a continuación lo ejecuta; y así sucesivamente lo hace hasta llegar a la última instrucción del programa, siempre y cuando no se produzca un error en el proceso.

Como ejemplo de lenguajes interpretados están PHP, Perl, Python, ...



Funcionamiento de un intérprete

Compilados



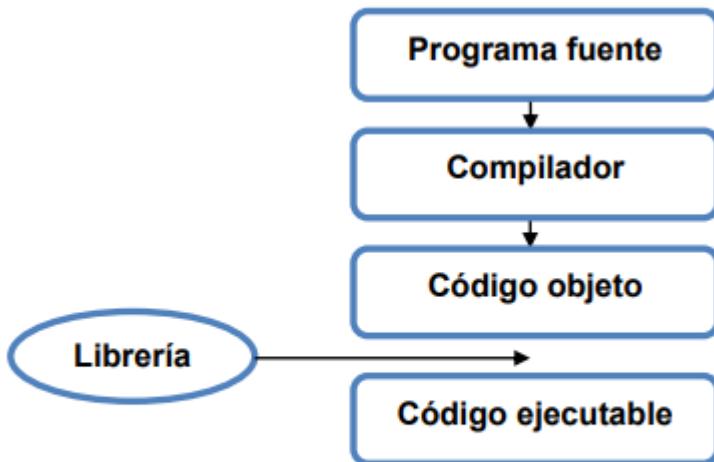
Como ejemplo de lenguajes que utilizan un compilador tenemos a C, C++, Visual Basic, ...

La compilación permite crear un programa de computadora que puede ser ejecutado por una computadora.

La compilación de un programa se hace en tres pasos:

1. Creación del código fuente,
2. Compilación del programa y
3. Enlace del programa con las funciones necesarias de la biblioteca.

La forma en que se lleva a cabo el enlace variará entre distintos compiladores, pero la forma general es:



Proceso de Compilación

Principales Lenguajes de Programación

- **Lenguaje Máquina.** Es el sistema de códigos directamente interpretable por un circuito microprogramable, como el microprocesador de una computadora.
- **Ensamblador** . Es el que proporciona poca o ninguna abstracción del microprocesador de una computadora. Es fácil su traslado al lenguaje máquina.
- **FORTRAN** (FORmula TRANslation). Es un lenguaje de programación desarrollado en los años 50 y activamente utilizado desde entonces. Se utiliza principalmente en aplicaciones científicas y análisis numérico.
- ALGOL (ALGOrithmic Language)
- COBOL (COmmon Business Oriented Language)
- BASIC
- Visual BASIC
- Visual BASIC Script
- Pascal
- Modula-2
- COMAL (COMmon Algorithmic Language)
- APL (A Programming Language)
- LOGO
- HYPERTALK
- ADA
- C
- **C++**. Es un lenguaje de programación, diseñado a mediados de los años 80, por Bjarne Stroustrup. Por otro lado, es un lenguaje que abarca dos paradigmas de la programación: la programación estructurada y la programación orientada a objetos.
- Visual C++
- C#
- LISP (LISt Processing)
- PROLOG (PROgramacion LOGica)
- FORTH
- **Perl** . Lenguaje práctico para la extracción e informe. Es un lenguaje de programación diseñado por Larry Wall creado en 1987. Perl toma características de C, del lenguaje interpretado shell sh, AWK, sed, Lisp y, en un grado inferior, muchos otros lenguajes de programación.
- **Python** . Es un lenguaje de programación creado por Guido van Rossum en el año 1990. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation.
- Clipper
- Delphi

- HTML
- XHTML
- PHP
- SQL
- PL/1
- **Java** . Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible.
- Javascript
- Otros

Características

Tipos de datos elementales

Las formas de organizar datos están determinadas por los tipos de datos definidos en el lenguaje.

Un tipo de dato determina el rango de valores que puede tomar el objeto, las operaciones a que puede ser sometido y el formato de almacenamiento en memoria.

Los tipos de datos pueden ser:

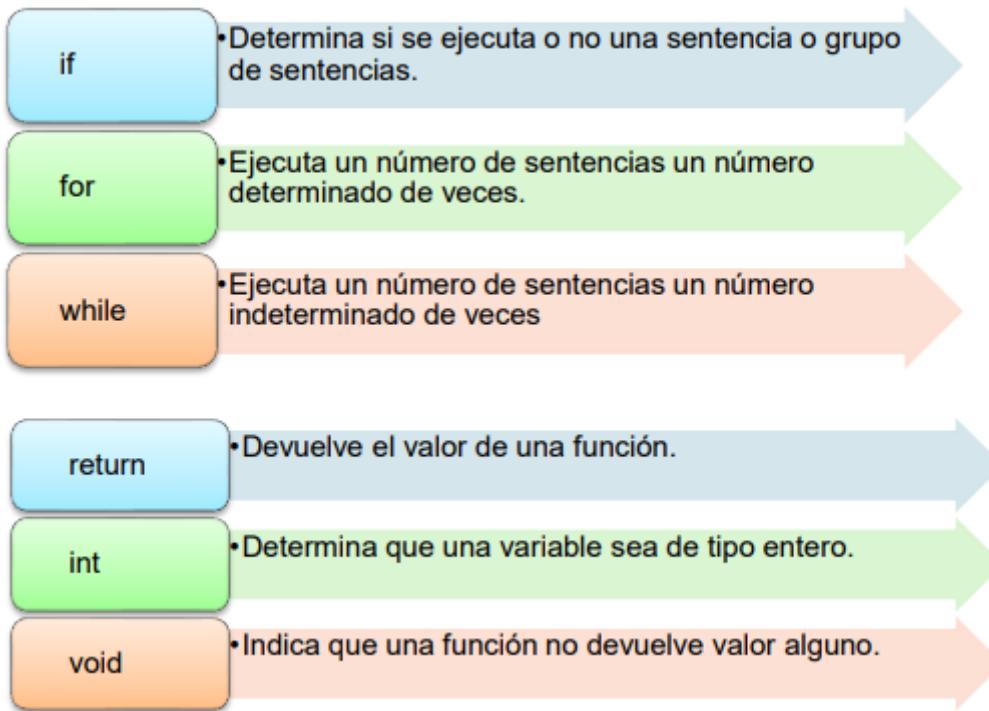
- Predefinidos
- Definidos por el usuario, a partir de los básicos

Tipos de datos elementales:

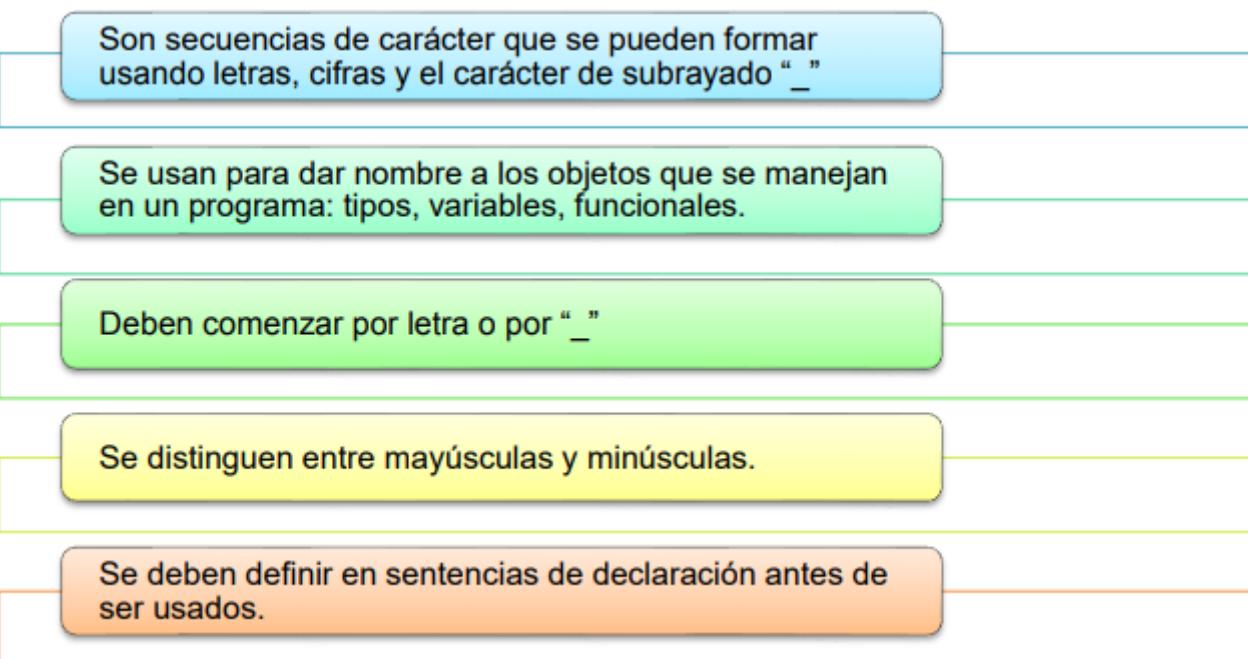
Tipo	Rango de Valores	Tamaño en Bytes	Descripción
char	-128 a 127	1	Para una letra o un dígito.
unsigned char	0 a 255	1	Letra o número positivo
int	-32.768 a 32.767	2	Para números enteros
unsigned int	0 a 65.535	2	Para números enteros
long int	+/-2.147.483.647	4	Para números enteros
unsigned long int	0 a 4.294.967.295	4	Para números enteros
float	3.4E-38 decimales (6)	6	Para números con decimales
double	1.7E-308 decimales (10)	8	Para números con decimales
long double	3.4E-4932 decimales (10)	10	Para números con decimales

Palabras reservadas

Las palabras reservadas son símbolos cuyo significado está predefinido y no se pueden usar para otro fin; aquí tenemos algunas palabras reservadas en el lenguaje C.



Identificadores



Hay que decir que no todos los lenguajes distinguen entre mayúsculas y minúsculas.

Operadores

Existen diferentes tipos de operadores: asignación, aritméticos, relacionales, lógicos y de bits.

Asignación

Al utilizarlo se realiza esta acción: el operador destino (parte izquierda) debe ser siempre una variable, mientras que en la parte derecha puede estar cualquier expresión válida. Con esto el valor de la parte derecha se asigna a la variable de la derecha.

Asignación

Los operadores de asignación son aquellos que nos permiten modificar el valor de una variable, el operador de asignación básico es el “igual a” (=), que da el valor que lo sigue a la variable que lo precede.

Sintaxis

```
variable=valor;  
variable=variable1;  
variable=variable1=variable2=variableN=valor;
```

Operador	Descripción
=	Asignación
+=	Incremento y Asignación
-=	Decremento y Asignación
%=	Módulo y Asignación
*=	Multiplicación y Asignación
/=	División y Asignación
<<=	Desplazamiento de bits a izquierda y Asignación
>>=	Desplazamiento de bits a derecha y Asignación
&=	AND lógico de bits y Asignación
=	OR lógico de bits y Asignación
^=	XOR lógico de bits y Asignación

Aritméticos

Los operadores aritméticos pueden aplicarse a todo tipo de expresiones. Son utilizados para realizar operaciones matemáticas sencillas, aunque uniéndolos se puede realizar cualquier tipo de operaciones.

En la siguiente tabla se muestran los operadores aritméticos:

Operador	Descripción	Ejemplo
-	Resta	a - b
+	Suma	a + b
*	Multiplica	a * b
/	Divide	a / b
%	Módulo (resto de división)	a % b
-	Signo negativo	-a
--	Decremento en 1	a--
++	Incremento en 1	a++

- Los operadores `-`, `+`, `*`, `/`, `%` se corresponden con operaciones matemáticas. Son binarios porque cada uno tiene dos operandos.
- Hay un operador unario `-`, pero no hay un operador `+`.
- La división de enteros devuelve el cociente entero y desecha la fracción restante.
- El operador módulo se aplica así: con dos enteros positivos, devuelve el resto de la división. Ejemplos: `12%3` tiene el valor 0. `12%5` tiene el valor 2.
- Los operadores `-` y `++` son unarios.
 - Tienen la misma prioridad que el `-` unario.
 - Se asocian de derecha a izquierda.
 - Pueden aplicarse a variables, pero no a constantes ni a expresiones.
 - Se pueden presentar como prefijo o como sufijo.
 - Aplicados a variables enteras, su efecto es incrementar o decrementar el valor de la variable en una unidad.
 - `++i` es equivalente a `i=i+1;`
 - `-i` es equivalente a `i=i-1;`
 - Con `++a` el valor de "a" se incrementa antes de evaluar la expresión.
 - Con `a++` el valor de "a" se incrementa después de evaluar la expresión.

Relacionales

Los operadores relacionales hacen referencia a la relación entre unos valores y otros.

Operador	Descripción
<code><</code>	Menor que
<code>></code>	Mayor que
<code><=</code>	Menor o igual que
<code>>=</code>	Mayor o igual que
<code>==</code>	Igual
<code>!=</code>	Distinto

Lógicos

Los operadores lógicos hacen referencia a la forma en que las relaciones pueden conectarse entre sí.

Operador	Descripción
<code>&&</code>	Y (AND)
<code> </code>	O (OR)
<code>!</code>	NO (NOT)

De Bits

Los operadores de bits son operadores binarios, excepto la negación que es unario. Se aplican a variables enteras, resultando otro entero aplicando operaciones lógicas correspondientes a los bits de los operandos. Al desplazar bits a la izquierda, suele rellenarse con 0 por la derecha y al desplazar a la derecha, se rellena por la izquierda con 0 si el dígito más significativo es 0, y con 1 si es 1. El tipo de variable es con signo.

Operador	Descripción
	OR de bits
&	AND de bits
>>	Desplazamiento de bits a la derecha
<<	Desplazamiento de bits a la izquierda
^	XOR (O Exclusivo) de bits
~	NOT de bits

Expresiones y reglas de prioridad

Una expresión se forma combinando constantes, variables, operadores y llamadas a funciones.

Una expresión representa un valor, el resultado de realizar las operaciones indicadas siguiendo las reglas de evaluación establecidas en el lenguaje.

Con expresiones se forman sentencias; con éstas, funciones, y con éstas últimas se construye un programa completo.

Cada expresión toma un valor que se determina tomando los valores de las variables y constantes implicadas y la ejecución de las operaciones indicadas.

Una expresión consta de operadores y operandos.

En la tabla de prioridades veremos operadores no estudiados, la línea de separación indica diferentes prioridades:

Operador	Tipo	Asociatividad
()	Paréntesis	Derecha – Izquierda
()	Llamada a función	Derecha – Izquierda
[]	Subíndice	Derecha – Izquierda
.	Acceso a miembros de un objeto	Derecha – Izquierda
<hr/>		
++	Prefijo incremento	Derecha – Izquierda
--	Prefijo decremento	Derecha – Izquierda
<hr/>		
+	Más unario	Derecha – Izquierda
-	Menos unario	Derecha – Izquierda
!	Negación lógica unaria	Derecha – Izquierda
(tipo)	Modelado unario	Derecha – Izquierda
New	Creación de objetos	Derecha – Izquierda
<hr/>		
*	Producto	Izquierda – Derecha
/	División	Izquierda – Derecha
%	Resto entero	Izquierda – Derecha
+	Suma	Izquierda – Derecha
-	Resta	Izquierda – Derecha
<hr/>		
<<	Desplazamiento bit a bit a la izquierda	Derecha – Izquierda
>>	Desplazamiento bit a bit a la derecha con extensión de signo	Derecha – Izquierda
>>>	Desplazamiento bit a bit a la derecha llenando con ceros	Derecha – Izquierda
<hr/>		
<	Menor que	Izquierda - Derecha
<=	Menor o igual que	Izquierda – Derecha
>	Mayor que	Izquierda – Derecha
>=	Mayor o igual que	Izquierda – Derecha
Instanceof	Verificación tipo de objeto	Izquierda – Derecha
<hr/>		
==	Igualdad	Izquierda – Derecha
!=	Desigualdad	Izquierda – Derecha
<hr/>		
&	AND bit a bit	Izquierda – Derecha
<hr/>		
^	OR exclusivo bit a bit	Izquierda – Derecha
<hr/>		
	OR inclusivo bit a bit	Izquierda – Derecha
<hr/>		
&&	AND lógico	Izquierda – Derecha
<hr/>		
	OR lógico	Izquierda – Derecha
<hr/>		

<code>?:</code>	Condicional ternario	Derecha – Izquierda
<code>=</code>	Asignación	Derecha – Izquierda
<code>+=</code>	Asignación de suma	Derecha – Izquierda
<code>-=</code>	Asignación de resta	Derecha – Izquierda
<code>*=</code>	Asignación de producto	Derecha – Izquierda
<code>/=</code>	Asignación de división	Derecha – Izquierda
<code>%=</code>	Asignación de módulo	Derecha – Izquierda
<code>&=</code>	Asignación AND bit a bit	Derecha – Izquierda
<code>^=</code>	Asignación OR exclusivo bit a bit	Derecha – Izquierda
<code> =</code>	Asignación OR inclusivo bit a bit	Derecha – Izquierda
<code><<=</code>	Asignación de desplazamiento a izquierda bit a bit	Derecha – Izquierda
<code>>>=</code>	Desplazamiento derecho bit a bit con asignación de extensión de signo	Derecha – Izquierda
<code>>>>=</code>	Desplazamiento derecho bit a bit con asignación de extensión a cero.	

Variabes, constantes, conversión y ámbito de variables

Todas las definiciones serán respecto al lenguaje C.

Variables

Variables

Unidad básica de almacenamiento de valores. La creación de una variable es la **combinación** de un *identificador*, un *tipo* y un *ámbito*. Todas las variables en C deben ser declaradas antes de ser usadas.

No en todos los lenguajes hace falta declarar antes las variables, aunque siempre es recomendable.

Las variables, también conocidas como identificadores, deben cumplir las siguientes reglas:

La longitud puede ir de 1 carácter a 31.

El primero de ellos debe ser siempre una letra.

No puede contener espacios en blanco, ni acentos y caracteres gramaticales.

Hay que tener en cuenta que el compilador distingue entre mayúsculas y minúsculas.

SINTAXIS:

```
tipo nombre=valor_numerico;  
tipo nombre='letra';  
tipo nombre[tamaño]="cadena de letras";  
tipo nombre=valor_entero.valor_decimal;
```

Constantes

Constantes

Las constantes se refieren a los valores fijos que no pueden ser modificados por el programa. Las constantes de carácter van encerradas en comillas simples. Las constantes enteras se especifican con números sin parte decimal, y las de coma flotante, con su parte entera separada por un punto de su parte decimal.

Las constantes son entidades cuyo valor no se modifica durante la ejecución del programa.

Sintaxis

```
const tipo  
nombre=valor_entero;  
const tipo  
nombre=valor_entero.valor_de  
cimal;  
const tipo nombre='carácter';
```

Conversión

Conversión

Las conversiones (*casting*) automáticas pueden ser controladas por el programador. Bastará con anteponer y encerrar entre paréntesis el tipo al que se desea convertir.

Este tipo de conversiones es temporal y la variable por convertir mantiene su valor.

SINTAXIS:

```
variable_destino=(tipo)variable_a_convertir;  
variable_destino=(tipo)(variable1+variable2+variableN);
```

Ámbito de variables

Ámbito de variables

Según el lugar donde se declaren las variables tendrán un ámbito. Según el ámbito de las variables pueden ser utilizadas desde cualquier parte del programa o únicamente en la función donde han sido declaradas.

Las variables pueden ser:

Locales

- Cuando se declaran dentro de una función. Pueden ser referenciadas (utilizadas) por sentencias que estén dentro de la función que han sido declaradas. No son conocidas fuera de su función. Pierden su valor cuando se sale de la función.

Globales

- Son conocidas a lo largo de todo el programa y se pueden usar desde cualquier parte del código. Mantienen sus valores durante toda la ejecución. Deben ser declaradas fuera de todas las funciones incluida main(). La sintaxis de creación no cambia nada con respecto a las variables locales.

De registro

- Otra posibilidad es que, en vez de ser mantenidas en posiciones de memoria de la computadora, se las guarde en registros internos del microprocesador. De esta manera, el acceso a ellas es más directo y rápido. Para indicar al compilador que es una variable de registro, hay que añadir a la declaración la palabra register delante del tipo. Solo se puede utilizar para variables locales.

Estáticas

- Las variables locales nacen y mueren con cada llamada y finalización de una función. Sería útil que mantuvieran su valor entre una llamada y otra sin por ello perder su ámbito. Para conseguir eso se añade a una variable local la palabra static delante del tipo.

Externas

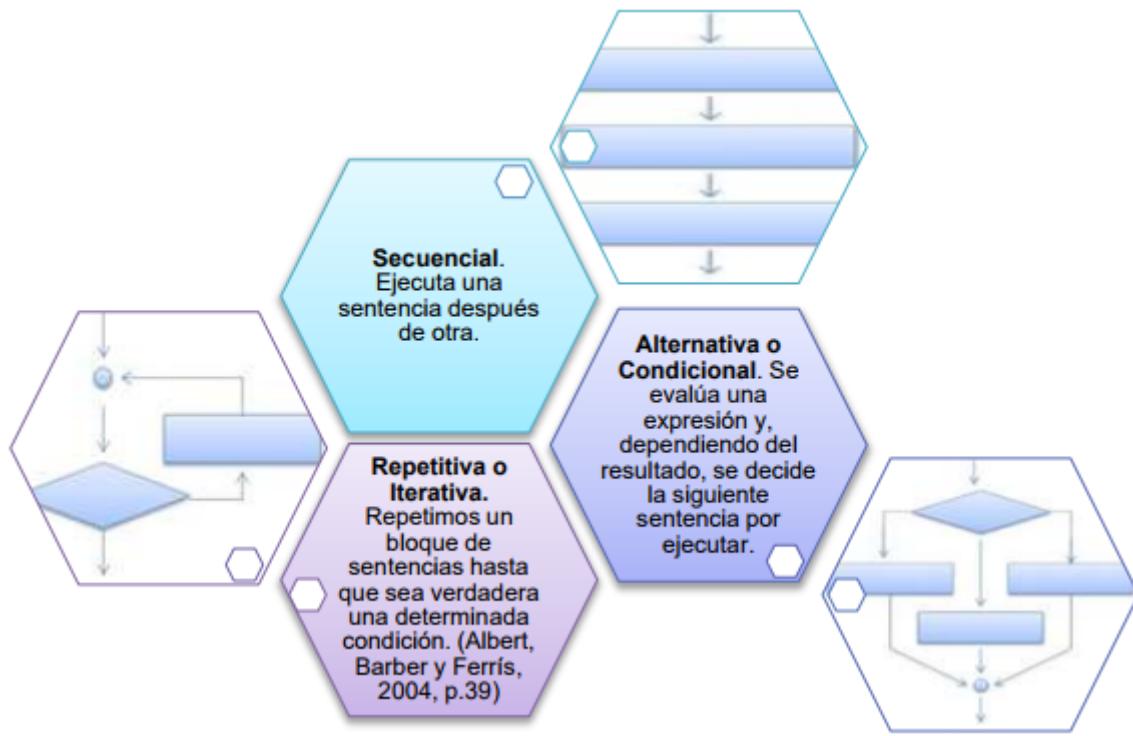
- Debido a que en C es normal la compilación por separado de pequeños módulos que componen un programa completo, puede darse el caso de que deba utilizar una variable global que se conozca en los módulos que nos interesen sin perder su valor. Añadiendo delante del tipo la palabra extern y definiéndola en los otros módulos como global ya tendremos nuestra variable global.

Estructura de un programa

La estructura de un programa está compuesto de bibliotecas (#include), función principal (main), llaves ({}), y dentro de éstas, la declaración de variables y desarrollo del programa (conjunto de instrucciones).

Control de flujo

El teorema del programa estructurado, demostrado por Böhm-Jacopini, demuestra que todo programa puede escribirse utilizando únicamente las tres instrucciones de control siguientes:



Solamente con estas tres estructuras se pueden escribir cualquier tipo de programa.

La programación estructurada crea programas claros y fáciles de entender, además los bloques de código son auto explicativos, lo que facilita la documentación. No obstante, cabe destacar que en la medida que los programas aumentan en tamaño y complejidad, su mantenimiento también se va haciendo más difícil.

Estructura secuencial

El control de flujo se refiere al orden en que se ejecutan las sentencias del programa. A menos que se especifique expresamente, el flujo normal de control de todos los programas es secuencial.

Estructura secuencial

La estructura secuencial ejecuta las acciones sucesivamente, sin posibilidad de omitir ninguna y sin bifurcaciones. Todas estas estructuras tendrán una entrada y una salida.

Estructura alternativa

Estructura alternativa

La estructura alternativa es aquella en que la existencia o cumplimiento de la condición, implica la **ruptura** de la **secuencia** y la **ejecución** de una determinada acción. Es la manera que tiene un lenguaje de programación de provocar que el flujo de la ejecución avance y se ramifique en función de los cambios de estado de los datos.

Las estructuras utilizadas son:

- **if-else** . La cláusula else es opcional. Se pueden anidar varios if.
- **switch**. Realiza distintas operaciones con base en el valor de la única variable o expresión. El valor de la expresión se compara con cada uno de los literales de la sentencia, si coincide alguno, se ejecuta el código, si no se realiza la sentencia default (opcional), si no existe default no se ejecuta nada. La sentencia break realiza la salida de un bloque de código.

Ejemplo:

<pre>switch (expresión){\n case valor1:\n sentencia;\n break;\n case valor2:\n sentencia;\n break;\n case valor3:\n sentencia;\n break;\n case valorN:\n sentencia;\n break;\n default:\n}\n\n}</pre>	<pre>if (expresion-boleana)\n{\n Sentencia1;\n Sentencia2;\n}\nElse\n Sentencia3;</pre>
--	--

Estructuras repetitiva

Estructuras repetitivas o iterativas

Las estructuras repetitivas o iterativas son aquellas en las que las acciones se ejecutan un número determinado o indeterminado de veces y dependen de un valor predefinido o el cumplimiento de una condición.

while

Ejecuta repetidamente el mismo bloque de código hasta que se cumpla una condición de terminación. Hay dos partes en un ciclo: *cuerpo* y *condición*.

While

Es la sentencia o sentencias que se ejecutarán dentro del ciclo.

Cuerpo

Es la condición de terminación del ciclo.

Condición

```
while(condición){  
    cuerpo;  
}
```



do-while

do-while

Es lo mismo que en el caso anterior pero aquí como mínimo siempre se ejecutará el cuerpo al menos una vez, en el caso anterior es posible que no se ejecute ni una sola vez.

```
do{  
    cuerpo;  
}while(terminación);
```

for

for

Realiza las mismas operaciones que en los casos anteriores pero la sintaxis es una forma compacta. Normalmente la condición para terminar es de tipo numérico. La iteración puede ser cualquier expresión matemática válida. Si de los 3 términos que necesita no se pone ninguno se convierte en un bucle infinito.

```
for (inicio; condición; incremento/decremento)  
{  
    sentencia(s);  
}
```

Funciones

En el ámbito de la programación, una función es el término para describir una secuencia de órdenes que hacen una tarea específica.

Características:

Un nombre único con el que se identifica y distingue a la función.

Un valor que la función devolverá al terminar su ejecución.

Una lista de parámetros que la función debe recibir para realizar su tarea.

Un conjunto de órdenes o sentencias que debe ejecutar la función.

Las funciones son las que realizan las tareas principales de un programa. Las funciones realizan una tarea específica. Subdivide en varias tareas un programa, con pocas líneas y haciendo una tarea simple. Las funciones pueden o no devolver y recibir valores del programa.

Hay dos tipos de funciones:

- Funciones Internas. Funciones internas del lenguaje que realizan tareas específicas. Por ejemplo, hay funciones para el manejo de caracteres y cadenas, matemáticas, de conversión, ...
- Definidas por el usuario. Primero hay que declarar el prototipo de la función, a continuación debemos hacer la llamada y por último desarrollar la función.

Sintaxis del prototipo:

• `tipo_devuelto nombre_funcion ([parámetros]);`

Sintaxis de la llamada:

• `nombre_funcion([parámetros]);`

Sintaxis del desarrollo:

• `tipo_devuelto nombre_funcion ([parámetros])
{
 cuerpo;
}`

Ámbito de las variables (locales y globales)

Variables locales

Las variables son **locales** cuando se declaran dentro de una función. Las variables locales sólo pueden ser referenciadas (utilizadas) por sentencias que estén dentro de la función donde han sido declaradas; cuando se sale de la función, los valores de estas variables se pierden.

Variables globales

Las variables son **globales** cuando son conocidas a lo largo de todo el programa, y se pueden usar desde cualquier parte del código. Mantienen sus valores durante toda la ejecución. Deben ser declaradas fuera de todas las funciones, incluida main(); sin embargo, al declarar variables locales dentro de la función main(); sus valores son reconocidos en todo el programa como si fueran variables globales, debido a que main() es la función principal.

Recursividad

Recursividad

La recursividad es el proceso de definir algo en términos de sí mismo, es decir, que las funciones pueden llamarse a sí mismas; esto se consigue cuando en el cuerpo de la función hay una llamada a la propia función, entonces se dice que es recursiva. Una función recursiva no hace una nueva copia de la función, sólo son nuevos los argumentos.

La principal ventaja de las funciones recursivas es que se pueden usar para crear versiones de algoritmos más claros y sencillos. Cuando se escriben funciones recursivas, se debe tener una sentencia *if* para forzar a la función a volver sin que se ejecute la llamada recursiva.

Las funciones recursivas pueden ahorrar la escritura de código, sin embargo, se deben usar con precaución, pues pueden generar un excesivo consumo de memoria.

Tipos de datos compuestos (estructuras)

Un array es una colección de variables del mismo tipo que se refieren por un nombre en común. A un elemento específico de un array se accede mediante un índice. Todos los arrays constan de posiciones de memoria contiguas. La dirección más baja corresponde al primer elemento. Los arrays pueden tener una o varias dimensiones.

Una estructura es una colección de variables que se refiere bajo un único nombre, y a diferencia del array, puede combinar variables de tipos distintos.

Tanto los arrays como los registros son estructuras de datos que sirven para almacenar valores en memoria, la diferencia radica en que el array solo te permite almacenar un tipo

específico de datos: entero, caracteres, fechas, ... y los registros, como se ha indicado, admite diferentes tipos de datos.

Arrays unidimensionales y bidimensionales

Arreglos

Los **arreglos** son una colección de variables del mismo tipo que se referencian utilizando un nombre común. Un arreglo consta de posiciones de memoria contigua.

La dirección más baja corresponde al primer elemento, y la más alta al último. Un array puede tener una o varias dimensiones. Para acceder a un elemento en particular de un array se usa un índice.

Elemento 1
Elemento 2
Elemento 3
.....
Elemento n

Elemento 1,1	Elemento 1,n
Elemento 2,1	Elemento 2,n
Elemento 3,1	Elemento 3,n
.....
Elemento m,1	Elemento m,n

Arreglo unidimensional

Arreglo bidimensional

- El formato para declarar un array unidimensional es:
 - *tipo nombre_array[tamaño]*
- El formato para declarar un array bidimensional es :
 - *tipo nombre_array[tamaño1][tamaño2]...[tamañoN]*

En la mayoría de los lenguajes los arrays usan el cero como índice para el primer elemento.

Arrays multidimensionales

Los arreglos multidimensionales nos sirven para almacenar tablas de valores que tengan varias filas y columnas, e inclusive profundidad (cubos de datos), así podemos hacer referencia a un dato indicando su posición, fila y columna, o bien, fila, columna, profundidad, en su caso.

Elemento 1,1	Elemento 1,n
Elemento 2,1	Elemento 2,n
Elemento 3,1	Elemento 3,n
.....
Elemento m,1	Elemento m,n

Arreglo bidimensional



Arreglo multidimensional

- El formato para declarar un array multidimensional es:
 - *tipo nombre_array[fila][columna]*

Estructuras

Estructura

Colección de variables que se referencia bajo un único nombre, proporcionando un medio eficaz de mantener junta una información relacionada.

Las variables que componen la estructura se llaman miembros de la estructura y están relacionadas lógicamente con las otras.

Otra característica es el ahorro de memoria y evitar declarar variables que técnicamente realizan las mismas funciones.

Una definición de estructura forma una plantilla que se puede usar para crear variables de estructura. Las variables que forman la estructura son llamados elementos estructurados.

Inteligencia de negocios: cuadros de mando integral, sistemas de soporte a las decisiones, sistemas de información ejecutiva y almacenes de datos. OLTP y OLAP.

Introducción. Evolución de los sistemas tradicionales a las Bases de Datos.

Sistemas tradicionales de ficheros. Inconvenientes.

Los sistemas tradicionales de ficheros son sistemas orientados hacia el proceso, es decir, se pone el énfasis en el tratamiento que reciben los datos, los cuales se almacenan en archivos diseñados para una aplicación concreta. Las aplicaciones se diseñan e implantan independientemente unas de otras y los datos se duplican si las diferentes aplicaciones los necesitan, en lugar de transferirse entre ellas.

Los principales inconvenientes de estos sistemas tradicionales de ficheros son:

- Redundancia: duplicidad innecesaria de información.
- Mal aprovechamiento del equipo de almacenamiento: como consecuencia inmediata de la redundancia.
- Aumento de los tiempos de proceso: se repiten los mismos controles y operaciones en los distintos ficheros, con lo que se consume más tiempo de CPU del necesario. En el caso de modificar un campo hay que hacerlo en todos los registros de todos los ficheros que lo contengan.
- Inconsistencia de la información: por la alta redundancia. Si se deja de actualizar un dato en uno de los archivos donde aparece, la información proporcionada por este dato se vuelve inconsistente.
- Aislamiento de los datos: Cada archivo pertenece a un programa y no es posible que estos sean usados por nuevos programas. Un nuevo programa necesitará sus propios archivos de datos que habrán de crearse aunque parte de los datos ya existan en otros archivos de otros programas, contribuyendo a aumentarla redundancia y las consecuencias de esta.
- Imposibilidad de responder a demandas inesperadas de información: los sistemas tradicionales de archivo son inoperantes para conseguir un sistema de información orientado a la toma de decisiones.
- Dependencia total entre los programas y la estructura física de los datos: no es posible modificar las características físicas (estructura y métodos de acceso) de los archivos sin afectar a los programas que los usan. Conseguir la independencia entre datos y aplicaciones va a ser uno de los principales objetivos de los sistemas de bases de datos.

Sistemas orientados a la base de datos

Ante los problemas descritos con los sistemas tradicionales de ficheros, surge la necesidad de una gestión más racional del conjunto de los datos. Poco a poco se fue poniendo más énfasis en un enfoque distinto, en el cual la información se organizaba y se mantenía como un conjunto estructurado que no se diseñaba para una aplicación concreta. Es decir, surge así un nuevo enfoque que se apoya sobre una base de datos en la que los datos son recogidos y almacenados una sola vez con independencia de los tratamientos que se van a aplicar sobre ellos.

De esta forma, la información contenida en una base de datos está integrada y compartida. *Integrada* porque puede considerarse como una unificación de varios archivos de datos de los que hemos eliminado la redundancia y *compartida* porque los programas que antes accedían a los archivos individuales acceden ahora al depósito común de datos, por lo que cada usuario o aplicación tendrá acceso a un subconjunto de los datos y como consecuencia diferentes usuarios verán de formas muy diferentes la misma base de datos.

Es importante destacar que los subconjuntos de datos a los que acceden las diferentes aplicaciones o usuarios no tienen por qué ser disjuntos, por lo que usuarios o aplicaciones distintas pueden acceder a la misma parte de la base de datos para utilizarla con propósitos diferentes.

Concepto de Base de Datos

Una primera definición de bases de datos sería: "Una Base de Datos (BD) es una colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos y su definición (estructura de la BD), única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos".

Veamos en qué consiste cada uno de los aspectos mencionados en esta definición de Base de Datos que no son más que distintas definiciones según distintas perspectivas. La BAD es un conjunto de datos relativos a una determinada parcela del mundo real que se almacenan en un soporte informático no volátil. Además, no debe existir redundancia, es decir, no deben existir duplicidades perjudiciales ni innecesarias (a ser posible un determinado tipo de dato, sólo deben aparecer en un sitio en la BD). En ocasiones, es necesaria cierta redundancia (a nivel de almacenamiento físico) que mejora la eficiencia de la BD. Sin embargo, esta redundancia siempre debe ser controlada por el sistema para que no se produzcan inconsistencias. Por otro lado, las BD han de atender a múltiples usuarios de la organización así como a distintas aplicaciones.

Otras definiciones de BD son:

- Conjunto de datos de la empresa memorizados en un ordenador, que es utilizado por numerosas personas y cuya organización está recogida en un modelo de datos.
- Colección o depósito de datos, donde estos se encuentran lógicamente relacionados entre sí, tienen una definición y una descripción comunes y están estructurados de una forma particular.
- Colección o depósito de datos integrados, con redundancia controlada y una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de estos, y su definición y descripción, únicas para cada tipo de dato, han de estar almacenados junto con los mismos. Los procedimientos de actualización y recuperación, comunes y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de los datos.

Principales características de los datos almacenados en una BD:

- Están organizados
- Están relacionados
- Son accesibles de diferentes formas sin grandes dificultades
- Se almacenan solo una vez

Independencia de los datos. Niveles de abstracción.

En los sistemas de base de datos se plantean dos objetivos principales:

- Independencia de la base de datos de los programas para su utilización.
- Proporcionar a los usuarios una visión abstracta de los datos. El sistema esconde los detalles de almacenamiento físico (como almacena y mantiene los datos), pero deben extraerse eficientemente.

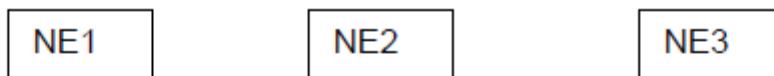
Independencia de los Datos. Definición de ANSI.

La independencia de los datos es la capacidad de un sistema para permitir que las referencias a los datos almacenados, especialmente en los programas y en sus descriptores de datos, estén aislados de los cambios y de los diferentes usos en el entorno de los datos, como puede ser la forma de almacenar dichos datos, el modo de compartirlos con otros programas y como se reorganizan para mejorar el rendimiento del sistema de Bases de Datos.

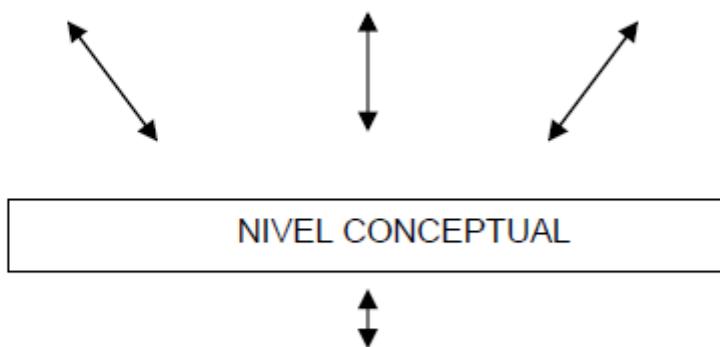
Niveles de abstracción

Para conseguir esta independencia entre los datos y las aplicaciones es necesario separar la representación física y lógica de los datos, distinción que fue reconocida oficialmente en 1978, cuando el comité ANSI/X3/SPARC propuso un esqueleto generalizado para sistemas de BD. Este esqueleto propone una arquitectura de tres niveles, los tres niveles de abstracción bajo los que podía verse una base de datos: el nivel interno, el nivel conceptual y el nivel externo.

NIVEL EXTERNO: visión que de la base de datos tiene un usuario o aplicación en particular.



NIVEL CONCEPTUAL: contiene el diseño conceptual de la base de datos.



NIVEL INTERNO: es el que define la estructura física de la base de datos.



Esta arquitectura de tres niveles nos proporciona la deseada independencia, que definimos como la capacidad para cambiar el esquema en un nivel sin tener que cambiarlo en ningún otro nivel. Distinguimos dos tipos de independencia:

- *Independencia lógica de datos* : cambio del esquema conceptual sin cambiar las vistas externas o las aplicaciones.
- *Independencia física de los datos* : cambio del esquema interno sin necesidad de cambiar el esquema conceptual o los esquemas externos.

Ventajas e inconvenientes del uso de bases de datos

• Ventajas

- *Control sobre las inconsistencias y redundancias* : en los sistemas tradicionales de ficheros cada aplicación tiene sus datos privados, lo que provoca una alta redundancia y desaprovechamiento del espacio en disco. La redundancia debe minimizarse y controlarse con las BD. Aunque se mantenga cierto grado de redundancia por motivos de rendimiento u otros, el sistema proporciona mecanismos para garantizar la consistencia. Se controla la redundancia garantizando que los datos redundantes se actualicen de forma automática.
- *Mejor servicio a los usuarios* : en los sistemas convencionales suele ser difícil obtener una información para la cual no fueron diseñados. Una vez que varios de estos sistemas se combinan para crear una BD centralizada, además de mejorar sustancialmente la disponibilidad de la información también garantiza que los datos son actuales. Asimismo es posible responder más ágilmente a nuevas necesidades del usuario no planteadas anteriormente.
- *Los datos pueden compartirse* : las necesidades de datos de nuevas aplicaciones pueden atenderse con los ya existentes sin tener que almacenar nuevos datos.
- *Mejora la flexibilidad del sistema* : a menudo se plantea la necesidad de cambiar los datos almacenados. Estos cambios, a través de un sistema de BD, no tienen el impacto sobre las aplicaciones que tendrían un sistema convencional.
- *Menor coste de desarrollo y mantenimiento* : aunque el coste inicial de una base de datos puede ser superior al de un sistema tradicional, los costes en mantenimiento y desarrollo de aplicaciones son menores.
- *Pueden hacerse cumplir las normas establecidas* : al tener un control centralizado de las BD, el administrador (a instancias del responsable de la información de la organización) puede garantizar que se observen las normas de la empresa aplicables a la representación de los datos.
- *Restricciones de seguridad* : el administrador puede asegurar que el único modo de acceso sea a través de los canales establecidos, y definir controles de autorización que pueden afectar a cada modo de acceso (modificación, inserción, borrado o lectura), según las necesidades de cada usuario.
- *Se desarrolla un modelo de datos* : en los sistemas convencionales los ficheros de datos se diseñan teniendo en mente las necesidades particulares de la aplicación que los va a utilizar, dejando de lado la visión más general. En los sistemas de BD al estar la información centralizada, se hace necesario un punto de vista más general a la hora de diseñar el modelo de datos.
- *Se reduce el espacio de almacenamiento* : al integrar en un solo espacio los datos de varios sistemas aislados y evitar que se repitan, se requiere un menor volumen para almacenar los mismos datos. Por otro lado esto facilita las tareas de realización de copias de seguridad y su recuperación.

• Inconvenientes

- Instalación costosa.
- Necesidad de personal especializado.
- Necesidad de hardware adicional: al ser los requerimientos superiores a los de un sistema tradicional, ya sea en cuanto a memoria, capacidad de proceso, etc.
- El sistema adquiere mayor complejidad: al integrarse dentro del SO un nuevo sistema activo que interacciona con él e influye en la capacidad de responder al usuario.
- Implantación larga y difícil.
- Falta de rentabilidad a corto plazo.

Principales componentes de un entorno de BD

Ahora vamos a desmenuzar un poco más el concepto de BD, y vamos a describir sus componentes básicos, justificando además la necesidad de cada uno de ellos.

Datos

Una BD no tiene sentido sino está compuesta por datos. Lo que no está tan claro es la forma en que estos datos se deben disponer, qué datos se deben almacenar y cómo los debe entender la máquina.

La disposición de los datos depende del ámbito de aplicación concreto en que se enmarque la BD. No es lo mismo una BD que almacene un dibujo vectorial de monumentos históricos que una BD para almacenar las reservas de clientes en un hotel. Lo que varía fundamentalmente en uno y otro caso, es la forma en que los datos se relacionan entre sí, y el tipo de accesos a la información que va a realizar el usuario. Además, los datos deben disponerse de manera que las consultas sean lo más eficientes posible, evitando a la vez la existencia de datos duplicados que pueden dar al traste con la coherencia de la BD. De esta forma, no sólo se consideran datos aquellos que el usuario desea almacenar, sino toda la estructura de apoyo que el sistema necesite para hacer más eficiente una consulta.

Los índices son ficheros auxiliares que facilitan el acceso a los datos que se encuentran en el fichero principal de la BD. La gran ventaja de los índices es que el mantenimiento de los mismos lo realiza de forma automática el sistema. De esta forma se obtiene una gran eficiencia en las consultas, sin que el usuario tenga que trabajar más. Pues bien, este fichero aparte, creado con el único objetivo de facilitar el acceso a la información, también lo consideramos datos, aunque deban ser gestionados automáticamente, e incluso algunos usuarios no tengan ni la menor idea de su existencia.

Por otro lado, es muy importante decidir qué datos se van a almacenar. Hay que encontrar un equilibrio entre las situaciones en las que almacenamos datos de más (históricos muy poco utilizados) y en las que se almacenan datos de menos (guardando los estrictamente necesarios en el momento de definir la BD).

Por último, también es interesante, por cuestiones de seguridad y aprovechamiento de los recursos, decidir en qué formato se van a almacenar los datos: si se van a almacenar encriptados para que nadie los pueda copiar, o si se van a codificar o a comprimir de alguna forma para hacer que ocupen menos espacio.

Metadatos

Desde el momento en que se crea una BD, hasta el momento en que se desecha porque se compra un sistema mejor, o se instala una nueva BD, la estructura de la BD (o sea, los datos que se deciden almacenar, y la estructura con que se almacenan) cambia a medida que cambian las necesidades sobre la información a obtener de la BD.

Dado que la BD que solucione unas necesidades concretas puede adoptar muchas formas posibles, es muy interesante poseer algún lugar que indique al personal encargado de mantener la BD, cuál es el objetivo de cada dato particular almacenado en la base, así como en qué aplicaciones es utilizado, y con qué propósito, si es un dato fundamental, o si puede ser omitido por el que introduce los datos, etc.

De esta forma, antes de modificar el esquema o estructura de la BD, el departamento de proceso de datos debe consultar esta información sobre los datos de la base, con cuidado para no cometer errores graves que repercutan sobre el buen funcionamiento de todo el sistema.

Esta información que el sistema guarda sobre los datos almacenados, es lo que se llaman metadatos, es decir, datos acerca de los datos. Es más, estos metadatos se almacenan en el diccionario de datos o catálogo.

El Sistema Gestor de Bases de Datos

En un Sistema de BD, debe existir una capa intermedia entre los datos almacenados en la BD, las aplicaciones y los usuarios del mismo. Se trata del Sistema de Gestión de la BD (SGBD). Actúa de intermediario entre los usuarios y aplicaciones y los datos proporcionando medios para describir, almacenar y manipular los datos, y proporciona herramientas al administrador para gestionar el sistema, entre ellas las herramientas de desarrollo de aplicaciones, generadores de informes, lenguajes específicos de acceso a los datos, como SQL (Structure Query Language) o QBE (Query By Example) en BD relacionales.

Un SGBD se puede definir como un conjunto coordinado de programas, procedimiento, lenguajes, etc. que suministra, tanto a los usuarios finales como a los analistas, programadores o el administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad. Entre sus funciones destacan:

- *Definición y control centralizado de los datos* : definición de todos los elementos de datos en la BD en los tres niveles definidos anteriormente (interno, conceptual y externo). Descripción de los datos (campos, grupos, registros, tablas), e interrelaciones entre las diferentes estructuras de datos. La BD es autodescriptiva, es decir, contiene información que describe su estructura (los metadatos).
- *Manipulación de los datos* : suministrar mecanismos que faciliten la interacción con la BD. El SGBD debe ser capaz de atender las solicitudes del usuario para extraer, modificar o añadir datos a la BD. Estos mecanismos suelen venir dados en forma de lenguajes de manipulación y definición de datos. Además, garantizan la independencia de los datos, en el sentido de que, pese a la evolución del esquema de los datos, las aplicaciones deben sufrir las mínimas modificaciones imprescindibles.
- *La Seguridad y la Integridad* : debe proporcionar los medios para definir y gestionar las autorizaciones de acceso, ya sea mediante claves de acceso al sistema o mediante la definición de vistas externas de usuario, para evitar así accesos fraudulentos a los datos. Por otro lado, también proporciona los medios para garantizar la integridad y la consistencia de los datos definiendo (en el diccionario de datos) restricciones sobre los valores que pueden tomar.
- *Permitir hacer copias de seguridad* : proporciona capacidades de recuperación ante fallos y de copia de seguridad.
- *Garantiza la disponibilidad de la información* : permite el acceso simultáneo a los datos por parte de varios usuarios. Debe controlar que la información representada por los datos al final de cada acceso de usuario siga siendo coherente.
- *Interactuar con el SO* : como el SO es el único que puede acceder a los dispositivos de E/S, si el SGBD debe leer o escribir en estos dispositivos debe interactuar con el SO.

Usuarios de la Base de Datos

Las personas que trabajan con una BD se pueden catalogar como usuarios de BD. Podemos clasificar en cuatro clases los usuarios de un sistema de BD:

- *Usuarios normales* : no saben nada de la estructura interna de la BD. Interaccionan con ella a través de las aplicaciones desarrolladas por los programadores, y son incapaces de acceder a los datos directamente a través del lenguaje del SGBD. La interfaz de este tipo de usuarios es una interfaz de formularios, donde el usuario puede llenar los campos apropiados del formulario.
- *Programadores de aplicaciones* . Reciben peticiones de otros usuarios, para el acceso a los datos, y se encargan de escribir los programas que satisfacen dichas

necesidades. Normalmente estos programas están escritos en lenguajes de programación:

- convencional (Pascal, Basic, C, Cobol, etc.), en el que se insertan órdenes especiales que es capaz de comprender el SGBD. De esta forma el SGBD suministra los datos, y el lenguaje convencional (también llamado anfitrión porque alberga las sentencias reconocidas por el SGBD) los procesa, presenta al usuario, modifica, etc.
- de cuarta generación que combinan estructuras de control imperativo, con instrucciones del lenguaje de manipulación de datos, y además incluyen características especiales para facilitar la generación de formularios y la presentación de datos en pantalla. La mayoría de los sistemas de BD comerciales incluyen un lenguaje de cuarta generación.
- *Usuarios sofisticados* : son aquellos que interactúan con la BD sin programas escritos, haciendo uso directamente del lenguaje que proporciona el SGBD. Por ejemplo, SQL.
- *Usuarios especializados* : son usuarios sofisticados que escriben aplicaciones de BD especializadas. Entre estas aplicaciones están los sistemas de diseño asistido por ordenador, sistemas de bases de conocimiento, sistemas expertos etc.

Finalizamos este apartado con la quinta clase de usuarios de BD. Son los *administradores de BD* . Usuarios especiales que son los responsables del control general del sistema desde el punto de vista técnico. Entre sus funciones cabe destacar:

- Definir el esquema conceptual, con el lenguaje de Definición de Datos.
- Definir el esquema interno. La estructura de almacenamiento y los métodos de acceso.
- Diseño físico e implementación de la BD.
- Modificar el esquema y la organización física de los datos. Para reflejar las necesidades cambiantes de la organización y para mejorar el rendimiento.
- Establecer las restricciones de seguridad, integridad y confidencialidad. Concesión de permisos y privilegios para el acceso a los datos.
- Definir los procedimientos de copia de seguridad y recuperación.
- Supervisar el rendimiento del sistema y responder a los cambios en los requerimientos.

Elementos de seguridad

El administrador debe conocer en profundidad los elementos de seguridad que suministra el SGBD y sacar el máximo partido posible de ellos. En general, se pueden tener niveles de acceso clasificados por:

- La información a que se tiene acceso. En una empresa grande y ampliamente informatizada, cada usuario debe poder acceder exclusivamente a los datos que competen a su tarea. Por tanto, debe existir un mecanismo de seguridad que restrinja el ámbito de acceso de cada usuario en función de sus competencias.
- Las operaciones que se pueden realizar sobre la información. No sólo es importante el acceder o no a los datos, sino también la forma en que este acceso se produce, en función de las características propias de la sección. En general, las operaciones que se pueden efectuar se agrupan en cuatro grandes bloques: Altas, Bajas, Modificaciones y Consultas.
- El acceso al diccionario de datos y a la estructura de la BD. Como se ha comentado anteriormente, los metadatos almacenados en el diccionario de datos y que almacenan información sobre la estructura de la BD, son gestionados, a su vez, como si de una BD especial se tratase. Sin embargo, dada su primordial importancia, su acceso debe estar muy restringido, ya que cualquier modificación puede dar lugar a resultados desastrosos en la BD: pérdida de información, corrupción en los datos, falta de integridad, etc. Por ello, es necesaria la existencia de prioridades o

privilegios especiales que sólo permitan el acceso al personal que compone la administración de la BD, que es el único capacitado para modificar estos metadatos.

Lenguajes de Bases de Datos

La interacción del usuario con la BD 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.

En función de estos requerimientos, podemos tener, fundamentalmente dos tipos de lenguajes para comunicarnos con el SGBD:

- *Lenguaje de definición de datos (LDD)* . Este lenguaje es utilizado en exclusiva por el administrador de la BD, ya que permite la construcción de sentencias que le indican al SGBD las características particulares de la BD sobre la que se está trabajando, así como la creación de nuevas BD. La creación de esquemas y su modificación, la creación y supresión de índices, la especificación de unidades de almacenamiento en los ficheros, etc.
- *Lenguaje de manipulación de datos (LMD)*. El lenguaje de manipulación de datos es el que usan los usuarios sofisticados para efectuar sus operaciones sobre la BD. Como se indicó, estas operaciones son básicamente de inserción, eliminación, modificación y consulta de datos, aunque también se pueden introducir capacidades para crear vistas de los datos que faciliten otros accesos. Los usuarios sofisticados interactúan con el SGBD a través de este lenguaje, mediante una interfaz agradable y fácil de usar. Los programadores de aplicaciones emplean el LMD dentro de un lenguaje de programación que les da potencia expresiva. Para ello, el LMD que emplean se extiende de diferentes formas para poderse integrar fácilmente en el lenguaje anfitrión, ya que ambos, LMD y lenguaje anfitrión deben poderse comunicar adecuadamente para que las aplicaciones resultantes sean simples de programar y de utilizar.

Utilización de la Base de Datos en la Organización

Las BD son ampliamente usadas en multitud de aplicaciones: Banca, Líneas aéreas, Universidades, Telecomunicaciones, Ventas, Recursos Humanos, etc. Esto significa que las BD forman una parte esencial de las empresas actuales.

Sistema de información en una organización. Componentes.

Concepto de Sistema de información

Toda organización necesita para su funcionamiento un conjunto de informaciones que han de transmitirse entre sus diferentes elementos, así como desde y hacia el exterior de la propia organización. Un sistema de información se diseña con el fin de satisfacer las necesidades de información de una organización, en la que está inmerso. El sistema de información toma datos del entorno (tanto de la organización, como de las fuentes externas) y los resultados de las operaciones sobre esos datos serán la información que dicha organización necesita para su gestión y toma de decisiones.

Componentes de un sistema de información

- Contenido: (Datos: hechos conocidos con significado implícito que pueden ser almacenados). Es el centro del sistema de información. Los datos contenidos en un sistema de información pueden ser:
 - de tipo referencial: son aquellos que contienen información acerca de donde se encuentra la información buscada.
 - de tipo factual: son aquellos que contienen la información en sí. A su vez pueden ser: estructurados y no estructurados.
- Equipo físico. Ordenadores y periféricos.
- Soporte Lógico. Incluye todo el software necesario para la implantación del sistema de información: SO, Sistemas de BD, software de comunicaciones y otros programas para tratamientos específicos.
- Administrador. Los datos y las informaciones manejadas por nuestro sistema de información han de ser gestionadas por las personas adecuadas. Tendremos, por un lado los responsables de tomar las decisiones estratégicas y políticas con respecto a la información de la empresa y por otro los responsables de dar apoyo técnico para poner en práctica estas decisiones.

Niveles de gestión y de usuarios en una organización

En toda organización hay tres niveles de gestión (operacional, táctico y estratégico) y el sistema de información debe diseñarse para satisfacer las necesidades y facilitar informaciones adecuadas a cada uno de los niveles.

En el nivel operacional, los usuarios manejan datos elementales que describen los sucesos que caracterizan las actividades de la organización. Esta información, compuesta por datos totalmente desagregados (microdatos) es necesaria para los procesos comúnmente denominados administrativos (tareas diarias y de rutina) y el volumen de datos manejado será muy grande. En este nivel situamos a los llamados sistemas transaccionales.

En el nivel táctico se definen los objetivos específicos y el control de gestión. En el nivel estratégico se definen los objetivos generales y la elaboración de planes. En los niveles táctico y estratégico, cuyos usuarios tienen necesidades de información muy distintas, obtendrán del nivel anterior (operacional), mediante procesos de elaboración adecuados (generalmente de agregación) junto con datos provenientes el exterior, las informaciones necesarias para la ayuda a la decisión.

Junto con estos niveles de gestión también se pueden distinguir 3 niveles de usuarios: personal, mandos intermedios y ejecutivos. Estos niveles se corresponden con los 3 diferentes tipos de automatización de los sistemas de negocios:

- Los PED (Procesamiento Electrónico de Datos) o DP (Data Processing) que tienen el foco de atención en el nivel operativo de almacenamiento, procesamiento y flujo de los datos, así como procesar eficientemente las transacciones y realizar informes resúmenes para los dirigentes. Típicamente el enfoque DP se usa para transformar un conjunto de datos “brutos” en la siguiente información:
 - Estadística (Ejemplo: Números que representan la media, la moda y la varianza de los datos).
 - Representaciones gráficas (Ejemplo: Histogramas, Diagramas de barras, Diagramas de pastes, etc).
- Los SIG (Sistemas de Información de Gestión) o MIS (Management Information Systems) que se caracterizan porque su foco de atención está en la información orientada a mandos intermedios, por la integración de las tareas de PED, por sus funciones en los negocios y por la generación de informes.
- Los STD (Sistema de Apoyo a la Toma de Decisiones) o DSS (Decision Support Systems), más centrados en la decisión y orientados a los altos ejecutivos.

Por otro lado los datos provenientes del nivel operacional almacenados en bases de datos, así como en otros soportes, pueden estar organizados en almacenes de datos (Data Warehouses) que sirven de base para la extracción y el descubrimiento de conocimiento en BD (KDD, Knowledge Discovering inDatabases) y para la minería de datos (Data Mining).

Los sistemas transaccionales (PED)

Concepto de transacción bajo un sistema de BBDD

Uno de los problemas más complejos que se plantea en los sistemas de BD es garantizar la consistencia de la base de datos a pesar de los fallos del sistema de la ejecución concurrente. Para dar solución a los problemas de recuperación y concurrencia, se introduce el concepto de transacción.

Una transacción es una secuencia de operaciones llevadas a cabo como una unidad lógica de trabajo simple. Para asegurar la integridad de los datos se necesita que el sistema de base de datos mantenga las siguientes propiedades de las transacciones:

- *Atomicidad* : Una transacción debe ser una unidad atómica de trabajo o todas sus operaciones se llevan a cabo o no se realiza ninguna de ellas.
- *Consistencia* : Cuando termina, una transacción debe dejar la BD en un estado consistente (suponiendo que está ejecutándose de forma aislada, sin otras transacciones concurrentes). Asegurar la consistencia es responsabilidad de los mecanismos de control de concurrencia.
- *Aislamiento* : Las modificaciones realizadas por una transacción deben aislarse de las modificaciones llevadas a cabo por otras posibles transacciones concurrentes. Una transacción debe ver los datos en el estado en el que estaban antes de que cualquier otra transacción concurrente los modificara o bien los ve tras su modificación, pero nunca en un estado intermedio. El aislamiento es una propiedad de las transacciones por lo cual una transacción ve en todo momento la BD en un estado consistente. Una transacción en ejecución no hace visibles sus datos a otras transacciones concurrentes hasta que no termina y hace permanentes sus cambios.
- *Durabilidad* : Una vez que la transacción ha terminado con éxito, sus efectos deben hacerse permanentes en la BD. Las modificaciones deben persistir incluso en caso de fallo del sistema. El SGBD garantiza que los resultados de las transacciones terminadas sobrevivan a fallos posteriores del sistema.

Suele referirse a estas propiedades como **ACID** , por sus siglas en inglés (Atomicity, Consistency, Isolation, Durability). Es frecuente también encontrar referencias a la “acidity” de una transacción.

Los programadores son los responsables de establecer el inicio y el final de cada transacción en puntos que hagan cumplir la consistencia lógica de los datos.

Es responsabilidad de SGBD proporcionar los mecanismos que garanticen la integridad física de cada transacción, aquí destacamos:

- Facilidades que protejan el aislamiento de las transacciones.
- Facilidades de registro que aseguren la durabilidad de las transacciones. Si la ejecución de la transacción es interrumpida por cualquier tipo de fallo, el SGBD es el responsable de determinar qué hacer con la transacción una vez recuperado del fallo:
 - Terminar la transacción (hacer lo que queda por hacer).
 - Abortarla (deshacer lo que se haya hecho)
- Características de gestión de transacciones que garantizan la atomicidad y la consistencia de las transacciones. Una vez que la transacción ha comenzado, esta

debe ser completada con éxito o el gestor deshace todas las modificaciones de datos realizadas por esta desde el comienzo de la transacción.

Transacciones y procesamiento multiusuario

Cuando dos o más usuarios acceden concurrentemente a una BD, el procesamiento de transacciones adquiere una nueva dimensión. Ahora el SGBD no solo debe recuperarse adecuadamente de los fallos del sistema sino que también debe asegurar que las operaciones de los usuarios o aplicaciones no interfieran unas con otras.

Idealmente, cada usuario debería poder acceder a la BD como si tuviera acceso exclusivo a ella, sin preocuparse de las acciones del resto de los usuarios. Esto se logra mediante diferentes mecanismos que se conocen como esquemas de control de concurrencia.

Sistemas OLTP. Características.

Para cualquier organización actual, la captura de datos sobre sus operaciones diarias es indispensable para su funcionamiento continuado, por lo que el almacenamiento sistemático de transacciones es una actividad común en el día a día. Los sistemas que se utilizan con tal fin son los llamados OLTP (On-Line Transaction Processing) y suelen estar constituidos por BD y sistemas on-line optimizados para la inserción de grandes volúmenes de registros, habitualmente recogidos de uno en uno. En adelante, denominaremos a los datos así recogidos como datos operativos, para distinguirlos de los que se guardan en los almacenes de datos.

En principio, la información necesaria para la gestión corporativa se podría extraer de estos sistemas y sus BD asociadas. No obstante, los sistemas OLTP tienen características que hacen que esa solución no sea la más adecuada:

- *Heterogeneidad* : viene producida por la cantidad de fuentes de las que proceden los datos operativos y se manifiesta en la falta de consistencia a la hora de elegir formatos de representación, la aparición de registros redundantes, erróneos, contradictorios o, simplemente, inútiles para la extracción de información.
- *Falta de organización* : es un producto tanto de la necesidad de efectuar el almacenamiento de forma rápida, para que no se produzcan cuellos de botella en los tiempos de respuesta de sistema, como de la dispersión de los sistemas en que se almacenan estos datos diarios. Esto conduce a que, de forma natural, los datos queden almacenados atendiendo a criterios geográficos (donde se recogen) y de transacción (que operación los produjo).
- *Inadecuación para dar respuesta a consultas complejas* : esto se debe a que las BD utilizadas están optimizadas para ofrecer grandes rendimientos en las operaciones de inserción y actualización, tanto en velocidad como en volumen. Para lograr este objetivo suelen tener un nivel de redundancia muy bajo, es decir, que distribuyen la información en gran número de tablas relacionadas entre sí y con un mínimo de información agregada o precalculada. Esta organización de las tablas, útil en BD que van a estar sufriendo constantes actualizaciones, se convierte en un obstáculo para realizar consultas complejas, ya que al estar la información dispersa entre múltiples tablas, la recuperación de la información requiere una gran cantidad de operaciones de unión natural (join), costosas tanto en recursos de máquina como en tiempo. Además, realizar una consulta compleja, cuya respuesta va a demandar gran cantidad de recursos por parte de la máquina OLTP, producirá un impacto negativo en la capacidad de procesamiento y almacenaje de nuevas transacciones. Dado el número suficiente de usuarios y consultas concurrentes de este tipo, puede suceder incluso que se inutilice el sistema. Por supuesto, el hecho mismo de que sigan llegando sin cesar nuevos registros mientras se realiza la consulta implica que sus resultados no va a ser todo lo fiables que sería de desejar.

Por estos motivos, trabajar sobre las BD operativas, que utilizan para llevar el día a día, no es la forma más eficaz de extraer información útil para la toma de decisiones, tanto por los errores que se pueden producir al tratar con datos dispersos y sin limpiar, como por la repercusión que podría tener sobre el rendimiento del sistema el trabajar directamente sobre los equipos que se usan para recoger transacciones on-line, vitales para el funcionamiento de la organización.

Sistemas de Información de Gestión (MIS)

¿Qué es un M.I.S.?

Existen varias definiciones, veamos dos de ellas:

- Conjunto de medios para reunir los datos necesarios para la gestión y difundir la información obtenida con el tratamiento de estos datos.
- Proceso por el que los datos importantes para la empresa son identificados, analizados, recolectados y puestos a su disposición.

De estas definiciones se deduce que el primer objetivo de un sistema de gestión es incrementar la “inteligencia” de los procesos del negocio y el conocimiento de los trabajadores implicados en estos procesos.

Un poco de historia. Antecedentes.

- En los primeros días de los sistemas de información vieron la luz las aplicaciones. Su diseño estaba marcado por las necesidades puntuales del día a día de diferentes departamentos. La integración entre ellas no era un objetivo.
- Pronto dio comienzo el mantenimiento de esas aplicaciones. Las aplicaciones necesitaban cambios por muchas razones: nuevos requisitos, cambios en el negocio, nuevas oportunidades.
- Al mismo tiempo que comenzaba el mantenimiento surgía la necesidad de extraer más información de estas aplicaciones. El primer intento para satisfacer esta necesidad fue la escritura de programas que listasen informes.
- La primera limitación de estos listados es que accedían a una única aplicación. Hubo que definir interfaces entre las aplicaciones para que pudiesen compartir datos entre ellas.
- La segunda limitación es que los informes había que modificarlos de forma constante.
- Se introdujeron herramientas 4GL para poder escribir y modificar informes a gran velocidad. Listaban muchos más informes, pero con los mismos problemas de antes. Se introducen entonces herramientas de extracción.
- Surge el PC. Con las herramientas de extracción los usuarios ya pueden acceder y manipular directamente la información. A medida que aumentaba la potencia de los PCs aumentaba el volumen de los datos almacenados en ellos. Luego las redes, ...
- Aquí llegados los usuarios padecen la misma falta de integración, consistencia, coherencia y las mismas limitaciones que antes de la llegada del PC. Pero esto no trae como consecuencia que decrezca la demanda de información, al contrario, ésta siempre crece.

Un cambio en la arquitectura

- El corazón del problema es que las aplicaciones están profundamente marcadas por las primeras consideraciones que dirigieron su desarrollo: las necesidades departamentales enfocadas sobre las necesidades del día a día.
- La arquitectura sobre la que se construyeron estas aplicaciones (OLTP) no es válida para soportar las necesidades de los sistemas de información de gestión actuales.

- En estos sistemas la arquitectura de datos nunca fue un objetivo del negocio. La complejidad y dinamismo de la “economía digital” ha colocado en un lugar predominante la habilidad de los gestores para ver lo que está ocurriendo, desvelando las dificultades de acceso a la información de la empresa.
- Es en este momento cuando la calidad y disponibilidad de la información se convierte en un objetivo primordial del negocio.
- ¿Cómo compensar la carencia de una arquitectura de datos?
- Creando una gran BD virtual para integrar los datos de las aplicaciones existentes, que pasarán a formar parte de esta BD una vez que hayan sido depurados y reconciliadas sus disparidades. Esto posibilitará que los datos sean utilizados para la gestión.
- La solución pasa por separar el procesamiento en dos grandes categorías:
 - Proceso operacional OLTP
 - Procesamiento para el sistema de soporte de decisiones (DSS/OLAP)

Puesta en marcha del M.I.S. dentro de la organización

La implementación consiste, en una primera fase, en el análisis de las necesidades de información a las que desea acceder cada empresa. Para ello se integrarán en el sistema todos aquellos datos operacionales necesarios, además de otras fuentes de información que sea necesario incorporar.

Definida la estructura de las BD se procederá a la carga de la información y se crearán las agregaciones de datos para mejorar el rendimiento del sistema en los procesos de consulta más habituales.

Finalmente, se incluirán en el sistema los procedimientos que permitan la actualización de la información, cuya periodicidad dependerá de las necesidades de cada usuario.

Sistemas de soporte a la decisión (DSS)

Como ya se comentó, los STD (Sistema de Apoyo a la Toma de Decisiones) o DSS (Decision Support Systems), están más centrados en la decisión y orientados a los altos ejecutivos. Las BD que soportan estos sistemas son de gran tamaño y pueden resultar minas de información para adoptar decisiones empresariales, como los artículos que debe haber en inventario y los descuentos que hay que ofrecer.

La utilización de Sistemas de Soporte a la Decisión (DSS) es muy adecuada para afrontar una variedad de situaciones al interactuar: interfaces de usuarios, que hacen que dichos sistemas se adecuen a las necesidades de los usuarios; BD, que incluyen toda la información necesaria; y finalmente; procesos de decisión que ayudan al experto en la difícil tarea de establecer posibles soluciones.

Se puede definir DSS como “programas informáticos interactivos que utilizan métodos analíticos, tales como análisis de decisión, algoritmos de optimización, programas de planificación de rutinas, etc., para el desarrollo de modelos ayudando a los creadores de decisión a formular alternativas, analizar sus impactos, e interpretar y seleccionar opciones apropiadas para la implementación”.

Los sistemas de soporte a la decisión pueden considerarse como una tercera generación de aplicaciones asistidas por ordenador. Al principio, los ordenadores mainframe fueron usados mayormente para el procesamiento de transacciones. Durante los años 70 y 80, el concepto de DSS creció y se desarrolló en los campos de búsqueda, desarrollo y práctica.

Los Sistemas de Información de Gestión (MIS) suministraron:

- Informes planificados para desarrollar bien las necesidades de información
- Informes de demandas para la información específica solicitada

- Habilidad para consultar en una BD, datos específicos

Los MIS carecían de algunos de los atributos necesarios para soportar la creación de decisión. Atributos tales como, enfoque, metodología de desarrollo, manejo de gestión de datos, uso de ayuda analítica, y diálogo entre el usuario y el sistema. El DSS se extendió y combinó la tecnología de la BD y la tecnología de modelado dando a los usuarios finales acceso a ellos. Los datos y modelos se unieron íntimamente junto con el usuario.

Arquitectura del DSS

Una manera útil de pensar en las partes de los componentes de un DSS y las relaciones entre las partes está en utilizar el diálogo, los datos, y el modelo (DDM). En esta conceptualización, hay un diálogo (D) entre el usuario y el sistema, los datos (D) que soporta el sistema, y los modelos (M) que suministra el análisis de las capacidades. Estudiamos ahora con más detalle cada una de estas partes.

El componente de Diálogo (D)

Una apreciación de la importancia del componente de diálogo se obtiene reconociendo que desde la perspectiva del usuario, el diálogo es el sistema. Lo importante es lo que el usuario debe conocer para usar el sistema, las opciones para dirigir las acciones del sistema, y las presentaciones alternativas de las respuestas del sistema. Dentro del componente diálogo destacan:

- *La Base de Conocimiento* . La base de conocimiento incluye lo que el usuario conoce acerca de la decisión y acerca de cómo usar el DSS.
- *El Lenguaje de acción* . Las acciones que el usuario realiza para controlar el DSS se describen de varias formas, dependiendo del diseño del sistema.
- *El Lenguaje de Presentación* . El PC o la estación de trabajo usada en una base autónoma, como una unidad en la red de área local, o como un terminal inteligente conectado a un mainframe tiene una significativa expansión y mejora la salida desde que está presente un DSS. Una de las mayores contribuciones del PC es su capacidad de presentación de gráficos.
- *Estilos de Diálogo* . Las combinaciones o conjuntos de opciones para implantar la base de conocimiento, el lenguaje de acción, y el lenguaje de presentación, tomados a la vez, son llamados “estilo de diálogo”.

El componente de Datos (D)

Los datos juegan un papel importante en el DSS: se acceden directamente por el usuario o son una entrada para el procesamiento de los modelos. El componente de datos es manejado en:

- *Las Fuentes de Datos* . Mientras ha crecido la importancia del DSS, llega a ser cada vez más crítico para el DSS utilizar todas las fuentes de datos importantes dentro de la organización, y también desde fuentes externas. Desde luego, el concepto de fuentes de datos debe expandirse para incluir documentos que contienen conceptos, ideas, y opiniones que son muy importantes para crear la decisión.
- *Los Almacenes de Datos* . Las BD separadas para aplicaciones de apoyo a la decisión se están desarrollando mediante la creación de almacenes de datos. Estas son BD especiales que están diseñadas para permitir a los creadores de decisión hacer sus propios análisis. También se conocen a veces como BD de información. Es un típico almacén de datos, los datos que se necesitan primero se extraen del mainframe y de otras BD. Con anterioridad a ponerlos en el almacén de datos, los datos se procesan (es decir, “se limpian”) para hacerlos más útiles para el apoyo a la decisión. Entonces los datos son mantenidos por un servidor de BD. Los administradores hacen los análisis de apoyo a la decisión, utilizando el comúnmente conocido como procesamiento analítico en línea (OLAP).

El componente Modelo (M)

El modelo suministra las capacidades de análisis para un DSS. Hay muchos tipos diferentes de modelos y varias formas en las que se pueden catalogar. Se pueden hacer distinciones importantes en base a su propósito, tratamiento de aleatoriedad, y por su aplicación o uso:

- El propósito de un modelo puede ser o la optimización o la descripción. El modelo de optimización es uno que busca identificar puntos de maximización o minimización. Un modelo descriptivo describe el comportamiento del sistema. Pero un modelo descriptivo sólo describe el comportamiento del sistema; no suiere perfeccionar las condiciones.
- Con respecto a la aleatoriedad, casi todos los sistemas son probabilísticos. Esto es, que el comportamiento del sistema no se puede predecir con seguridad porque se presenta un grado de aleatoriedad. Aunque la mayoría de los sistemas son probabilísticos, la mayoría de los modelos matemáticos son deterministas.
- Según para qué se emplean existe una variedad de modelos, tales como:
 - Los modelos estratégicos los usan los directivos para ayudar a determinar los objetivos de la organización, los recursos que se necesitan para cumplir sus objetivos, y las políticas que rigen la adquisición, el uso y la disposición de estos recursos.
 - Los modelos tácticos comúnmente son empleados para la gestión media para ayudar a atribuir y controlar el uso de los recursos de la organización.
 - Los modelos operacionales normalmente son para soportar decisiones de términos pequeños (es decir, diariamente, semanalmente) usualmente encontrados en niveles organizativos inferiores.

El área de ayuda a la toma de decisiones

Dentro de una organización el área de ayuda a la toma de decisiones puede abarcar a su vez todas o algunas de las áreas que se muestran a continuación:

- El área de procesamiento analítico en línea (Online Analytical Processing, OLAP) trata de las herramientas y las técnicas para el análisis de los datos que pueden dar respuestas casi instantáneas a las consultas que soliciten datos resumidos, aunque la BD sea extremadamente grande.
- El campo del análisis estadístico, también se incluye en la ayuda a la toma de decisiones. Los lenguajes de consulta de BD no resultan adecuados para el rendimiento de los análisis estadísticos detallado de los datos. Se han creado una serie de paquetes que ayudan en el análisis estadístico. A estos paquetes se les ha añadido interfaces con las BD para permitir que se almacenen en la BD grandes volúmenes de datos y se recuperen de forma eficiente para su análisis.
- Las técnicas de búsqueda de información intentan descubrir de manera automática las reglas y las pautas estadísticas de los datos. El campo de la minería de datos combina las técnicas de búsqueda de la información creadas por los investigadores en inteligencia artificial y los expertos en análisis estadísticos con las técnicas de implantación eficiente que permiten utilizarlas en BD muy grandes.
- Las grandes empresas tienen varios orígenes de datos que necesitan utilizar para adoptar decisiones empresariales. Para ejecutar de manera eficiente las consultas sobre datos tan diferentes, las empresas han creado los almacenes de datos. Los almacenes de datos reúnen los datos de varios orígenes bajo un esquema unificado en un solo sitio. Por tanto, ofrecen al usuario una sola interfaz uniforme para los datos.

Ventajas y desventajas de usar DSS

- **Ventajas**

- Aumento en el número de alternativas examinadas.
- Mejor entendimiento del sistema.
- Respuesta rápida a situaciones inesperadas.
- Capacidad para efectuar análisis específico.
- Comunicación mejorada.
- Control y Ahorro de costes.
- Mejores decisiones.
- Ahorro de tiempo.
- Mejor uso de elaboración de recursos informáticos.

- **Desventajas**

- Alto coste de adquisición y mantenimiento.
- Para pequeños volúmenes de información no son rentables.
- Alto grado de sofisticación requerido; tiene algo grado de incertidumbre y potencial para el error.
- El DSS no es un experto, sólo comunica a los usuarios los resultados de suposiciones y modelos correspondientes de sus constructores a problemas actuales.

Almacenes de Datos (Data Warehouse)

Ya se ha comentado que tanto los Sistemas de Soporte a la Decisión (DSS), como los Sistemas de Información de Gestión (MIS), presentan problemas para recuperar datos de las BD operacionales. Para lograr la integración de estos tipos de sistemas se deberá contar con un repositorio de datos preparado para tal fin. Este repositorio se creará bajo las características de un Data Warehouse (DW). El DW, convertirá entonces los datos operacionales en una herramienta competitiva, por hacerlos disponibles a los usuarios que lo necesiten para el análisis y toma de decisiones. Una vez definido el DW se implementarán las aplicaciones de acceso a los datos, estas aplicaciones están determinadas por las características nombradas en los sistemas MIS y DSS.

Concepto de Almacén de Datos

La tecnología de los almacenes de datos “Data Warehouse”, se encuentra dentro de la línea de evolución de las BD hacia una mayor funcionalidad e inteligencia. Las empresas actuales han visto aumentada su capacidad de generar y recoger datos (introducción de internet en las empresas, tecnologías de entrada de datos ...). Estas grandes cantidades de datos (obtenidas a un coste relativamente bajo) no aportan, en principio, información a las organizaciones. Ante esta situación se puede llegar a la siguiente conclusión: “Una organización puede ser rica en datos y pobre en información, si no sabe como identificar, resumir y categorizar los datos”.

Los encargados de adoptar las decisiones empresariales necesitan tener acceso a la información de todas las fuentes que contienen datos relevantes de la empresa. La formulación de consultas a cada una de las fuentes es a la vez engorrosa e ineficiente. Además, puede que los orígenes de datos solo almacenen los datos actuales, mientras que es posible que los encargados de adoptar las decisiones empresariales necesiten tener acceso también a datos anteriores. Los almacenes de datos proporcionan una solución a estos problemas.

El almacén de datos pretende dar un soporte a la organización para proporcionarle una buena gestión de sus datos, que le ayude en la toma de decisiones estratégicas y tácticas.

Antecedentes de los Almacenes de Datos

A finales de los 80 aparece el DRI (Diccionario de recursos de información), cuyos antecedentes son:

- Directorio de datos: Componente del SGBD encargado de describir donde y como se almacenan los datos, y modo de acceder a los datos contenidos en la BD.
- Diccionario de datos: Reúne la información sobre los datos almacenados para que los usuarios comprendan su significado.
- Diccionario / Directorio de datos: Que aúna las dos tareas anteriores.
- Enciclopedia o Repositorio: Donde se almacenan los datos generados durante el ciclo de vida de un SI (Sistema de Información): esquemas, información relativa a la gestión de proyectos, etc.
- Diccionario de recursos de información: Engloba las capacidades y funciones de todos los “almacenes” de datos anteriores. Pretende ser el corazón de toda arquitectura de información de la empresa, sirviendo de soporte para la integración de sistemas.

Los almacenes de datos recogen la herencia de los DRI con algunos matices derivados de la nueva tecnología y de la experiencia.

¿Qué es un Data Warehouse?

Vamos a dar varias definiciones, para entender mejor lo que es un almacén de datos:

- Depósitos de información reunida de varios orígenes, almacenada bajo un esquema unificado en un solo sitio. Una vez reunida, los datos se almacenan mucho tiempo, lo que permite el acceso a datos históricos. Así, los almacenes de datos proporcionan a los usuarios una sola interfaz consolidada con los datos, lo que hace más fáciles de escribir las consultas de ayuda a la toma de decisiones.
- Un Data Warehouse (DW a partir de ahora) es una nueva arquitectura informática para dar soporte a la obtención de información relevante. Combina potentes herramientas de modelado multidimensional con herramientas de acceso a BD, contribuyendo, no sólo a mostrar los hechos (datos), sino a comprender las causas de los hechos.
- Un DW es la creación de una vista lógica unificada de los datos, aún cuando estos estén dispersos entre varias BD físicas, para así disponer de un único modelo de trabajo de los datos de la organización.
- Se puede considerar un DW como un repositorio lógico central (aunque los componentes físicos pueden estar distribuidos), que almacena los datos de la organización a diferentes niveles (desde el más bajo del dato puro hasta los niveles más altos que contienen agregados o resúmenes de los datos de niveles inferiores), que solo contiene datos relevantes para la toma de decisiones y que está optimizado para permitir el análisis y la recuperación de información corporativa.

Características

Esta colección de datos tiene las siguientes características:

- *Orientado a las materias* : Se centra en entidades de alto nivel (como por ejemplo cliente, producto, ...) y no en los procesos (como hacen los sistemas operacionales).
- *Integrado* : La integración implica que los datos del almacén son consistentes al elegir convenciones en nombres, unidades de medida, representación de campos comunes, etc. Se construye mediante la integración de fuentes de datos múltiples, y heterogéneas: BD relacionales, ficheros planos, registros de transacciones on-line. Se aplican técnicas de limpieza e integración para asegurar la consistencia en el nombrado, estructuras codificadas, medidas de los atributos, y demás aspectos entre las múltiples BD. Cuando los datos se mueven al DW, éstos se tienen que transformar.

- *No volátil* : Los datos no cambian una vez que se encuentran en el almacén. Las únicas operaciones que permite un almacén de datos con la carga de nuevos datos y el acceso a los ya almacenados.
- *Sirve de soporte a consultas de ayuda a la decisión* .
- *Es dependiente del tiempo* : Los datos están asociados a un instante en el tiempo (semestre, año). Por lo tanto, los datos representan una imagen estática del estado de la organización en cada momento. Es decir, que si, por ejemplo, se accede a los datos de hace un mes de obtendrán los datos que describían la organización en aquel momento, sin que se hayan modificado de ninguna manera.

Objetivos

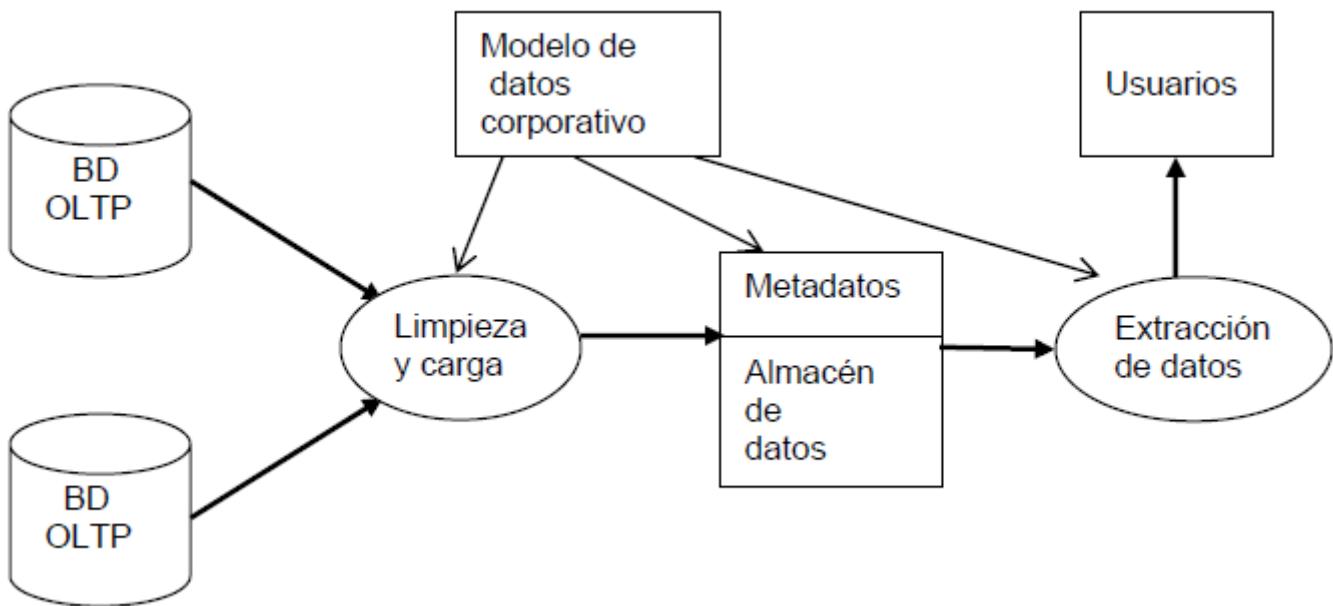
Después de ver las características de un DW, podemos ahora comprender mejor cuales son sus objetivos:

- Debe conseguir que la información sea fácilmente accesible para una organización y sus contenidos comprensibles. Los datos deben ser intuitivos y obvios para el usuario y no sólo para el desarrollador. Los contenidos del DW deben estar etiquetados con nombres comprensibles. Las herramientas de acceso al DW deben ser fáciles de usar y ofrecer resultados en el mínimo tiempo posible.
- Debe presentar a la organización información consistente y “creíble”. La información debe ser cuidadosamente agrupada según los orígenes de datos sin inconsistencias o duplicidades.
- Debe adaptarse a los cambios y al crecimiento. Las necesidades de usuario, las condiciones del negocio, los datos y la tecnología cambian inevitablemente a lo largo del tiempo. Estos cambios no deben en ningún caso invalidar la información que había hasta el momento en el DW.
- Debe proteger la información relevante, sensible y confidencial de una organización. Los accesos al DW deben mostrar sólo la información relevante a aquellas personas que están autorizadas para verla.
- Debe ser la base para tomar decisiones dentro de las organizaciones. Es decir, la información necesaria para tomar ciertas decisiones, operacionales, tácticas y estratégicas debe estar contenida en el DW.

Componentes de un almacén de datos

La estructura básica de un sistema de almacén de datos está compuesta por un modelo de datos corporativo, que representa la vista conceptual de los datos de la organización, un procedimiento de limpieza e inserción de los datos operativos en el almacén, por el propio almacén junto con unos metadatos que proporcionan información sobre su contenido y, finalmente por unos procesos de extracción de información que son las herramientas de minería de datos y OLAP.

En la siguiente figura se muestran estos componentes y las relaciones entre ellos:



Vamos a presentar a continuación cada uno de estos componentes, su funcionamiento y cómo encajan en el proceso global de construcción y mantenimiento de un almacén de datos.

Modelo de datos corporativos

Es probable que los orígenes de datos que se han creado de manera independiente tengan esquemas diferentes. Parte de la labor de los almacenes de datos es llevar a cabo la integración de los esquemas y convertir los datos a un único modelo de datos corporativo. Este modelo establece un único esquema lógico de datos para toda la organización, evitando la fragmentación producida por la existencia de sistemas de información departamentales.

Uno de los fundamentos de un DW y paso previo obligatorio a emprender su construcción, es determinar un modelo de datos corporativo que identifique y estructure los requisitos de información del almacén. Como resultado se obtiene un esquema conceptual válido para toda la organización, que ofrece una visión estratégica global de los datos, permitiendo a la vez la creación de vistas parciales con el grado de detalle adecuado a las necesidades de cada departamento.

Esta estrategia “top-down” supone la construcción de un modelo de datos que comprenda todas las entidades y objetos manejados por una organización, lo que es una tarea suficientemente compleja como para que esta sea una de las principales causas de fracaso en la implantación de DW. Para disminuir el impacto negativo de esta tarea, se ha creado el concepto de Supermercado de Datos (Data Marts) que no son sino el resultado de aplicar una estrategia “bottom-up”, esto es, comenzar construyendo almacenes de datos departamentales, con sus correspondientes modelos de datos, antes de construir el almacén de datos corporativos.

La creación del modelo de datos corporativo es, por lo tanto, la etapa del diseño del DW. Se trata de definir:

- Qué información, relevante para los usuarios, se va a incluir en el DW.
- Qué datos se precisan para obtener la información. La mayoría vienen de las BD de producción pero otros pueden venir de fuentes externas.
- Representación de la información: nombres, formatos, unidades, etc. La descripción detallada de la información del DW está en el diccionario de datos, éste suele incluir:
 - Identificación de la fuente de datos.
 - Estructura, unidades, precisión, etc, de los datos.
 - Estructura, unidades, precisión, etc, de la información.

- Estructura de la información que ven los usuarios finales.
- Normas para encontrar, limpiar, transformar y agregar los datos para transformarlos en información.
- Normas de seguridad aplicables a los datos y a la información.

Limpieza y carga de datos operativos

Identificación de las fuentes de datos

Esta operación sirve para reconocer a qué sistemas (datos propios de la empresa, fuentes externas ...) hay que acceder para conseguir los datos que se van a introducir en el almacén. Esto permitirá identificar los sistemas OLTP afectados, el volumen de datos que se va a capturar, la frecuencia con la que se deberá realizar la captura y los departamentos o áreas de la organización que se verán afectados por el proyecto de creación del DW.

Limpieza de los datos operativos

El proceso de limpieza de los datos importados, consiste en:

- El descubrimiento y la corrección de inconsistencias, errores, repeticiones o cualquier otra anomalía en los datos que se van a introducir en el almacén.
- Transformación de los términos operacionales en términos de negocio uniformes, estándar y auto-explicativos.
- Definición física de un atributo: usar tipos de datos y longitudes significativas.
- Uso consistente de los valores de los atributos de las entidades: valores diferentes pero que signifiquen lo mismo, se convierten a un único valor.
- Asuntos relacionados con valores por defecto y valores perdidos: si no se tiene claro es más seguro dejar estos valores en blanco.

Documentación de los formatos

Es necesario documentar los formatos de los datos que se van a transferir al almacén, para ello hay que tener en cuenta el significado de cada uno de ellos, el procedimiento que se ha seguido para su obtención a partir de los datos operativos originales y los procesos de summarización o agregación que se les ha aplicado.

Los datos que conforman esta información acerca del contenido del propio almacén reciben el nombre especial de metadatos que dan lugar al Diccionario de datos. Forman lo que se podría considerar el manual técnico del almacén y son imprescindibles para realizar el mantenimiento del almacén e interpretar adecuadamente los resultados de las consultas que se realicen sobre el mismo.

Transformación y carga

Este es el último paso y consiste, lógicamente, en la transformación y carga efectiva de los datos una vez procesados y documentados. En este proceso se incluye la inserción en el almacén tanto de los datos operativos procesados y limpiados, como de los metadatos que sirven para documentarlos.

Esta tarea debe realizarse periódicamente para mantener un grado de sincronización aceptable entre el contenido del almacén y el de las bases de datos OLTP, ya que sino la información proporcionada por las herramientas de extracción de datos no sería lo suficientemente actual como para ser útil.

Por otro lado, el proceso de carga suele ser largo y costoso, ya que hay que procesar una gran cantidad de registros, lo que hace necesario buscar un equilibrio entre el deseo de mantener actualizado el DW y el coste que supone en recursos máquina.

Resultado: Almacén de datos

Tras realizar todas estas etapas, se obtiene como resultado un almacén que contiene unos datos tratados, documentados y listos para ser utilizados como materia prima de las herramientas de extracción de información.

Extracción y recuperación de los datos

Acceso a los datos

El DW debe ofrecer soluciones a los problemas de acceso a los datos. Algunos de estos problemas suelen ser:

- Los datos están en sistemas a los que el usuario no puede acceder.
- El usuario no tiene herramientas para leer correctamente los datos y para ello se le ofrecen una serie de herramientas de varios tipos: Visualización de datos, Análisis estadístico y Generadores de informes.
- Los datos podrían estar siendo usados por aplicaciones que impiden su utilización por otras aplicaciones.

Recuperación de los datos

Un DW debe contener toda la información sobre el tema correspondiente y debe contar con mecanismos para recuperarla. Existen tres conceptos fundamentales, que ayudan a llevar a cabo dicha recuperación, son:

- Base de datos. Una BD sobre un tema determinado es un conjunto de datos sobre dicho tema que cumple los criterios de: Exhaustividad, Ausencia de redundancia y Estructura adecuada.
- Diccionario de datos. Define los datos, se sabe cuáles existen, qué significa cada elemento de datos. Además determina el tema y los criterios que deben cumplirse para que los datos sean exhaustivos.
- Complementación relacional. Garantiza la recuperación de cualquier subconjunto de la información, basándose en cinco operadores: selección, proyección, intersección, unión y diferencia.

Complejidad de las consultas

Cuando se realizan consultas complejas a BD de producción, estas deben ser hechas por especialistas que saben cómo y dónde buscar esos datos. El DW elimina la intervención de estos especialistas, eliminando los posibles problemas de disponibilidad.

Análisis multidimensional

Con el análisis multidimensional se da respuesta a consultas complejas de los usuarios que reflejan los diversos componentes de sus organizaciones. Estos componentes pueden ser de dos tipos: Cuantitativos o Cualitativos.

A estos componentes también se les llama dimensiones, y a los valores de los componentes o dimensiones se les llama atributos. Además el detalle con el que se muestran los atributos puede variar, cada dimensión se puede descomponer en diferentes niveles de detalle, estos dependen de las necesidades del usuario.

Las dimensiones definen dominios como geografía, producto, tiempo, cliente, etc. Los miembros de una dimensión se agrupan de forma jerárquica (dimensión geográfica: ciudad, provincia, autonomía, país...).

El usuario puede navegar por los datos de diferentes maneras:

- Perforación (drill-down): Consisten en variar el nivel de detalle de los datos, desde los datos más resumidos a los más detallados. Se dice que drill-down es desagregar y Roll-up es agregar.
- Segmentación (slicing and dicing): Consisten en “recortar” un subconjunto de los datos moviéndose por los distintos datos de una misma dimensión o cambiando de dimensión. Es decir, es la capacidad de ver la BD desde diferentes puntos de vistas. El corte suele hacerse a lo largo del eje del tiempo para analizar tendencias. Se dice que slicing es proyección y que dicing es selección.

Estructura lógica del almacén: datos y metadatos

La estructura lógica de un almacén se encuentra dividida en cuatro niveles más uno adicional de metadatos:

METADATOS
DATOS DETALLADOS ACTUALES
DATOS DETALLADOS HISTÓRICOS
DATOS LIGERAMENTE RESUMIDOS
DATOS MUY RESUMIDOS

Veamos a continuación cada uno de estos niveles.

Metadatos

Este nivel no es el superior jerárquico de los otros cuatro, sino que se encuentra completamente aparte del resto. Esto se debe a que no está compuesto por datos extraídos a partir de sistemas OLTP, sino por la descripción del tratamiento a que se han sometido dichos datos originales.

Los metadatos describen la estructura de los datos contenidos en el almacén, de donde proceden y que tratamiento sufrieron. También detallan los algoritmos utilizados para crear los resúmenes de los dos niveles superiores de la estructura del almacén (datos ligeramente resumidos y datos muy resumidos). Esta información, será de utilidad para las herramientas de extracción de información, que la usarán para determinar estrategias de navegación y recuperación.

Datos detallados actuales

Son los datos obtenidos directamente al limpiar y homogeneizar los datos provenientes de sistemas OLTP. Constituyen el nivel más bajo de detalle, representan el estado de la organización en en momento presente y, debido a que están sin resumir, constituyen una gran porción del volumen total de los datos almacenados.

Estos datos son de acceso frecuente, ya que son los más actualizados, por tanto, es conveniente que se almacenen en dispositivos de acceso rápido como discos.

Datos detallados históricos

Son los datos detallados correspondientes a momentos anteriores al presente, por lo que el nivel de detalle es el mismo que el de los datos actuales. Al no ser datos a los que se deba acceder con frecuencia, se almacenan en cintas o cualquier otro dispositivo de almacenamiento masivo.

Datos ligeramente resumidos

Es el primer nivel de agregación de los datos detallados actuales. Corresponden a consultas o informes de uso habitual, por lo que al tenerlos preparados de antemano se consigue acelerar considerablemente el rendimiento global del almacén. Es importante identificar sobre qué variables se van a realizar estos resúmenes así como su frecuencia de actualización.

Datos muy resumidos

Representan el nivel más elevado de agregación, tanto de los datos ligeramente resumidos como de los de detalle. Corresponden a consultas o informes que se solicitan muy a menudo y que deben obtenerse con gran rapidez. Dado el alto grado de accesibilidad que deben tener estos datos muy resumidos es normal encontrarlos fuera del almacén de datos corporativo, formando parte de los almacenes de datos departamentales o Data Mart.

Estructura física del almacén: arquitectura

La estructura física del almacén puede presentar cualquiera de las siguientes arquitecturas:

Arquitectura centralizada

Consiste en utilizar un único servidor para guardar todo el almacén de datos.

La ventaja de esta configuración reside en que maximiza la potencia de cálculo disponible para trabajar sobre el almacén y facilita el mantenimiento del mismo.

La desventaja estriba en que la realización de consultas que consumen muchos recursos puede afectar seriamente al resto de usuarios que sólo necesiten acceder a datos de alto nivel (resumidos o muy resumidos). Además, un fallo en este servidor puede resultar catastrófico para la organización, por lo que la seguridad del mismo cobra una especial relevancia.

Arquitectura distribuida

Esta segunda opción se basa en la existencia de varios servidores entre los que se reparten los datos del almacén. Dado que una de las características del almacén es que está organizado en torno a temas, resulta lógico que la distribución física de los datos refleje esta propiedad, asignando así cada servidor a uno o varios temas lógicos.

La ventaja de esta arquitectura es una mayor distribución de la carga de proceso a cambio de una mayor complejidad en el mantenimiento de la estructura del almacén. También sigue presentando el problema de la no-discriminación de los datos de más alto nivel de los de menor nivel, por lo que una operación que requiera muchos recursos máquina seguirá bloqueando el acceso al resto de usuarios.

Arquitectura distribuida por niveles

Esta arquitectura refleja la estructura lógica del almacén, ya que asigna los servidores en función del nivel de agregación de los datos que contienen. De esta manera se tendrá un servidor para los datos de detalle, otro para los resumidos y otro para los muy resumidos.

Un caso particular se presenta cuando los datos muy resumidos no están en un único servidor, sino que se duplican en varios para agilizar el acceso a los mismos. En este caso los servidores que los mantienen son Data Marts. La ventaja de esta arquitectura es que permite un acceso rápido a los datos que se utilizan con más frecuencia, sin que sea penalizado por las consultas que se realicen sobre datos de detalle.

Comparación de DW y BD

Diferencias entre un sistema de BD tradicional y un DW:

	BD	DW
Contenido	Datos aptos para la producción.	Información para la toma de decisiones.
Modelo de datos	Jerárquico, CODASYL, relacional o incluso secuencial de índice.	Relacional, multidimensional
Utilización	Producción de OLTP, por lotes.	Toma de decisiones.
Acceso	Lectura y escritura con frecuencia aleatoria.	Lectura con frecuencia secuencial.
Nº operaciones/ Nº de elementos afectados	Numerosas transacciones, y cada una afecta a pocos registros.	Pocas consultas y cada una puede afectar a muchos registros.
Volumen de salida de operaciones	(transacciones) normalmente pequeño.	(consultas) puede ser muy grande.
Tiempo improductivo	Provoca pérdidas empresariales.	Demora en las decisiones.

Elementos básicos de un DW

Sistema Origen

Es un sistema operacional cuya función es capturar las transacciones del negocio. A menudo se le llama "legacy system". Las principales características de este sistema son la disponibilidad y la actualidad de su información.

Asumiremos que el sistema origen mantiene poca información histórica y que se realizará un cierto tipo de reporting directamente sobre el sistema fuente que no tiene porque estar recogido en el propio DW.

Los sistemas fuentes son normalmente independientes entre sí, por lo que seguramente en ellos no se habrá invertido para conformar dimensiones básicas como los productos, los clientes, la geografía o el tiempo.

Los sistemas origen tienen claves que identifican ciertos objetos del análisis de manera única, como por ejemplo la clave de producto o la clave de cliente. A las claves de estos sistemas fuente se les denomina "claves de producción", pero no las usaremos directamente en el DW como claves, trataremos estas claves como atributos de las dimensiones que se creen.

Data staging área

Es un área de almacenamiento y un conjunto de procesos que limpian, transforman, combinan, unifican, almacenan, archivan y preparan la fuente de datos para su uso en el DW. Está compuesta por un sistema de ficheros planos.

Consiste en todo aquello que hay entre el origen de datos y el servidor de presentación. Aunque sería ideal que incluso consistiera en una única máquina, la realidad es que está distribuida por un conjunto de máquinas.

Esta área está dominada por las actividades de ordenación y procesamiento secuencial y en algunos casos ni siquiera tiene que estar basada en tecnología relacional. Una restricción de esta área es que no se utilizará en ningún caso para hacer consultas directamente sobre ella o para generar informes a partir de ella misma.

Servidor de presentación (Presentation server)

Es la máquina física destino donde el DW es organizado y almacenado para ser directamente consultado por los usuarios finales y otras aplicaciones.

En este sistema los datos se presentarán y almacenarán en un marco multidimensional. Si el servidor está basado en una BD relacional, entonces las tablas estarán organizadas como “esquemas en estrella”. Si el servidor de presentación está basado en una tecnología de procesamiento analítico on-line no relacional (tecnología OLAP), los datos tendrán dimensiones reconocidas como tal.

ESQUEMA - RESUMEN

Una BD es una colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos y su definición (estructura de la BD), única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos.

En los sistemas de BD se plantean dos objetivos principales:

- Independencia de la BD de los programas para su utilización.
- Proporcionar a los usuarios una visión abstracta de los datos. El sistema esconde los detalles de almacenamiento físico (como se almacenan y se mantienen los datos), pero estos deben extraerse eficientemente. Niveles de abstracción: externo, conceptual e interno.

Principales componentes de un entorno de BD:

- **Datos** : una BD no tiene sentido si no está compuesta por datos. Hay que definir la forma en que estos datos se deben disponer, qué datos se deben almacenar y cómo los debe entender la máquina.
- **Metadatos** : la información que el sistema guarda sobre los datos almacenados, es lo que se llaman metadatos, es decir, datos acerca de los datos. Es más, estos metadatos se almacenan como otra BD propiamente dicha, y puede ser gestionada y consultada como tal. Estos metadatos suelen conformar lo que se da en llamar diccionario de datos o catálogo.
- **El Sistema Gestor de BD** : se puede definir como un conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra, tanto a los usuarios finales como a los analistas, programadores o el administrador, los medios necesarios

para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad.

- **Usuarios de la BD** : usuarios normales, programadores de aplicaciones, usuarios sofisticados y especializados y finalmente los administradores de BD.
- **Lenguajes de BD** : Lenguaje de Definición de Datos (LDD) y Lenguaje de Manipulación de Datos (LMD).

En cuanto a la utilización de las BD en las organizaciones, hablamos de 3 diferentes tipos de automatización de los sistemas de información:

- Los PED (Procesamiento Electrónico de Datos) o DP (Data Processing) que se caracterizan por tener el foco de atención en el nivel operativo de almacenamiento, procesamiento y flujo de los datos, así como procesar eficientemente las transacciones y realizar informes resúmenes para los dirigentes.
- Los SIG (Sistemas de Información de Gestión) o MIS (Management Information Systems) que se caracterizan porque su foco de atención está en la información orientada a mandos intermedios, por la integración de las tareas de PED, por sus funciones en los negocios y por la generación de informes.
- Los STD (Sistema de Apoyo a la Toma de Decisiones) o DSS (Decision Support Systems) que están más centrados en la decisión y orientados hacia altos ejecutivos.

En los sistemas transaccionales basados en BD, una transacción es una secuencia de operaciones llevadas a cabo como una unidad lógica de trabajo simple. Para asegurar la integridad de los datos se necesita que el sistema de BD mantenga las siguientes propiedades de las transacciones: Atomicidad, Consistencia, Aislamiento y Durabilidad. Son conocidas como propiedades ACID.

Dos definiciones de un Sistema de Información de Gestión (MIS):

- Un Sistema de Información de Gestión puede definirse como un conjunto de medios para reunir los datos necesarios para la gestión y difundir la información obtenida con el tratamiento de estos datos.
- Definimos Sistema de Información de Gestión como el proceso por el cual los datos que son importantes para la empresa son identificados, analizados, recolectados y puestos a disposición de la empresa.

Se pueden definir los Sistemas de Soporte a la Decisión (DSS) como programas informáticos interactivos que utilizan métodos analíticos, tales como análisis de decisión, algoritmos de optimización, programas de planificación de rutinas, etc., para el desarrollo de modelos ayudando a los creadores de decisión a formular alternativas, analizar sus impactos, e interpretar y seleccionar opciones apropiadas para la implementación.

Una manera útil de pensar en las partes de los componentes de un DSS y las relaciones entre las partes está en utilizar el diálogo, los datos, y el modelo (DDM). En esta conceptualización, hay un diálogo (D) entre el usuario y el sistema, los datos (D) que soporta el sistema, y los modelos (M) que suministra el análisis de las capacidades.

Los DSS abarcan diversos campos como: procesamiento analítico en línea (Online Analytical Processing, OLAP), análisis estadístico, minería de datos y almacenes de datos.

Se define un DW como un repositorio lógico central (aunque los componentes físicos pueden estar distribuidos), que almacena los datos de la organización a diferentes niveles (desde el más bajo del dato puro hasta los niveles más altos que contienen agregados o resúmenes de los datos de niveles inferiores), que solo contiene datos relevantes para la toma de decisiones y que está optimizado para permitir el análisis y la recuperación de información corporativa.

Los almacenes de datos presentan las siguientes características: es orientado a materias, integrado, no volátil, sirve de soporte a consultas de ayuda a la decisión y es dependiente del tiempo. Además el DW busca los objetivos siguientes:

- Debe conseguir que la información sea fácilmente accesible para una organización.
- Debe presentar a la organización información consistente y “creíble”.
- Debe adaptarse a los cambios y al crecimiento.
- Debe proteger la información relevante, sensible y confidencial de una organización.
- Debe ser la base para tomar decisiones dentro de las organizaciones.

Componentes que forman parte de un DW:

- **Modelo de datos corporativo** : La creación del modelo de datos corporativo es la etapa del diseño del DW, que identifica y estructura los requisitos de información que va a tener que satisfacer el almacén.
- **Limpieza y carga de datos operativos** : Este proceso está compuesto de las siguientes etapas:
 - Identificación de las fuentes de datos.
 - Limpieza de los datos operativos.
 - Documentación de los formatos.
 - Transformación y carga.
- **El almacén de datos** : se obtiene como resultado un almacén que contiene unos datos tratados, documentados y listos para ser utilizados como materia prima de las herramientas de extracción de información.
- **Extracción y recuperación de los datos** : En este proceso hay que tener presente los puntos siguientes:
 - Acceso y Recuperación de los datos.
 - Complejidad de las consultas.
 - Análisis multidimensional.

La estructura lógica de un almacén se encuentra dividida en cuatro niveles más uno adicional de metadatos que está por encima de ellos. Los cuatro niveles son:

- Datos detallados actuales
- Datos detallados históricos
- Datos ligeramente resumidos
- Datos muy resumidos

La estructura física del almacén puede presentar cualquiera de las siguientes arquitecturas:

- **Arquitectura centralizada** : Consiste en utilizar un único servidor para guardar todo el almacén de datos.
- **Arquitectura distribuida** : Esta segunda opción se basa en la existencia de varios servidores entre los que se reparten los datos del almacén. Dado que una de las características del almacén es que está organizado en torno a temas, resulta lógico que la distribución física de los datos refleje esta propiedad, asignando así cada servidor a uno o varios temas lógicos.
- **Arquitectura distribuida por niveles** : Esta arquitectura refleja la estructura lógica del almacén, ya que asigna los servidores en función del nivel de agregación de los datos que contienen. De esta manera se tendrá un servidor para los datos de detalle, otro para los resumidos y otro para los muy resumidos.

Sistemas de gestión de bases de datos relacionales: características y elementos constitutivos. Antecedentes históricos. El lenguaje SQL. Estándares de conectividad: ODBC y JDBC.

Antecedentes históricos

Se suele hablar de tres generaciones en la historia de las BD, son:

- Primera generación: sistema jerárquico y sistema de red.
 - Requieren complejos programas de aplicación.
 - La independencia de datos es mínima.
 - No tienen un fundamento teórico.
- Segunda generación: modelo relacional.
 - Lenguaje de consultas estructurado: SQL.
 - Desarrollo de SGBD relacionales comerciales.
 - Limitada capacidad para modelar datos.
- Tercera generación: modelo orientado a objetos y modelo relacional extendido.

Veamos ahora con más detalle la historia de cada uno de estos modelos, soportados en diferentes SGBD.

Las Bases de Datos Jerárquicas

A finales de los 60, coincidiendo en el tiempo con el desarrollo de los sistemas gestores de archivos, IBM y North American Aviation desarrollan el modelo jerárquico. Con la finalidad de resolver problemas de diseño aeroespacial y de producción se desarrolla Information Management System (IMS) con su lenguaje DL/1. Fue el primer sistema de gestión de BD comercial basado en el modelo jerárquico. Aparece IMS DB/DC (Database/Data Communication), el primer sistema de BD de gran escala.

Sobre 1969, IMS dio como resultado un sistema de gestión de BD de tipo jerárquico de propósito general: el IMS/1 de IBM que constituye la primera familia de sistemas de gestión de BD. American Airlines e IBM desarrollan SABRE, el primer sistema que proporciona acceso a datos compartidos por múltiples usuarios a través de una red de comunicación.

Las Bases de Datos en Red

A mitad de los sesenta, se desarrolló IDS (Integrated Data Store), de General Electric. Este trabajo fue dirigido por uno de los pioneros en los sistemas de BD, Charles Bachman. IDS era un nuevo tipo de sistema de BD conocido como estructura en red, que produjo un gran efecto sobre los sistemas de información de aquella generación. El sistema en red se desarrolló, en parte, para satisfacer la necesidad de representar relaciones más complejas entre datos que las que se podían modelar con los sistemas jerárquicos, y, en parte, para imponer un estándar de BD.

Para ayudar a establecer dicho estándar, CODASYL (Conference on Data Systems Languages), formado por el gobierno de EEUU y representantes del mundo empresarial, organiza el grupo DBTG (Data Base Task Group), para definir especificaciones estándar

que permitan la creación y el manejo de BD. El DBTG presentó su informe final en 1971 y aunque no fue formalmente aceptado por ANSI (American National Standards Institute), muchos sistemas se desarrollaron según la propuesta del DBTG. Estos sistemas se conocen como sistemas en red, sistemas CODASYL o DBTG.

Los modelos jerárquico y de red constituyen la primera generación de los sistemas de BD, pero presentan algunos de los siguientes inconvenientes: no tienen un fundamento teórico, la independencia de datos es mínima y es necesario escribir complejos programas de aplicación para cualquier consulta de datos, por simple que sea.

En la década de los 70, la tecnología de BD experimenta un rápido crecimiento. Algunos sistemas, desarrollados a lo largo de los años 70, que siguen las propuestas de CODASYL son: DMS-1.110 de UNIVAC, DMS-170 de CDC, IDMS de DF Goodrich, DBMA-11 de DIGITAL, etc. Sin embargo ninguna de estas implementaciones desarrolló completamente las propuestas de CODASYL.

El modelo de datos en red siempre tuvo pretensiones de generalización y estandarización, mientras que la familia de sistemas jerárquicos está constituida por una serie de sistema de gestión de BD de los que posteriormente se obtuvo la abstracción del modelo de datos jerárquico. Ambos tipos de SGBD eran accesibles desde un lenguaje de programación, usualmente Cobol, usando un interfaz de bajo nivel. Esto hacía que la creación de una aplicación, el mantenimiento de la BD, así como el ajuste y el desarrollo fuesen controlables, pero aún a costa de una gran inversión de tiempo.

Hasta 1980 los modelos de red y jerárquico fueron populares. Cullinet, una empresa fundada por Bachman, fue la mayor empresa de software y con más rápido crecimiento en el mundo, en aquellos años.

Las Bases de Datos Relacionales

A pesar del éxito del modelo de datos en red, muchos diseñadores de software reconocieron que la interfaz de programación para navegación por los registros era de demasiado bajo nivel.

En 1970 E.F.Codd, basándose en el álgebra y la teoría de conjuntos, propone un nuevo modelo de datos llamado modelo relacional. Sugiere que todos los datos de la BD se podrían representar como una estructura tabular (tablas con columnas y filas, que denominó relaciones) y que esas relaciones se podrían acceder con un lenguaje no procedimental (declarativo). En este tipo de lenguajes, en lugar de escribir algoritmos para acceder a los datos, sólo se necesita un predicado que identifica los registros o combinación de registros deseados. Es más, este nuevo modelo integraba los lenguajes de definición, navegación y manipulación en un solo lenguaje unificado.

El modelo relacional encontró inicialmente una gran oposición debido a que requería más recursos informáticos que los SGBD existentes en la época y sus implementaciones no estaban lo suficientemente refinadas como para competir con el resto de modelos y, por tanto, resultaban demasiado lentos.

Los SGBD relacionales no fueron prácticos hasta la década de los ochenta en que se desarrollaron computadores más rápidos y a menor precio.

Los programadores se debieron adaptar a una nueva forma de pensar en el tratamiento de los datos. Hasta ahora los programadores estaban acostumbrados a procesar los datos en registro, en lugar de procesar simultáneamente los datos.

Se desarrollaron proyectos de investigación que dieron lugar a algunos prototipos entre los que destacan:

- INGRES de la Universidad de Berkeley (1973-1975)

- System R de IBM (1974-1977)
- System 2000 de la Universidad de Austin en Texas
- El proyecto Sócrates de la Universidad de Grenoble en Francia
- ADABAS de la Universidad técnica de Darmstadt en Alemania

Durante este periodo se desarrollaron diversos lenguajes de consulta: SQUARE, SEQUEL (SQL), QBE y QUEL. De fundamental importancia es el lenguaje SQL, que fue el resultado de la convergencia de muchos de los prototipos desarrollados en la época.

El trabajo de investigación en IBM conducido por Ted (E.F.) Codd, Raymond Boyce y Don Chamberlain y el trabajo en la Universidad de Berkeley conducido por Michael Stonebraker, dieron como resultado SQL. Se estandarizó por primera vez en 1986 por el comité ANSI X3H2 como estándar de ANSI, que fue denominado SQL-86. ANSI publicó un estándar extendido en 1989, SQL-89. La siguiente versión del estándar fue SQL-92 y la más reciente SQL-99.

Ya la primera estandarización de SQL, provocó la desaparición de su más inmediato competidor, QUEL. Sin embargo, QBE ha sobrevivido hasta nuestros días gracias a las interfaces de usuario amigables y porque supone un primer contacto más intuitivo y rápido con las BD relacionales.

Posteriormente a los prototipos aparecieron numerosos sistemas relacionales comerciales, tales como: INGRES de RTI (1980), SQL/DS de IBM (1981), ORACLE de RSI (1981), DB2 de IBM (1983), RDB de DIGITAL (1983), etc.

En la década de los 80 se desarrolla SQL Server en Sybase para sistemas UNIX y posteriormente se transportó a sistemas Windows NT. Desde 1994 Microsoft ha lanzado nuevas versiones de este producto de BD independientemente de Sybase, que dejó de usar el nombre SQL Server a finales de los 90.

El modelo de datos relacional ha proporcionado beneficios inesperados además del aumento de productividad y facilidad de uso. Es muy adecuado para el enfoque cliente/servidor, el procesamiento paralelo y las interfaces gráficas de usuario.

El modelo relacional constituye la segunda generación de los sistemas de BD. Hoy en día, existen cientos de SGBD relacionales, tanto para ordenadores personales como para sistemas multiusuario, aunque muchos no son completamente fieles al modelo relacional.

El modelo relacional también tiene sus fallos, siendo uno de ellos su limitada capacidad para modelar los datos. En 1976, Chen presentó el modelo entidad-relación, que es la técnica más utilizada en el diseño de BD. En 1979, Codd intentó subsanar algunas de las deficiencias de su modelo relacional con una versión extendida denominada RM/T (1979) y posteriormente RM/V2 (1990).

Como respuesta a la creciente complejidad de las aplicaciones que requieren BD, ha surgido un nuevo modelo: el modelo de datos orientado a objetos. Esta evolución representa la tercera generación de los sistemas de BD.

Sistemas de Gestión de Bases de Datos (SGBD)

En un sistema de BD debe existir una capa intermedia entre los datos almacenados en la BD, las aplicaciones y los usuarios del mismo. Se trata del Sistema de Gestión de la BD (SGBD). Actúa de intermediario entre los usuarios y aplicaciones y los datos, proporcionando medios para describir, almacenar y manipular los datos y proporciona herramientas al administrador para gestionar el sistema, entre ellas las herramientas de desarrollo de aplicaciones, generador de informes, lenguajes específicos de acceso a los datos, como SQL (Structured Query Language) o QBE (Query By Example) en BD relacionales.

Un SGBD se puede definir como un conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o el administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la BD, manteniendo su integridad, confidencialidad y seguridad.

El objetivo primordial de un SGBD es proporcionar un entorno conveniente y eficiente para extraer, almacenar y manipular información de la BD. El SGBD gestiona de forma centralizada todas las peticiones de acceso a la BD, por lo que este paquete funciona como interfaz entre los usuarios y la BD. Además, el SGBD gestiona la estructura física de los datos y su almacenamiento. Por lo tanto, el SGBD libera al usuario de conocer exactamente la organización física de los datos y de crear algoritmos para almacenar, actualizar o consultar dicha información que está contenida en la BD.

Todos los SGBD no presentan la misma funcionalidad, depende de cada producto y del modelo de datos que implanten. Los sistemas más grandes son conjuntos de programas complejos y sofisticados. Los SGBD están en continua evolución, tratando de satisfacer los requerimientos de todo tipo de usuarios.

Veamos a continuación las principales funciones o características que debe proporcionar un SGBD.

Características de un SGBD

En general todos los SGBD presentan unas características comunes. Estas fueron ya definidas por Codd y posteriormente revisadas en función de las nuevas necesidades detectadas con la generalización del uso de las BD.

Idealmente, el SGBD debe poseer una serie de características indispensables para satisfacer a los usuarios, tales como:

- Mantener la independencia entre los programas y la estructura de la BD. Así se simplifica el mantenimiento de las aplicaciones que acceden a la BD. Aunque esta independencia nunca es absoluta, los SGBD, principalmente los relacionales, van respondiendo cada vez mejor a esta exigencia.
- Asegurar la coherencia de los datos. En lo posible, no debe existir redundancia de datos, los datos deben estar almacenados una sola vez en la BD.
- Permitir a los usuarios almacenar datos, acceder a ellos y actualizarlos. Además, el SGBD debe hacerlo de forma transparente al usuario, ocultando la estructura física interna de los datos y la forma de almacenarlos.
- Contener un catálogo accesible por los usuarios en el que se almacenen las descripciones de los datos de forma centralizada. Este catálogo se denomina diccionario de datos y permite identificar y eliminar las redundancias y las inconsistencias.
- Garantizar que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la BD. Si la transacción falla durante su realización, la BD quedará en un estado inconsistente. Algunos de los cambios se habrán hecho y otros no, por lo tanto, los cambios realizados deberán ser deshechos para devolver la BD a un estado consistente.
- Permitir que varios usuarios tengan acceso al mismo tiempo a los datos. Cuando dos o más usuarios acceden a la BD y al menos uno de ellos está actualizando datos, el SGBD deberá gestionar el acceso concurrente, impidiendo que haya datos corruptos o inconsistentes. Aquí el SGBD puede permitir la simultaneidad de accesos mediante el manejo eficiente de los bloqueos de la BD.
- Garantizar la recuperación de la BD en caso de que algún suceso la dañe. El fallo puede ser debido a una avería en algún dispositivo hardware o un error del software,

que hagan que el SGBD aborte, o puede ser debido a que el usuario detecte un error durante la transacción y la aborte antes de que finalice. En todos estos casos, el SGBD debe proporcionar un mecanismo capaz de recuperar la BD llevándola a un estado consistente.

- Garantizar la seguridad de la BD. Esto es, sólo los usuarios autorizados pueden acceder a la BD, permitiendo diferentes niveles de acceso. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.
- Garantizar la integridad de la BD. Esto requiere la validez y consistencia de los datos almacenados. Normalmente se expresa mediante restricciones, que son una serie de reglas que la BD no puede violar.
- Mantener la disponibilidad continua. La BD debe estar siempre disponible para su acceso. El SGBD debe proporcionar utilidades de administración, mantenimiento y gestión que puedan realizarse sin detener el funcionamiento de la BD.
- Proporcionar herramientas de administración de la BD. Estas herramientas permiten entre otras funcionalidades: importar y exportar datos, monitorizar el funcionamiento y obtener estadísticas de utilización de la BD, reorganizar índices y optimizar el espacio liberado para reutilizarlo.
- Integrarse con algún software gestor de comunicaciones. Muchos usuarios acceden a la BD desde terminales remotos, por lo que la comunicación con la máquina que alberga al SGBD se debe hacer a través de una red. Todas estas transmisiones de mensajes las maneja el gestor de comunicaciones de datos. Aunque este gestor no forma parte del SGBD, es necesario que el SGBD se pueda integrar con él.
- Garantizar la escalabilidad y elevada capacidad de proceso. El SGBD debe aprovechar todos los recursos de máquina disponibles en cada momento, aumentando su capacidad de proceso, conforme disponga de más recursos.
- Poseer un lenguaje de definición de datos que permita fácilmente la creación de nuevas BD, así como la modificación de su estructura.
- Poseer un lenguaje de manipulación de datos, que permita la inserción, eliminación, modificación y consulta de los datos de la base, de la forma más eficiente y conveniente posible.
- Permitir el almacenamiento de enormes cantidades de datos (miles de millones de caracteres), sin que el usuario perciba una degradación en cuanto al rendimiento global del sistema. Para ello el SGBD debe utilizar índices, partición de tablas, etc.

La forma en que las distintas BD comerciales y académicas abordan estas características difieren enormemente, no sólo por las técnicas utilizadas sino también por las aproximaciones o paradigmas con que se han desarrollado. En este tema nos centraremos exclusivamente en el tipo más extendido: las BD relacionales, ya que tienen un formalismo subyacente que las hace muy potentes. Además, fueron desarrolladas hace ya bastantes años, y han evolucionado lo suficiente como para suministrar poderosas herramientas que hacen fácil su gestión. De hecho, todas las características que hemos visto que debe poseer un SGBD, son suministradas a través de entornos e interfaces amigables y comprensibles que permiten un rápido aprendizaje de todas las funciones propias de una BD.

En contraposición, otro tipo de BD, como las orientadas a objetos, requieren que casi todas las funciones de creación de BD, manipulación, etc, se efectúen a través de programas, lo cual requiere un profundo conocimiento de las técnicas de programación.

Por otro lado, sistemas igual de evolucionados, como el jerárquico o en red, han caído en desuso, y su aprendizaje supone un esfuerzo que aporta más bien poco al diseñador que debe enfrentarse de inmediato ante un mundo de datos básicamente relacional.

Niveles de abstracción: Interno, Conceptual y Externo

Se puede observar en los Sistemas de Información la existencia de dos niveles distintos:

- Un nivel lógico o externo, que es la vista que tiene el usuario del sistema.

- Un nivel físico o interno, que es la forma en la que los datos están almacenados.

En la BD aparece un nuevo nivel de abstracción llamado: nivel conceptual. Este nivel intermedio pretende una representación global de los datos que se interponga entre el nivel lógico y el físico, y que sea independiente tanto del equipo, como de cada usuario en particular.

Una de las características más importantes de los SGBD es la independencia entre programas y datos.

Según ANSI (American National Standard Institute), “la independencia de los datos es la capacidad de un sistema para permitir que las referencia a los datos almacenados, especialmente en los programas y en sus descriptores de los datos, están aislados de los cambios y de los diferentes usos en el entorno de los datos, como pueden ser la forma de almacenar dichos datos, el modo de compartirlos con otros programas y como se reorganizan para mejorar el rendimiento del sistema de BD”.

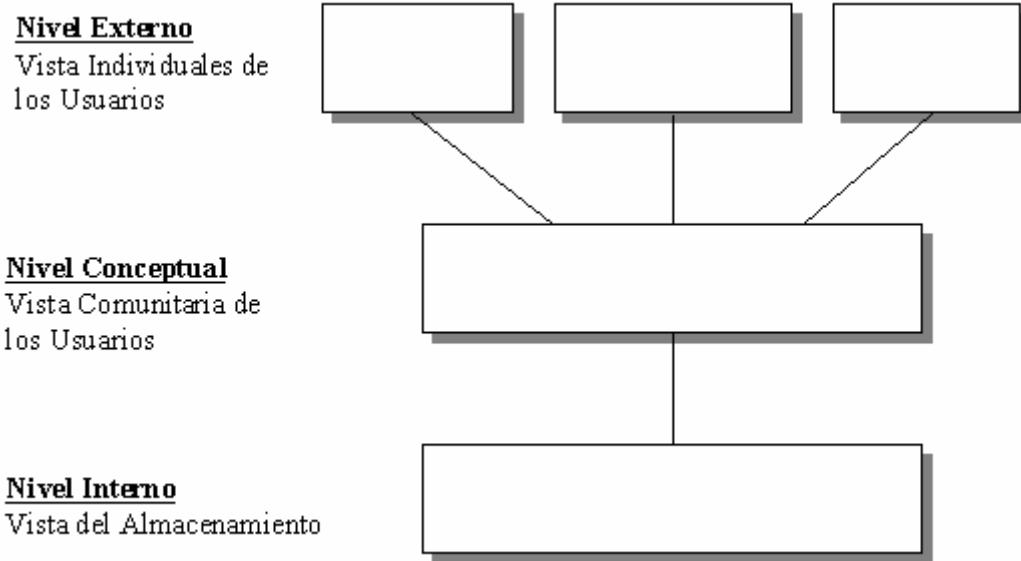
Para asegurar esta independencia entre los datos y las aplicaciones es necesario separar la representación física y lógica de los datos, distinción que fue reconocida oficialmente en 1978, cuando el comité ANSI/X3/SPARC propuso una arquitectura de 3 niveles: nivel interno, nivel conceptual y nivel externo:

- **Nivel interno** : Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la estructura física de la BD: dispositivos de almacenamiento físico, estrategias de acceso, índices, etc. Ningún usuario necesita conocer este nivel, su organización y conocimiento está reservado a los administradores de la BD.
- **Nivel conceptual** : El siguiente nivel de abstracción, describe qué datos son almacenados realmente en la BD y las relaciones que existen entre los mismos, describe la BD completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de BD, quienes deben decidir qué información se va a guardar en la BD. En el nivel conceptual la BD aparece como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad los archivos conceptuales no existen físicamente. La transformación de registros conceptuales a registros físicos para el almacenamiento se lleva a cabo por el sistema y es transparente al usuario. Consta de las siguientes definiciones:
 - Definición de los datos: Se describen las características y tipos de campo de todos los elementos direccionables en la BD.
 - Relaciones entre datos: Se definen las relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.
- **Nivel externo** : Nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe sólo una parte de la BD al usuario acreditado para verla. El sistema puede proporcionar muchas visiones para la misma BD.

La arquitectura de tres niveles es útil para explicar el concepto de independencia de datos que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin modificar el esquema del nivel superior. Se pueden definir dos tipos de independencia de datos:

- La independencia lógica permite modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación.
- La independencia física permite modificar el esquema interno sin alterar el esquema conceptual (o los externos). Por ejemplo, permite cambiar el disco en que se almacenan parte de los ficheros físicos con el fin de mejorar el rendimiento de las operaciones de consulta o aumentar la capacidad de almacenamiento de datos. La independencia física es más fácil de conseguir que la independencia lógica.

La siguiente figura, muestra los tres niveles de abstracción mencionados.



Lenguajes de los SGBD

Para proporcionar a los usuarios las diferentes facilidades, los SGBD deben ofrecer lenguajes especializados e interfaces apropiadas para cada tipo de usuario: administradores de la BD, diseñadores, programadores de aplicaciones y usuarios finales.

La interacción del usuario con la BD 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. 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 que interactúan con los SGBD, se pueden clasificar en dos grandes grupos:

- Unos orientados hacia la función: Son los lenguajes de definición, manipulación y control.
- Otros orientados a los diferentes tipos de usuarios o de procesos.

Dentro del segundo grupo se encuentran los lenguajes de programación a los que están habituados los usuarios informáticos: programadores, analistas, etc. A este tipo de lenguajes se les conoce como "lenguaje anfitrión". A las sentencias de manipulación de los lenguajes de las BD que son utilizadas en estos lenguajes se les conoce como "lenguaje huésped".

Los SGBD pueden admitir varios lenguajes de tipo anfitrión para manipulación de datos, como: Cobol, Ensamblador, Fortran, PL/I, Basic, Pascal, C, etc.

Ahora nos vamos a centrar en los lenguajes del primer grupo, orientados hacia la función.

Lenguaje de Definición de Datos (LDD) o Data Definition Language (DDL)

El lenguaje de definición de datos está orientado a la definición, descripción y mantenimiento de la estructura de la BD. Permite al administrador definir los datos con facilidad y precisión, especificando sus distintas estructuras. Debe tener facilidad para describir la estructura del esquema conceptual, hacer las especificaciones relativas al esquema físico, y declarar las estructuras del esquema externo, requeridas por las aplicaciones.

Para el caso concreto de los SGBD relacionales, se utiliza como estándar el SQL, para crear las BD a partir del esquema relacional. Mediante el DDL del SQL se crean tablas,

columnas con los dominios correspondientes, índices, claves, las restricciones de integridad, etc.

El SGBD posee un compilador de DDL cuya función consiste en procesar las sentencias del lenguaje para identificar las descripciones de los distintos elementos de los esquemas y almacenarlas generalmente en una BD especial que contiene los "metadatos".

Esta BD especial, es comúnmente llamada diccionario de datos o catálogo del SGBD. Dicho catálogo es el que se consulta, para obtener la estructura de la BD, toda vez que se quiere leer, modificar o eliminar los datos de la BD.

El DDL permite especificar:

- Elementos de datos
- Estructura de datos
- Relaciones entre datos
- Reglas de integridad
- Vistas lógicas
- Espacio reservado para la BD
- Formato de representación (binario, decimal, ...)
- Modo de acceso (punteros, índices, ...)

Lenguaje de Manipulación de Datos (LMD) o Data Manipulation Language (DML)

El lenguaje de consulta y manipulación de datos sirve para obtener, insertar, eliminar y modificar los datos de la BD.

Al igual que el programador necesita el LMD como lenguaje huésped dentro de un lenguaje anfitrión que maneja, el usuario no informático necesita de un instrumento para comunicarse con la BD. Este instrumento suele ser un LMD autocontenido, que da facilidades a los usuarios con pocos conocimientos de programación a acceder y manipular los datos en modo interactivo.

El lenguaje de manipulación de datos SQL, puede actuar al mismo tiempo como huésped y como autocontenido, cumpliendo la propiedad dual (Codd 1990).

En una primera clasificación de los LMD, hay dos tipos de lenguajes según su definición:

- *DML procedural* . El programador especifica qué datos se necesitan y cómo obtenerlos. Se deben especificar todas las operaciones de acceso a datos llamando a los procedimientos necesarios para obtener la información requerida. Estos lenguajes acceden a un registro, lo procesan y basándose en los resultados obtenidos, acceden a otro registro, que también deben procesar. Así se va accediendo a registros y se van procesando hasta que se obtienen los datos deseados. Las sentencias de un DML procedural deben estar embebidas en un lenguaje de alto nivel. Como ya hemos comentado este es el lenguaje conocido como lenguaje anfitrión.
- *DML no procedural* . El usuario o programador especifica qué datos quiere obtener sin decir cómo se debe acceder a ellos. El SGBD traduce las sentencias del DML en uno o varios procedimientos que manipulan los conjuntos de registros necesarios. Esto libera al usuario de tener que conocer cuál es la estructura física de los datos y qué algoritmos se deben utilizar para acceder a ellos. A los DML no procedurales también se les denomina lenguajes declarativos.

El lenguaje DML no procedural más conocido es el SQL. Los lenguajes no procedurales son más fáciles de utilizar y conocer que los procedurales porque el SGBD oculta al usuario los detalles sobre cómo se ha realizado la operación solicitada.

En una segunda clasificación de los LMD, hay dos tipos de lenguajes según como recuperan la información:

- *Navegacionales* : Recuperan o actualizan los datos registro a registro, debiendo el programador indicar el camino que se ha de recorrer, a través de la estructura definida, hasta llegar al registro buscado. Se utilizan estos lenguajes en BD en red y jerárquicas.
- *No navegacionales* : Actúan sobre un conjunto de registros. Una única sentencia puede dar lugar a recuperar o actualizar todos los registros que cumplan una determinada condición. El SQL es de este tipo.

En el caso del SQL, asociado al LMD se suele encontrar un módulo optimizador que se ocupa de analizar la petición contra la BD y decidir el mejor camino de acceso con el fin de acelerar la ejecución. Para la toma de decisiones, el optimizador necesita de la información contenida en el catálogo o diccionario del SGBD.

La manipulación de datos comprende las siguientes operaciones:

- Recuperación de información
- Inserción de nueva información
- Eliminación de información existente
- Modificación de información almacenada

Lenguaje de Control de Datos (LCD) o Data Control Language (DCL)

El lenguaje de Control de Datos sirve para trabajar en un entorno multiusuario, donde es muy importante la protección y la seguridad de los datos y la compartición de datos por parte de usuarios.

Se encarga principalmente de tres actividades sobre la BD:

- Control de permisos de acceso
- Control de concurrencia
- Control de transacciones

Estructura de un SGBD

Los SGBD son paquetes de software muy complejos que deben proporcionar los servicios comentados anteriormente. Los elementos que componen un SGBD varía mucho unos de otros. El SO proporciona servicios básicos al SGBD, que es construido sobre él.

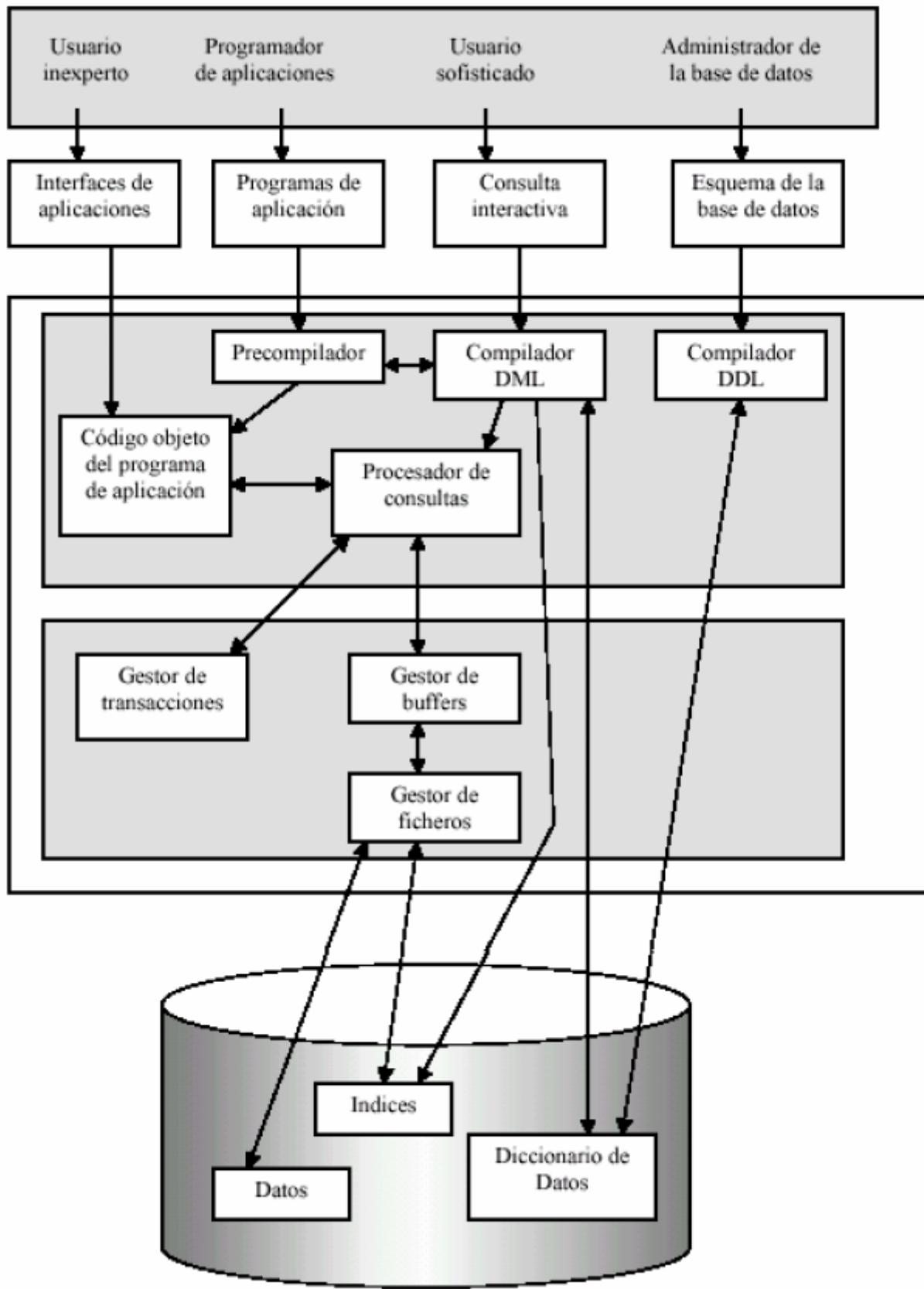
Los principales módulos del SGBD son:

- El compilador del DDL. Chequea la sintaxis de las sentencias del DDL y actualiza las tablas del diccionario de datos o catálogo que contienen los metadatos.
- El precompilador del DML. Convierte las sentencias del DML embebidas en el lenguaje anfitrión, en sentencias listas para su procesamiento por parte del compilador de lenguaje anfitrión y además extrae dichas sentencia DML para que puedan ser procesadas de forma independiente por el compilador del DML.
- El compilador del DML. Chequea la sintaxis de las sentencias del DML y se las pasa al procesador de consultas.
- El procesador de consultas. Realiza la transformación de las consultas en un conjunto de instrucciones de bajo nivel que se dirigen al gestor de la BD.
- El gestor de la BD. Es el interfaz con los programas de aplicación y las consultas de los usuarios. El gestor de la BD acepta consultas y examina los esquemas externo y conceptual para determinar qué registros se requieren para satisfacer la petición. Entonces el gestor de la BD realiza una llamada al gestor de ficheros para ejecutar la petición.

Los principales componentes del gestor de la BD son los siguientes:

- El gestor de transacciones. Realiza el procesamiento de las transacciones.
- El gestor de buffers. Transfiere los datos entre memoria principal y los dispositivos de almacenamiento secundario.
- El gestor de ficheros. Gestiona los ficheros en disco en donde se almacena la BD. Este gestor establece y mantiene la lista de estructuras e índices definidos en el esquema interno. Para acceder a los datos pasa la petición a los métodos de acceso al SO que se encargan de leer o escribir en los ficheros físicos que almacenan la información de la BD.

En la figura se ilustra cómo se relacionan entre sí todos los elementos.



La primera fila de esta figura son los distintos tipos de usuarios que pueden acceder al SGBD (usuarios inexpertos, programadores de aplicaciones, usuarios sofisticados y administradores).

La segunda fila son los distintos métodos de acceso a la información que utilizan los usuarios (interfaces de aplicaciones, programas de aplicación, consultas interactivas o el esquema de la BD).

Los usuarios inexpertos y los programadores de aplicaciones utilizan aplicaciones informáticas para acceder a la BD y los usuarios sofisticados y administradores acceden directamente a ella.

El tercer bloque es el SGBD y se subdivide en dos partes:

- La primera recibe las peticiones de los cuatro tipos de usuarios y las dirige al gestor de la BD o directamente al diccionario de datos. Independientemente de si las peticiones de manipulación de datos llegan de programas de aplicación o de una consulta directa a la BD por un usuario sofisticado, finalmente es el procesador de consultas el que las redirige al gestor de la BD.
- La segunda es el gestor de la BD, que consta de los gestores de transacciones, de buffer y de ficheros. El gestor de buffer sólo se comunica con el gestor de ficheros y es este último el que accede a la estructura física de la BD.

Por último tenemos la BD formada por los datos, sus índices y el diccionario de datos.

Esta estructura es general y algunos de los SGBD actuales incorporan otras funciones y módulos para dotar al SGBD de nuevas facilidades o incrementar su eficiencia. Entre las funciones adicionales deseables en un SGBD se encuentran las siguientes:

- Utilidades de carga de datos (importación y exportación de datos) con posibilidad de conversión de formatos de ficheros.
- Copia de seguridad (backup).
- Estadísticas de utilización.
- Reorganización de ficheros (mejora del rendimiento).
- Control del rendimiento.
- Registro de transacciones.
- Gestor de bloqueos.
- Distribución de procesos entre máquinas.

SGBD Relacionales (SGBD-R)

A continuación vamos a estudiar los Sistemas de Gestión de BD Relacionales (SGBD-R) y para ello veremos antes a nivel conceptual el modelo relacional. Este modelo es fundamental porque dio origen a los primeros sistemas comerciales de SGBD-R, que son los que hoy en día dominan el mercado de BD.

El modelo relacional

El modelo de datos relacional fue presentado por Codd en 1970 y se basa en la representación del universo del discurso mediante el álgebra relacional. Codd, que era un experto matemático, utilizó una terminología perteneciente a las matemáticas, en concreto de la teoría de conjuntos y de la lógica de predicados.

Las características principales del modelo son las siguientes:

- Está basado en un modelo matemático con reglas y algoritmos algebraicos establecidos, lo cual permite el desarrollo de lenguajes de acceso y manipulación potentes y de fiabilidad demostrable.
- Estructura los dato en forma de relaciones que se modelan mediante tablas bidimensionales. Estas tablas representan tanto las entidades del universo del discurso como las relaciones entre las mismas.
- Permite incorporar aspectos semánticos del universo del discurso mediante el establecimiento de reglas de integridad. Estas reglas permiten trasladar al esquema conceptual restricciones o comportamientos de los datos presentes en el universo del discurso que no se podrían modelar exclusivamente con tablas.

Características de los SGBD-R

Los SGBD construidos sobre el modelo relacional se caracterizan, por tanto, por las estructuras de datos, los operadores asociados y los aspectos semánticos. A continuación vamos a ver estos tres conceptos.

Estructuras de datos: Relaciones y Claves

En la estructura básica del modelo relacional se distinguen los siguientes elementos:

- **Relación** : Es un subconjunto de un producto cartesiano entre conjuntos formados por atributos. Por ejemplo, una relación R, definida sobre los atributos A_1, A_2, \dots, A_N , sería un subconjunto formado por m elementos del producto cartesiano A_1, A_2, \dots, A_N . En el modelo relacional se representa mediante una tabla con m filas y n columnas. Como las tablas son esencialmente relaciones, se utilizarán los términos matemáticos relación y tupla, en lugar de los términos tabla y fila.
- **Atributos** : Son las columnas de la tabla. Corresponden a las propiedades de las entidades presentes en el universo del discurso que nos interesa almacenar en la BD. Cada uno de estos atributos puede tomar valores dentro de un rango determinado, que se llama dominio. Varios atributos pueden compartir un único dominio.
- **Dominio** : Rango de valores aceptable para un atributo dado. Este rango depende exclusivamente del atributo y va a condicionar los valores posibles dentro de cada celda de la tabla. Los valores que forman el dominio deben ser homogéneos, es decir, del mismo tipo y atómicos, o sea, indivisibles. Un valor de dominio que es miembro de todos los dominios posibles, es el valor *nulo*, que indica que el valor es desconocido o no existe.
- **Tuplas** : Es el nombre que recibe cada una de las filas de la tabla. Corresponden a cada una de las ocurrencias de la relación que representa la tabla o, lo que es lo mismo, a cada uno de los elementos que forman el conjunto R de la relación. El orden en el que aparecen las tuplas es irrelevante, dado que la relación es un conjunto de tuplas.
- **Cardinalidad de la relación** : es el número m de tuplas de la relación.
- **Grado de la relación** : es el número n de atributos que intervienen en la relación.

Una vez visto qué es una tabla o relación, vamos a enumerar sus propiedades principales:

- Todas las filas de una tabla están compuestas por el mismo número y tipo de atributos que, además, aparecen siempre en el mismo orden.
- No puede haber filas repetidas. Es decir, todas las filas de la tabla deben diferenciarse entre sí al menos en el valor de un atributo.
- El orden en que aparecen las filas dentro de la tabla no es relevante.
- En cada celda de la tabla sólo puede aparecer un valor. Además este valor debe estar dentro del dominio de la columna correspondiente.

Una tabla no puede contener dos filas iguales. Esto obliga, necesariamente, a que haya uno o varios atributos que se puedan utilizar para distinguir unas tuplas de otras. Cualquier atributo o conjunto mínimo de ellos que sirva para este propósito se denomina **clave candidata**. Es decir, una clave candidata permite identificar de forma única una fila de una tabla.

Por conjunto mínimo se entiende aquel conjunto de atributos tal que si se elimina uno de ellos el conjunto resultante deja de ser clave candidata. Es posible que la única clave candidata de una relación esté formada por todos los atributos de la misma.

A la clave candidata que el usuario escoge para identificar las tuplas de una relación se la denomina **clave primaria**. La elección de esta claves es decisión del usuario, aunque se suele utilizar la más corta por razones de eficiencia. Una propiedad fundamental de la

clave primaria consiste en que, bajo ninguna circunstancia, puede adoptar el valor nulo, ya que si así lo hiciera perdería su capacidad para identificar las tuplas de la relación.

El resto de claves candidatas que no han sido elegidas como clave primaria reciben el nombre de **claves alternativas o secundarias**.

Una relación R1 puede incluir entre sus atributos la clave primaria de otra relación R2. Esta clave es una **clave ajena** de R1 que hace referencia a R2. La relación R1 también se denomina relación de *referencia* de la dependencia de clave ajena, y R2 se denomina la relación *referenciada* de la clave ajena.

Operadores asociados

Los operadores asociados al modelo de datos relacional forman el álgebra relacional. Se puede demostrar matemáticamente que éste álgebra es completa, o sea, que por medio de ella se puede realizar cualquier acceso a la BD.

Los operandos con los que trabaja el álgebra son relaciones del modelo relacional y los operadores básicos son:

- **Unión** . La unión de dos relaciones R y S ($R \cup S$) es el conjunto formado por todas las tuplas de R más todas las tuplas de S. Este operador sólo se puede aplicar a relaciones del mismo grado y con los mismos atributos.
- **Diferencia** . La diferencia de dos relaciones R y S ($R - S$) es el conjunto formado por todas las tuplas de R que no están en S. Este operador sólo se puede aplicar a relaciones del mismo grado y con los mismos atributos.
- **Producto cartesiano** . El producto cartesiano de dos relaciones R y S, de grados m y n respectivamente, se denota $R \times S$ y es el conjunto formado por todas las posibles tuplas de $m + n$ elementos en las que los m primeros elementos son de R y los n restantes pertenecen a S.
- **Proyección** . Si x es un subconjunto de atributos de la relación R, entonces la proyección $\pi_x(R)$ es la relación formada por las columnas de R correspondientes a los atributos x .
- **Selección** . Si F es una fórmula compuesta por operadores lógicos, aritméticos y de comparación y sus operandos son los valores de los atributos de una relación R, entonces la selección $\sigma_F(R)$ es el conjunto de tuplas de la relación R que hacen verdadera la condición establecida por la fórmula F.

A partir de estos cinco operadores básicos, es posible definir otros derivados tales como la intersección, el cociente y la unión natural.

Aspectos semánticos

Los aspectos semánticos son todos aquellos aspectos del universo del discurso que no pueden modelarse mediante la definición de relaciones, sino que necesitan de un nivel superior de descripción.

Este nivel superior de descripción del modelo se traduce, en la práctica, en el establecimiento de restricciones adicionales a las propias del modelo relacional que ya se han mencionado (tuplas no repetidas, orden de las columnas constante, etc.) y que tienen como fin mantener la integridad y validez de los datos almacenados así como aumentar el grado de información que el esquema lógico de datos proporciona.

A continuación describiremos las dos principales restricciones que se manejan en el modelo relacional:

- **Restricciones de usuario** . Son restricciones a los valores del dominio de los atributos. La capacidad de definir estas restricciones de usuario, así como su

potencia y los elementos sobre los que se pueden aplicar (dominios, atributos, tuplas, tablas, etc.) dependen del gestor de BD.

- **Integridad referencial** . Otro aspecto que se puede incluir en el modelo relacional es la denominada integridad referencial, que se ocupa del mantenimiento de referencias entre las propias relaciones o tablas.

Formalmente, se define integridad referencial de la siguiente manera “Sean dos relaciones R1 (relación que referencia) y R2 (relación referenciada), no necesariamente distintas entre sí. Si ocurre que la relación R1 tiene un atributo o conjunto de atributos que es clave primaria de R2, entonces cualquier valor de dicho atributo o conjunto de atributos en R1 debe concordar con un valor de la clave primaria de R2 o bien ser nulo”.

El mantenimiento de la integridad referencial supone la realización de alguna acción cuando se borra o modifica una tupla en la tabla referenciada R2. Esta acción debe ser alguna de las siguientes:

- Impedir la operación de borrado o modificación. Así se asegura que una vez establecida no se puede romper la relación entre dos tuplas de ambas tablas.
- Transmitir en cascada la modificación. O sea, borrar o modificar en consecuencia las tuplas que hacen referencia a la que se acaba de borrar o modificar.
- Poner a nulo. Esto quiere decir que se asigna el valor nulo al atributo que ejerce de clave referencial para mantener la integridad.
- Poner valor por omisión o lamar una procedimiento de usuario. En este caso cuando se altera el valor del atributo referenciado, se pone como valor de la clave referencial un valor por omisión o se ejecuta un procedimiento por el usuario que establezca algún valor que sirva para mantener la integridad referencial.

Lenguajes de interrogación de Bases de Datos

Un lenguaje de interrogación o consulta es un lenguaje en el que un usuario solicita información de la BD. Estos lenguajes suelen ser de un nivel superior que el de los lenguajes de programación habituales. Los lenguajes de consulta pueden clasificarse, tal como vimos al estudiar los LMD, en dos grupos:

- **Lenguajes procedurales o procedimentales** : El usuario instruye al sistema para que lleve a cabo una serie de operaciones en la BD para calcular el resultado deseado.
- **Lenguajes no procedurales o no procedimentales** : El usuario describe la información deseada sin dar un procedimiento concreto para obtener esa información.

El álgebra relacional

El álgebra relacional forma la base del lenguaje de consulta SQL.

El álgebra relacional es un lenguaje de consulta procedimental. Consta de un conjunto de operaciones que toman como entrada una o dos relaciones y producen como resultado una nueva relación.

Las operaciones asociadas a este modelo de datos forman el álgebra relacional. Se puede demostrar matemáticamente que ésta álgebra es completa, o sea, que por medio de ella se puede realizar cualquier acceso a la BD.

Las operaciones fundamentales del álgebra relacional son: *selección, proyección, unión, diferencia de conjuntos, producto cartesiano y renombramiento* . Además de estas operaciones fundamentales hay otras operaciones que se definen a partir de las fundamentales, tales como: *intersección de conjuntos, unión natural y asignación* .

Operaciones Fundamentales

Se dividen estas operaciones en dos tipos:

- **Unarias** : Porque operan con una sola relación o tabla. Son: selección, proyección y renombramiento.
- **Binarias** : Porque operan con dos relaciones. Son: unión, diferencia de conjuntos y producto cartesiano.

La operación selección

La operación selección selecciona tuplas que satisfacen un predicado dado. Se utiliza la letra griega sigma minúscula (σ) para denotar la selección. El predicado aparece como subíndice de σ . La relación de entrada es el argumento que aparece entre paréntesis a continuación de σ .

Por lo tanto, la definición formal dice: Si P es un predicado compuesto por operadores lógicos, aritméticos y de comparación y sus operandos son los valores de los atributos de una relación R, entonces la selección $\sigma_P(R)$ es el conjunto de tuplas de la relación R que hacen verdadera la condición establecida por el predicado P.

En general, se permite las comparaciones que utilizan los signos: $=$, \neq , $<$, \leq , $>$ o \geq , en el predicado de selección. Además se pueden combinar varios predicados en uno mayor utilizando las conectivas y(\wedge) y o(\vee). El predicado de selección puede incluir comparación entre dos atributos.

$\sigma_{campo="valor"} \text{ (Tabla)}$

Ejemplo:

La operación proyección

La operación proyección es una operación unaria que dada una relación de entrada, devuelve todos los atributos de dicha relación que aparecen en los argumentos de dicha operación. Dado que las relaciones son conjuntos, se eliminan todas las filas duplicadas.

La proyección se denota por la letra griega pi (π). Se crea una lista de atributos que se desea que aparezcan en el resultado, como subíndice de π . La relación de entrada es el argumento que aparece entre paréntesis a continuación de π .

Por lo tanto, la definición formal dice: Si x es un subconjunto de atributos de la relación R, entonces la proyección $\pi_x(R)$ es la relación formada por las columnas de R correspondientes a los atributos x.

$\pi_{campo1,campo2, \dots} \text{ (Tabla)}$

Por ejemplo:

Composición de operaciones relacionales

Es importante el hecho de que el resultado de una operación relacional sea también una relación.

En general, dado que el resultado de una operación de álgebra relacional es del mismo tipo (relación) que los datos de entrada, las operaciones del álgebra relacional pueden componerse para formar una expresión del álgebra relacional. La composición de operaciones del álgebra relacional para formar expresiones del álgebra relacional es igual que la composición de operaciones aritméticas (como $+$, $-$, $*$ y \div) para formar expresiones aritméticas.

Por ejemplo: $\pi_{campo1,campo2,\dots}(\sigma_{campo="valor"}(Tabla))$

La operación unión

Se debe asegurar que las uniones se realicen entre relaciones compatibles, es decir, que deben cumplir las dos condiciones siguientes:

- Las dos relaciones deben ser de la misma *aridad*, es decir, deben tener el mismo número de atributos.
- Los dominios de los atributos, deben ser iguales.

Por lo tanto, la definición formal dice: La unión de dos relaciones R y S (R U S) es el conjunto formado por todas las tuplas de R más todas las tuplas de S. Este operador sólo se puede aplicar a relaciones del mismo grado y con los mismos atributos.

Por ejemplo: $\pi_{campo1,campo2,\dots}(Tabla1) \cup \pi_{campo1,campo2,\dots}(Tabla2)$

La operación diferencia de conjuntos

La operación diferencia de conjuntos, denotada por -, permite buscar las tuplas que estén en una relación pero no en otra. La definición formal dice: La diferencia de dos relaciones R y S (R - S) es el conjunto formado por todas las tuplas de R que no están en S.

Este operador, al igual que el operador unión, solo puede realizarse entre relaciones compatibles. Por lo tanto el operador diferencia sólo se puede aplicar a relaciones del mismo grado y con los mismos atributos.

Por ejemplo: $\pi_{campo1,campo2,\dots}(Tabla1) - \pi_{campo1,campo2,\dots}(Tabla2)$

La operación producto cartesiano

La operación producto cartesiano se denota por un aspa (x), permite combinar información de cualesquiera dos relaciones. Hay que considerar dos posibles problemas:

- Si las dos relaciones de entrada tienen un atributo con el mismo nombre, se adjunta a dicho atributo el nombre de la relación, para así distinguir uno de otro.
- Si el nombre de las dos relaciones de entrada es el mismo (producto cartesiano de una relación consigo misma) o si se utiliza el resultado de una expresión del álgebra relacional en un producto cartesiano, se debe dar un nuevo nombre a una de las relaciones o a la expresión del álgebra relacional utilizando una operación de renombramiento que veremos en el apartado siguiente.

La definición formal dice: El producto cartesiano de dos relaciones R y S, de grados m y n respectivamente, se denota $R \times S$ y es el conjunto formado por todas las posibles tuplas de $m + n$ atributos en las que los m primeros atributos son de R y los n restantes pertenecen a S.

La operación renombramiento

A diferencia de las relaciones de BD, los resultados de las expresiones del álgebra relacional no tienen un nombre que se pueda utilizar para referirse a ellas. Resulta, por lo tanto, útil ponerles nombre. La operación renombramiento denotado por la letra griego rho (ρ), permite realizar esta tarea.

La definición formal dice: Data una expresión E del álgebra relacional, la expresión $\rho_{\text{valor}}(E)$, devuelve el resultado de la expresión E con nombre x.

Las relaciones por sí mismas también se consideran expresiones triviales del álgebra relacional. Por lo tanto, también se puede aplicar la operación renombramiento a una relación dada, para obtener la misma relación con un nuevo nombre.

$$\rho_{\text{valor}}(\pi_{\text{campo1}, \text{campo2}, \dots}(\sigma_{\text{campo}=\text{"valor"}}(\text{Tabla})))$$

Por ejemplo:

Otras operaciones derivadas

Las operaciones fundamentales del álgebra relacional son suficientes para expresar cualquier consulta del álgebra relacional. Sin embargo, si uno se limita únicamente a las operaciones fundamentales, algunas consultas habituales resultan ser algo más complejas de expresar. Por este motivo, se definen otras operaciones que no añaden potencia al álgebra, pero que simplifican las consultas habituales.

A partir de las operaciones básicas, es posible definir otras operaciones derivadas tales como la intersección, la unión natural y la asignación.

La operación intersección de conjuntos

La operación intersección de conjuntos denotada por \cap , permite buscar las tuplas que estén al tiempo en las dos relaciones sobre las que actúa.

Se observa que esta operación no es fundamental y no añade potencia al álgebra relacional, ya que puede ser expresada en función de la operación de conjuntos, de la manera siguiente: $R \cap S = R - (R - S)$.

$$\pi_{\text{campo1}, \text{campo2}, \dots}(\text{Tabla1}) \cap \pi_{\text{campo1}, \text{campo2}, \dots}(\text{Tabla2})$$

Por ejemplo:

La operación unión natural

Suele resultar deseable simplificar ciertas consultas que exigen producto cartesiano. Generalmente, las consultas que implican producto cartesiano incluyen un operador selección sobre el resultado del producto cartesiano.

La unión natural es una operación binaria que permite combinar ciertas selecciones y un producto cartesiano en una sola operación. Se denota por el símbolo de la “reunión”. La operación unión natural forma un producto cartesiano de sus dos argumentos, realiza una selección forzando la igualdad de los atributos que aparecen en ambas relaciones y, finalmente elimina los atributos duplicados.

La operación asignación

En ocasiones resulta conveniente escribir una expresión de álgebra relacional por partes utilizando la asignación a una variable de relación temporal.

La operación asignación, denotada por \leftarrow , actúa de manera parecida a la asignación de los lenguajes de programación.

Con la operación asignación se puede escribir las consultas como programas secuenciales consistentes en una serie de asignaciones seguida de una expresión cuyo valor se muestra como resultado de la consulta.

Por ejemplo: $\text{temp1} \leftarrow \pi_{\text{campo1}, \text{campo2}, \dots} (\sigma_{\text{campo}=\text{"valor}} (\text{Tabla}))$

El cálculo relacional de tuplas

Cuando escribimos una expresión de álgebra relacional proporcionamos una serie de procedimientos que generan la respuesta a la consulta. El cálculo relacional de tuplas, en cambio, es un lenguaje de consulta no procedimental. Describe la información deseada sin dar un procedimiento específico para obtenerla.

Las consultas se expresan en el cálculo relacional de tuplas como: $\{ t \mid P(t) \}$, es decir, son el conjunto de todas las tuplas tales que el predicado P es cierto para t . Se utiliza la notación $t[A]$ para denotar el valor de la tupla t en el atributo A y $t \in R$ para denotar que la tupla t está en la relación R .

Para poder hacer una definición formal del cálculo relacional de tuplas, debemos conocer los tres conceptos siguientes:

- Constructor “existe” de la lógica matemática. La notación $\exists t \in R(Q(t))$ significa: existe una tupla t en la relación R que el predicado $Q(t)$ es verdadero.
- Implicación denotada por \Rightarrow , es decir P implica Q , se escribe: $P \Rightarrow Q$
- Constructor “para todo” de la lógica matemática. La notación $\forall t \in R(Q(t))$ significa: Q es verdadera para todas las tuplas t de la relación R .

Ahora podemos dar una definición formal del cálculo relacional de tuplas.

Como ya hemos dicho, las expresiones del cálculo relacional de tupla son de la forma: $\{ t \mid P(t) \}$ donde P es un predicado o una fórmula. En una fórmula pueden aparecer varias variables tupla. Se dice que una variable tupla es una variable libre a menos que esté cuantificada mediante \exists o \forall .

Las fórmulas del cálculo relacional de tuplas se construyen con átomos. Los átomos tienen una de las formas siguientes:

- $s \in R$, donde s es una variable tupla y R es una relación.
- $s[x] \theta u[y]$, donde s y u son variables tuplas, x es un atributo en el que está definida s , y es un atributo que está definida en u y θ es un operador de comparación ($<$, \leq , $>$, \geq , \neq , $=$). Es necesario que los atributos x e y tengan dominios cuyos miembros puedan compararse mediante θ .
- $s[x] \theta c$, donde s es una variable tupla, x es un atributo en el que está definida s , θ es un operador de comparación y c es una constante en el dominio de atributo x .

Las fórmulas se construyen a partir de los átomos utilizando las reglas siguientes:

- Si P_1 es una fórmula, también lo son $\neg P_1$ y (P_1)
- Si P_1 y P_2 son fórmulas, también lo son $P_1 \vee P_2$, $P_1 \wedge P_2$ y $P_1 \Rightarrow P_2$
- Si $P_1(s)$ es una fórmula que contiene una variable tupla libre s , y R es una relación: $\exists s \in R(P_1(s))$ y $\forall s \in R(P_1(s))$ también son fórmulas.

El cálculo relacional de dominios

Hay una segunda forma de cálculo relacional denominada cálculo relacional de dominios. Esta forma utiliza variables de dominio que toman sus valores del dominio de un atributo, en vez de tomarlos de una tupla completa. El cálculo relacional de dominios, sin embargo, se halla estrechamente relacionado con el cálculo relacional de tuplas.

Las expresiones de cálculo relacional de dominios son de la forma:

$\{<x_1, x_2, \dots, x_n> \mid P(x_1, x_2, \dots, x_n)\}$, donde x_1, x_2, \dots, x_n representan las variables de dominio, P representa una fórmula compuesta de átomos, como era el caso en el cálculo relacional de tuplas.

Los átomos del cálculo relacional de dominios tienen una de las formas siguientes:

- $<x_1, x_2, \dots, x_n> \in R$, donde R es una relación con n atributos y x_1, x_2, \dots, x_n son variables de dominio o constantes de dominio.
- $x \theta y$, donde x e y son variables de dominio y θ es un operador de comparación ($<$, \leq , \geq , $>$, $=$). Se exige que los atributos x e y tengan dominios que puedan compararse mediante θ .
- $x \theta c$, donde x es una variable de dominio, θ es un operador de comparación y c es una constante de dominio del atributo para el que x es una variable de dominio.

Las fórmulas se construyen a partir de los átomos utilizando las reglas siguientes:

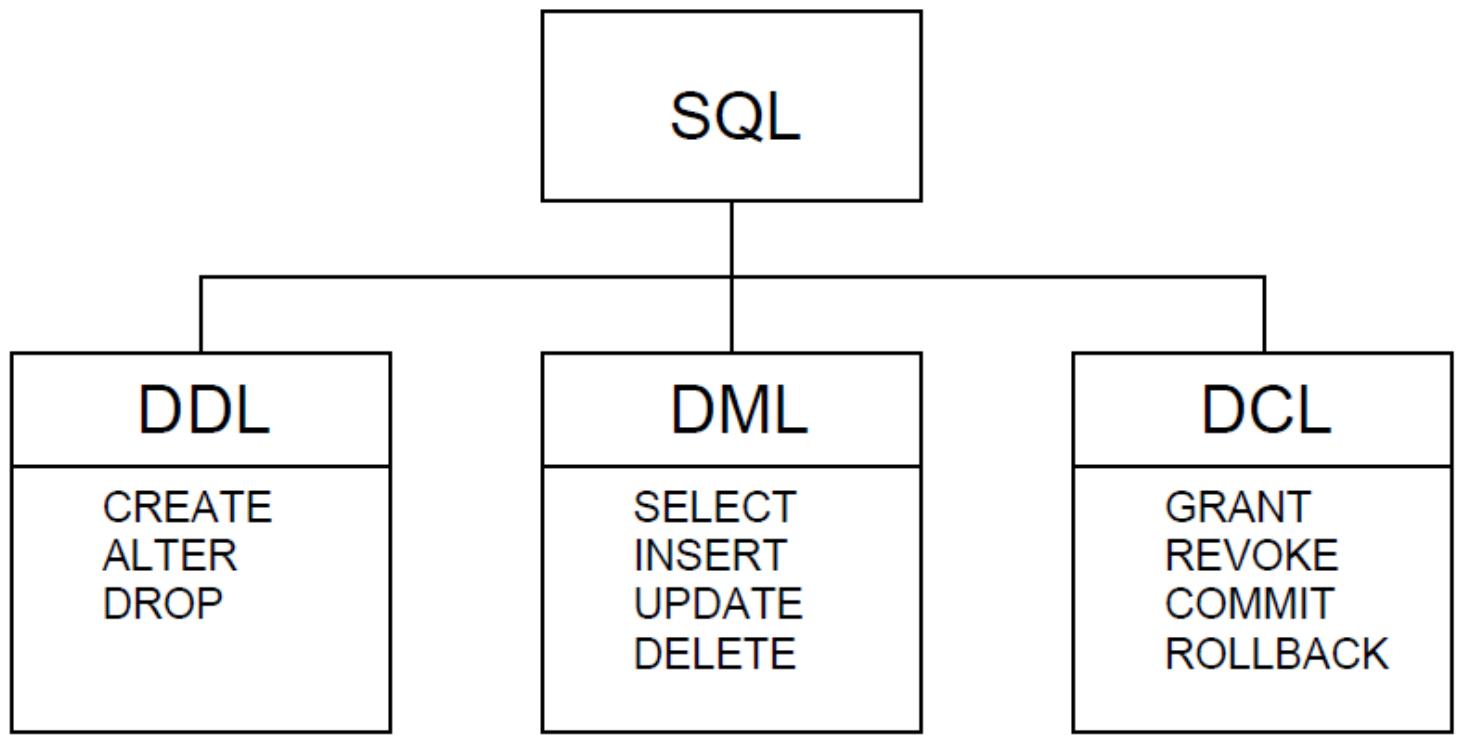
- Un átomo es una fórmula.
- Si P_1 es una fórmula, también lo son $\neg P_1$ y (P_1) .
- Si P_1 y P_2 son fórmulas, también lo son $P_1 \vee P_2$, $P_1 \wedge P_2$ y $P_1 \Rightarrow P_2$.
- Si $P_1(x)$ es una fórmula en x , donde x es una variable de dominio:
 $\exists x (P_1(x))$ y $\forall x (P_1(x))$ también son fórmulas.

El lenguaje SQL (Structured Query Language)

Los lenguajes formales descritos en el epígrafe anterior proporcionan una notación concisa para la representación de las consultas. Sin embargo, los sistemas de BD comerciales necesitan un lenguaje de consulta cómodo para el usuario. SQL es una combinación de álgebra relacional y construcciones de cálculo relacional.

Aunque el lenguaje SQL se considere un lenguaje de consultas, contiene muchas otras capacidades además de la consulta en BD. Incluye características para definir la estructura de los datos, para la modificación de los datos en la BD y para especificación de restricciones de integridad.

El lenguaje SQL es un lenguaje de alto nivel para dialogar con los SGBD-R. Como todo lenguaje de un SGBD, está formado por tres componentes claramente diferenciados, según muestra la figura:



Destacamos algunas de las características principales del lenguaje SQL:

- Utilizado por todo tipo de usuarios:
 - Administradores de BDR.
 - Programadores.
 - Usuarios Finales.
- Lenguaje no procedimental: Se especifica QUÉ se quiere obtener, sin decir CÓMO.
- Permite especificar cualquier consulta.

Lenguaje de Definición de Datos (DDL)

Tipos básicos de datos

- Datos Alfanuméricos o Cadenas de Caracteres:
 - CHAR(longitud), donde: longitud = número máximo de caracteres del campo
 - VARCHAR(longitud)
- Datos Numéricos:
 - SMALLINT, INTEGER
 - DECIMAL ó DECIMAL(precisión, decimal), donde: precisión = número de dígitos del número y decimal = número de dígitos decimales del nº decimal
 - REAL
 - FLOAT
- Datos tipo fecha y tiempo:
 - DATE: Se puede elegir entre varios formatos
 - TIME: También tiene diferentes formatos
 - TIMESTAMP: Su valor es: fecha + tiempo + nanosegundos

Creación y borrado de bases de datos

- Creación de una BD: CREATE DATABASE nombre_base_datos;
- Borrado de la BD: DROP DATABASE nombre_base_datos;

Creación, modificación y borrado de tablas

Creación

```

CREATE TABLE nombre_tabla (
    <definición_atributo_1> [NOT NULL][CHECK Condicion],
    <definición_atributo_2> [NOT NULL][CHECK Condicion],
    .....
    <definición_atributo_n> [NOT NULL][CHECK Condicion],
    [PRIMARY KEY (ListadeAtributos)]
);

```

Donde:

- definición_atributo = nombre_atributo tipo_dato (tamaño)
- NOT NULL: no se permiten valores nulos en la columna
- ListadeAtributos: uno o más atributos separados por comas

Modificación

- Añadir un nuevo atributo:

```
ALTER TABLE <nombre_tabla> ADD <def_atributo>|<def_integridad>;
```

- Modificar un atributo ya existente:

```
ALTER TABLE <nombre_tabla> ALTER <atributo> TYPE <nuevo_tipo>;
```

- Borrar un atributo ya existente:

```
ALTER TABLE <nombre_tabla> DROP <atributo>;
```

Eliminación

```
DROP TABLE <nombre_tabla>;
```

Definición de vistas

Una vista es una estructura tabular no física (tabla virtual), que permite consultar y/o modificar datos de la tabla real.

Las principales características de las vistas son:

- Se utilizan como si fuesen tablas reales.
- No contienen datos propios.
- No tienen asociada estructura física.

Las ventajas del uso de vistas son:

- Menor complejidad en consultas: Permiten obtener igual información de forma más simple.
- Aumento de confidencialidad: Permiten acceder sólo a ciertos datos de las tablas reales.

Las vistas se pueden crear y borrar con las siguientes sentencias:

- Creación de vistas:

```
CREATE VIEW <nombre_vista> [<lista_atributos>] AS (<cláusula SELECT>);
```

Las filas de la vista serán aquellas que resulten de ejecutar la consulta sobre la que está definida.

- Eliminación de vistas:

```
DROP VIEW <nombre_vista>;
```

Creación y borrado de índices

Es el sistema el encargado de utilizar los índices, para optimizar el acceso a los datos, el usuario sólo puede crear o eliminar índices, pero no indicar su utilización.

- Creación de índices:

```
CREATE [UNIQUE] INDEX <nombre_índice> ON <nombre_tabla> (<lista_atributos>);
```

- Eliminación de índices:

```
DROP INDEX <nombre_índice>;
```

Definición de claves referenciales

Justo debajo de PRIMARY KEY cuando estamos creando una tabla:

```
[FOREIGN KEY (<lista_de_columnas>) REFERENCES nombre_de_tabla(<lista_de_columnas>)
ON UPDATE [NO ACTION | SET DEFAULT | SET NULL | CASCADE]
ON DELETE [NO ACTION | SET DEFAULT | SET NULL | CASCADE]]
```

Lenguaje de Manipulación de Datos (DML)

Inserción, actualización y borrado de filas

Inserción

- Inserción de una fila:

```
INSERT INTO <nombre_tabla> [<lista_atributos>] VALUES (<valor1>, ..., <valorN>);
```

- Inserción de varias filas:

```
INSERT INTO <nombre_tabla> [<lista_atributos>] (<cláusula SELECT>);
```

La cláusula “SELECT” especifica una consulta cuyo resultado (filas) se insertará en la tabla especificada.

Modificación

```
UPDATE <nombre_tabla>
    SET <atributo_1> = <valor_1>,
        <atributo_2> = <valor_2>,
        .....
        <atributo_n> = <valor_n>
[WHERE <condición>];
```

La modificación afectará a todas las filas que cumplan la condición, si se especifica ésta. Si no se especifica condición, la modificación afectará a todas las filas de la tabla.

Eliminación

```
DELETE
FROM <nombre_tabla>
[WHERE <condición>];
```

No se pueden eliminar partes de una fila. Si no aparece la cláusula “WHERE” se eliminarán todas las filas de la tabla, no eliminándose la definición de ésta en el esquema.

Operaciones de consulta

- Sintaxis de la sentencia:

```
SELECT [DISTINCT] <expresión>
FROM <lista_de_tablas>
[WHERE <condicion>]
[GROUP BY <lista_de_atributos>
[HAVING <condición_de_grupo> ]]
[ORDER BY <lista_de_atributos> [ASC/DESC]];
```

- - SELECT: especifica la información que se desea obtener.
 - DISTINCT: elimina los valores repetidos.
 - FROM: indica las tablas o vistas en las que se encuentran los atributos implicados en la consulta.
 - WHERE: especifica la condición de búsqueda.
 - GROUP BY: permite agrupar el resultado.
 - HAVING: especifica una condición de grupo.
 - ORDER BY: permite ordenar el resultado.
- Operadores: Los operadores que se pueden utilizar para expresar condiciones de fila (cláusula WHERE) o de grupo (cláusula HAVING) son:
 - De comparación: <, <=, >, >=, <>, =
 - Lógicos: AND, OR, NOT
 - BETWEEN ... AND establece una comparación dentro de un intervalo cerrado. También se puede utilizar NOT BETWEEN.
 - LIKE: establece una comparación entre cadenas de caracteres, también se puede utilizar NOT LIKE, emplea los siguientes comodines:
 - %: sustituye a una cadena de caracteres cualquiera.
 - _: sustituye a un único carácter cualquiera.
 - IN: comprueba la pertenencia de un valor a un conjunto dado.
 - IS NULL: comprueba si un valor determinado es nulo (NULL). También se puede utilizar IS NOT NULL.
 - Cuantificadores: ANY (alguno), ALL (todos).
 - Existencial: EXISTS, indica la existencia o no de un conjunto. También se puede utilizar NOT EXISTS.
- Reglas de Evaluación de Operadores: El Orden de Evaluación es el siguiente:
 - Operadores de Relación: BETWEEN, IN, LIKE, IS, NULL y después NOT, AND, OR.
 - Se pueden utilizar paréntesis para establecer el orden de evaluación deseado por el usuario.
- Consultas con UNION, DIFERENCIA e INTERSECCIÓN:
 - Unión de conjuntos: operador UNION.
 - Diferencia de conjuntos: operador MINUS.
 - Intersección de conjuntos: operador INTERSECT.
- Expresiones en la cláusula SELECT: No sólo se pueden seleccionar atributos, sino expresiones en las que aparezcan atributos y/o constantes y operadores aritméticos.
- Funciones agregadas: Devuelven un valor único, numérico. No se pueden combinar, con columnas que devuelvan más de un valor, a menos que la consulta contenga una cláusula GROUP BY.
 - COUNT (*): contador de tuplas (totalizador)
 - COUNT (DISTINCT Atributo): contador de tuplas (parcial), no tiene en cuenta valores nulos ni duplicados.
 - AVG(Atributo): media aritmética de un atributo numérico.
 - SUM(Atributo): suma de atributos o expresiones numéricas.
 - MAX(Atributo): valor máximo de un atributo o expresión numérica.
 - MIN(Atributo): valor mínimo de un atributo o expresión numérica.

- Cláusula GROUP BY: GROUP BY <lista_de_atributos>
 - Agrupa el resultado, devolviendo una única fila por grupo.
 - El agrupamiento no se realiza ordenado.
 - Los atributos que aparezcan en GROUP BY, deben aparecer en la cláusula SELECT.
- Cláusula HAVING: HAVING <condición_de_grupo>
 - Siempre va acompañada de la cláusula GROUP BY.
 - Especifica una condición de grupo.
- Cláusula ORDER BY: ORDER BY <lista_de_atributos> [ASC | DESC]
 - El resultado de la consulta se ordena en base a los atributos que se indiquen en la lista.
 - Los atributos de ordenación deben aparecer en SELECT.

Lenguaje de Control de Datos (DCL)

Este lenguaje se preocupa principalmente del control de acceso a los datos (seguridad) y del control de la integridad de los datos.

Control de acceso a los datos

- Niveles de acceso soportados por los SGBDR: Se establecen privilegios de acceso por los niveles siguientes:
 - Base de Datos
 - Tabla
 - Atributo
 - Tupla
- Concesión de Privilegios: Permite dar a los usuarios el acceso completo o restringido a la BD:

```
GRANT <privilegio_de_acceso>
[ON <lista_de_objetos>]
TO <lista_de_usuarios>
[WITH GRANT OPTION]
```

- - <privilegio_de_acceso>: CONNECT, RESOURCE, DBA, ALL PRIVILEGES, SELECT, UPDATE, INSERT, DELETE
 - WITH GRANT OPTION concede permiso para que el usuario a su vez, conceda esos permisos a otros usuarios.
- Nivel de Base de Datos: El SGBDR chequea los privilegios del usuario al iniciar la sesión. Los posibles privilegios o permisos son:
 - CONNECT: Conectarse a la BDR.
 - RESOURCE: Crear objetos.
 - DBA: Ejecución de comandos restrictivos. Acceso a cualquier objeto. Privilegio RESOURCE implícito.
- Nivel de Tabla: Las tablas son propiedad del usuario que las creó. Los posibles privilegios o permisos son:
 - DELETE: Autoriza el borrado de tuplas.
 - INSERT: Autoriza la inserción de nuevas tuplas.
 - SELECT: Permite la realización de consultas.
 - UPDATE: Permite la actualización de tuplas.
 - ALL PRIVILEGES: Concede todos los privilegios.
- Niveles Atributo y Tupla: Se implantan a través de la definición de vistas.
 - Nivel de Atributo: Se crea una vista sin condiciones. Se establecen los permisos sobre la vista.
 - Nivel de Tupla. Se crea una vista con sólo las tuplas permitidas. Se establecen los permisos sobre la vista.
- Revocación de privilegios: Se utiliza para anular privilegios ya concedidos a los usuarios:

```
REVOKE <privilegio_de_acceso>
[ON <lista_de_objetos>]
TO <lista_de_usuarios>;
```

Control de integridad

Este control está asociado al concepto de Transacción. Existen dos sentencias principales:

- COMMIT: Los cambios que se puedan estar realizando sobre la BD se hacen fijos únicamente al completar la transacción (COMMIT automático) o al hacer un COMMIT explícito.
- ROLLBACK: Elimina todos los cambios que se hayan podido producir en la BD desde la ejecución de la última instrucción COMMIT. Si se produce un error de programa o un fallo hardware el sistema realiza un ROLLBACK automáticamente.

Estándares de conectividad: ODBC y JDBC

Los programas de aplicación son programas que se usan para interaccionar con la BD. Como ya se comentó, estos programas se escriben usualmente en un lenguaje anfitrión, tal como Cobol, C, C++, Java, etc.

Para acceder a la BDR, las instrucciones del LMD del SQL necesitan ser ejecutadas desde el lenguaje anfitrión. Hay dos maneras de hacerlo:

- SQL incorporado: Extendiendo la sintaxis del lenguaje anfitrión para incorporar las llamadas del LMD dentro del programa del lenguaje anfitrión. Usualmente, un carácter especial o una sentencia concreta precede a las llamadas del LMD y un precompilador LMD, convierte las instrucciones LMD en llamadas normales a procedimientos del lenguaje anfitrión.
- SQL dinámico: Proporcionando una interfaz de programas de aplicación, API (Application Program Interface), que se deben usar para enviar tanto instrucciones LMD, como LDD, a la BD, y recuperar los resultados. Existen dos estándares:
 - El estándar de conectividad abierta de BD (ODBC, Open Data Base Connectivity) definido por Microsoft para el uso con el lenguaje C, usado comúnmente.
 - El estándar de conectividad de Java con BD (JDBC, Java Data Base Connectivity) que proporciona las características correspondientes para el lenguaje Java.

En el resto del apartado, vamos a examinar las dos normas de conexión de BD, ODBC y JDBC, utilizando el lenguaje SQL.

Para comprender estas normas es necesario comprender el concepto de sesión SQL. El usuario o aplicación se conecta a un servidor de BD SQL, estableciendo una sesión. Así todas las actividades del usuario o aplicación están en el contexto de una sesión SQL. Además de las órdenes normales de SQL (LMD y LDD), una sesión también puede contener órdenes para comprometer el trabajo realizado hasta ese momento en la sesión (COMMIT) o para echarlo atrás (ROLLBACK).

Las normas ODBC y JDBC, se desarrollaron para hacer de interfaz entre clientes y servidores. Cualquier cliente que utilice interfaces ODBC o JDBC puede conectarse a cualquier servidor de BD que proporcione esta interfaz.

ODBC

¿Qué es ODBC?

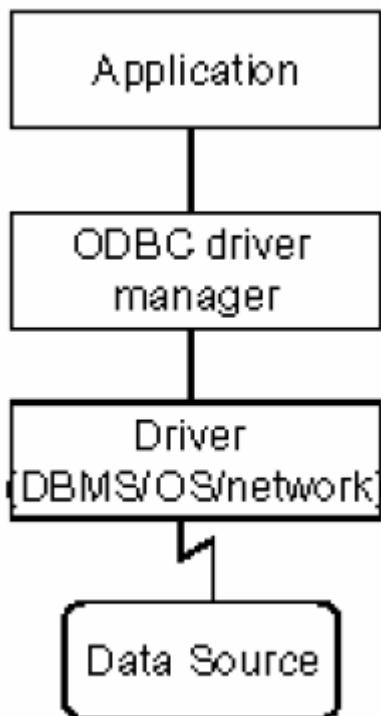
- ODBC son las siglas Open Database Connectivity.
- Es un interface estándar de programas de aplicación (API) para acceso a BD.

- Impulsado por SQL Access Group, principalmente por Microsoft, desde 1992.
- Usando sentencias ODBC en un programa, se puede acceder a las tablas de diferentes BD, tales como: Access, dBase, DB2, etc.
- Permite a los programas utilizar sentencias SQL que acceden a las BD, sin tener que conocer los interfaces propietarios de dichas BD.
- ODBC maneja la sentencia SQL requerida y la convierte en una petición a la BD.
- No soporta el COMMIT en dos fases, para coordinar el acceso simultáneo a varias BD.

ODBC presenta una arquitectura de cuatro niveles:

- La aplicación propiamente dicha.
- ODBC driver manager: Módulo separado por cada BD a la que se quiere acceder. A este módulo es al que se conecta dinámicamente la aplicación.
- Driver DBMS/OS/Network: es un controlador que hace transparente el gestor de BD, el SO y los protocolos de red.
- El propio servidor de datos o fuente de datos.

Esta arquitectura, es la que muestra la figura:



Las aplicaciones como las interfaces gráficas de usuario, los paquetes estadísticos y las hojas de cálculo pueden usar la misma API ODBC, para conectarse a cualquier servidor de BD compatible con ODBC.

Cada sistema de BD que sea compatible con ODBC proporciona una biblioteca que se debe enlazar con el programa cliente. Cuando este programa cliente realiza una llamada a la API ODBC, el código de la biblioteca se comunica con el servidor de BD para realizar la acción solicitada y obtener los resultados.

ODBC se basa en las normas SQL de interface de nivel de llamada, CLI (Call-Level Interface) desarrolladas por el consorcio industrial X/Open y el grupo SQL Access, pero tienen varias extensiones. La API ODBC define una CLI, una definición de sintaxis SQL y reglas sobre las secuencias admisibles de llamadas CLI. La norma también define los niveles de conformidad para CLI y la sintaxis SQL:

- El nivel central de la CLI tiene comandos para conectarse con BD, para preparar y ejecutar sentencias SQL, para devolver resultados o valores de estado y para administrar transacciones.

- El nivel uno, siguiente nivel de conformidad, exige el soporte de la recuperación de información de los catálogos de los SGBD, como la información sobre las relaciones existentes y los tipos de sus atributos, y otras características que superan la CLI del nivel central.
- El nivel dos exige más características, como la capacidad de enviar y recuperar arrays de valores de parámetros y de recuperar información de catálogo más detallada.

JDBC

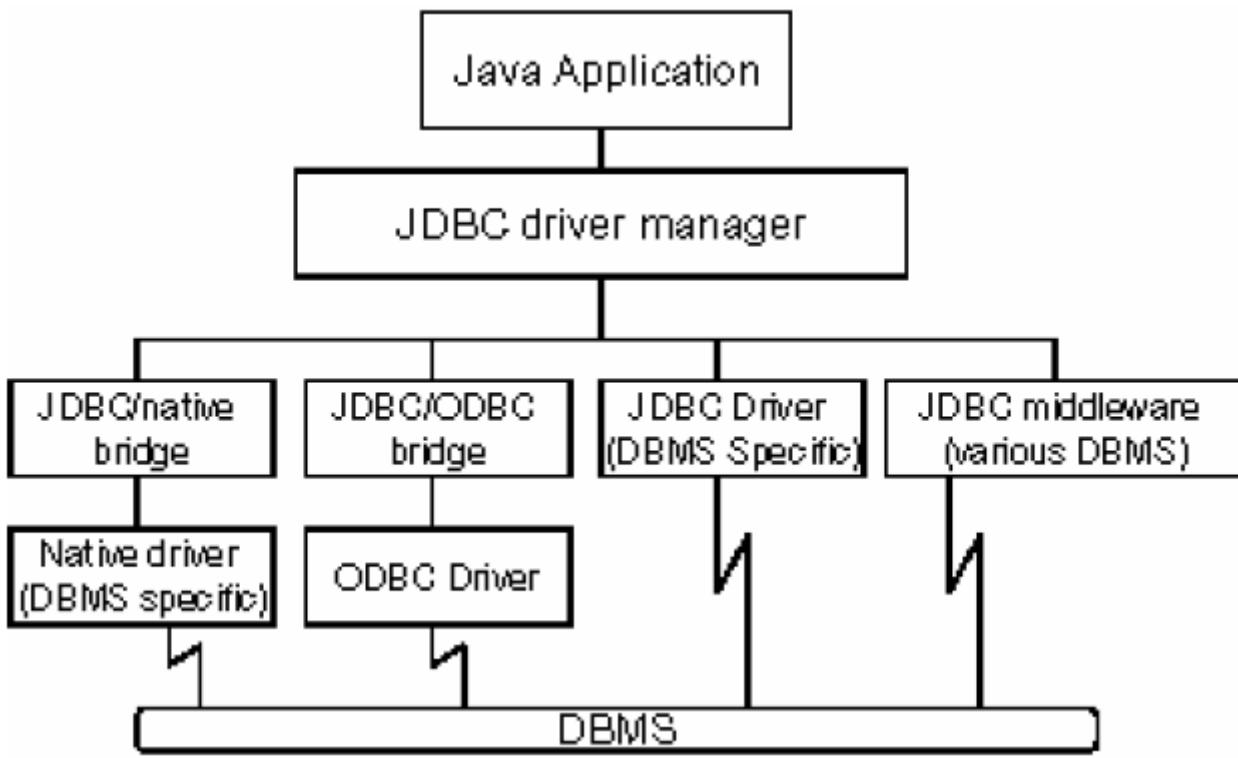
¿Qué es JDBC?

- JDBC son las siglas de Java Database Connectivity.
- Es un Java API, para conectar programas escritos en Java a datos almacenados en SGBDR.
- Consiste en un conjunto de clases e interfaces escritos en el lenguaje de programación Java.
- Suministra un API estándar para los programadores, haciendo posible desarrollar aplicaciones con acceso a BD usando “puro” Java API.
- Este estándar es definido por Sun Microsystems, y permitiendo a los diversos suministradores de BD, implementar y extender dicho estándar con su propios JDBC drivers.
- JDBC establece una conexión con la BD, envía sentencias SQL y procesa los resultados.
- Con un pequeño programa “puente” se puede utilizar el interface JDBC, para acceder a las BD a través de ODBC.

Pasos que hay que realizar en un programa Java utilizando el API JDBC:

- Importación de paquetes: Estos paquetes contienen el propio JDBC y los drivers para una determinada BD.
- Registrar los drivers de JDBC: `DriveManager.registerDriver(new oracle.jdbc.driver.OracleDriver());` En este caso se registra un driver para acceder a Oracle.
- Abrir una conexión a la BD: `Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@aardvark:1256:teach", user, password);` Se indica el tipo de driver, nombre de host, puerto, nombre de la BD, usuario y password, es decir, todo lo necesario para localizar la BD y poder acceder a ella.
- Crear un objeto de tipo Statement: `PreparedStatement pstmt = conn.prepareStatement(x);` En este caso x es la sentencia SQL que se quiere ejecutar.
- Procesar el Result Set que nos ha devuelto la BD.
- Cerrar los objetos creados: Result Set y Statement.
- Cerrar la conexión.

En la figura siguiente, se muestra las cuatro arquitecturas JDBC que existen actualmente:



Se reflejan los cuatro tipos de driver con los que puede trabajar JDBC. Son:

- Driver tipo 1: JDBC/ODBC bridge. Acceso a BD a través del API ODBC.
- Driver tipo 2: JDBC Driver (DBMS específico). Acceso directo a una BD concreta.
- Driver tipo 3: JDBC/native bridge. Acceso a BD a través de un driver Java nativo, que está arrancando en la parte del Servidor.
- Driver tipo 4: JDBC middleware. Acceso directo a varias BD. Soporte de COMMIT en dos fases para coordinar las actualizaciones en las diversas BD.

ESQUEMA - RESUMEN

El modelo relacional constituye la segunda generación de los sistemas de BD.

En 1970 E.F. Codd, basándose en el álgebra y la teoría de conjuntos, propone un nuevo modelo de datos llamado modelo relacional. Sugiere que todos los datos de la BD se podrían representar como una estructura tabular (tablas con columnas y filas, que denominó relaciones) y que esas relaciones se podrían acceder con un lenguaje no procedimental (declarativo). En este tipo de lenguajes, en lugar de escribir algoritmos para acceder a los datos, sólo se necesita un predicado que identifica los registros o combinación de registros deseados. Es más, este nuevo modelo integraba los lenguajes de definición, navegación y manipulación en un solo lenguaje unificado.

El modelo de datos relacional ha proporcionado beneficios inesperados además del aumento de productividad y facilidad de uso. Es muy adecuado para el enfoque cliente/servidor, el procesamiento paralelo y las interfaces gráficas de usuario.

El sistema de gestión de BD (SGBD) es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica. El objetivo primordial de un SGBD es proporcionar un entorno conveniente y eficiente para extraer, almacenar y manipular información de la BD.

El SGBD gestiona centralizadamente todas las peticiones de acceso a la BD, por lo que este paquete funciona como interfaz entre los usuarios y la BD. Además, el SGBD gestiona la estructura física de los datos y su almacenamiento.

Una de las características más importantes de los SGBD es la independencia entre programas y datos. Para asegurar esta independencia es necesario separar la representación física y lógica de los datos, distinción que fue reconocida oficialmente en 1978, cuando el comité ANSI/X3/SPARC propuso una arquitectura de 3 niveles:

- Nivel interno: Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la estructura física de la BD: dispositivos de almacenamiento físico, estrategias de acceso, índices, etc.
- Nivel conceptual: El siguiente nivel de abstracción describe que datos son almacenados realmente en la BD y las relaciones que existen entre los mismos, esto es, describe la BD completa en términos de su estructura de diseño.
- Nivel externo: Nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe sólo una parte de la BD al usuario acreditado para verla.

Los SGBD deben ofrecer lenguajes e interfaces apropiadas para cada tipo de usuario: administradores de la BD, diseñadores, programadores de aplicaciones y usuarios finales. Estos lenguajes son básicamente tres:

- El lenguaje de definición de datos (DDL) define y mantiene la estructura de la BD, es decir, creación, borrado y mantenimiento de BD, tablas, columnas, índices, claves, etc.
- El lenguaje de consulta y manipulación de datos (DML) sirve para obtener, insertar, eliminar y modificar los datos de la BD.
- El lenguaje de Control de Datos (DCL) sirve para trabajar en un entorno multiusuario, donde es importante la protección y la seguridad de los datos y la compartición de datos entre usuarios.

Los principales módulos del SGBD son:

- El compilador del DDL (Data Definition Language)
- El precompilador del DML (Data Manipulation Language)
- El compilador del DML (Data Manipulation Language)
- El procesador de consultas
- El gestor de BD

El modelo de datos relacional fue presentado por Codd en 1970, se basa en la representación del universo del discurso mediante el álgebra relacional.

La estructura básica del modelo relacional es la tabla, que representa una relación, y en la cual se distinguen los siguientes elementos: Relación, Atributos, Dominio, Tuplas, Cardinalidad de la relación y Grado de la relación.

Los operadores asociados al modelo de datos relacional forman el álgebra relacional. Se puede demostrar matemáticamente que ésta álgebra es completa, o sea, que por medio de ella se puede realizar cualquier acceso a la BD. Los operandos con los que trabaja el álgebra son relaciones del modelo relacional y los operadores básicos son: Unión, Diferencia, Producto cartesiano, Proyección y Selección.

En el nivel superior de la descripción del modelo se establecen restricciones adicionales a las propias del modelo relacional que tienen como fin mantener la integridad y validez de los datos almacenados así como aumentar el grado de información que el esquema lógico de datos proporciona. Estas restricciones son dos: Restricciones de Usuario y de Integridad referencial.

Ahora vamos a estudiar los lenguajes formales de consulta de lenguajes "puros". Los tres que se estudian no son cómodos de usar, pero a cambio sirven como base formal para lenguajes de consulta que sí resultan cómodos.

El álgebra relacional es un lenguaje de consulta procedimental. Consta de un conjunto de operaciones que toman como entrada una o dos relaciones y producen como resultado una nueva relación.

Operaciones fundamentales del álgebra relacional:

- Unarias: Porque operan con una sola relación de la tabla. Son:
 - Selección: Si P es un predicado compuesto por operadores lógicos, aritméticos y de comparación y sus operandos son los valores de los atributos de una relación R , entonces la selección $\sigma P(R)$ es el conjunto de tuplas de la relación R que hacen verdadera la condición establecida por el predicado P .
 - Proyección: Si x es un subconjunto de atributos de la relación R , entonces la proyección $\pi x(R)$ es la relación formada por las columnas de R correspondientes a los atributos x .
 - Renombramiento: Dada una expresión E del álgebra relacional, la expresión $\rho x(E)$, devuelve el resultado de la expresión E con nombre x .
- Binarias: Porque operan con dos relaciones. Son:
 - Unión: La unión de dos relaciones R y S ($R \cup S$) es el conjunto formado por todas las tuplas de R más todas las tuplas de S . Este operador sólo se puede aplicar a relaciones del mismo grado y con los mismos atributos.
 - Diferencia de conjuntos: La diferencia de dos relaciones R y S ($R - S$) es el conjunto formado por todas las tuplas de R que no están en S .
 - Producto cartesiano: La definición formal dice: El producto cartesiano de dos operaciones R y S , de grados m y n respectivamente, se denota $R \times S$ y es el conjunto formado por todas las posibles tuplas de $m + n$ atributos en las que los m primeros atributos son de R y los n restantes pertenecen a S .

Otras operaciones derivadas de las fundamentales:

- La operación intersección de conjuntos denotada por \cap , permite buscar las tuplas que estén al tiempo en las dos relaciones sobre las que actúa.
- La operación unión natural forma un producto cartesiano de sus dos argumentos, realiza una selección forzando la igualdad de los atributos que aparecen en ambas relaciones y, finalmente elimina los atributos duplicados.
- La operación asignación, denotada por \leftarrow , actúa de manera parecida a la asignación de los lenguajes de programación.

Las consultas se expresan en el cálculo relacional de tuplas como: $\{ t \mid P(t) \}$, es decir, son el conjunto de todas las tuplas tales que el predicado P es cierto para t . Se utilizar la notación $t[A]$ para denotar el valor de la tupla t en el atributo A y $t \in R$ para denotar que la tupla t está en la relación R .

Hay una segunda forma de cálculo relacional denominada cálculo relacional de dominios. Esta forma utiliza variables de dominio que toman sus valores del dominio de un atributo, en vez de tomarlos de una tupla completa. El cálculo relacional de dominios, sin embargo, se halla estrechamente relacionado con el cálculo relacional de tuplas.

El lenguaje SQL es un lenguaje de alto nivel para dialogar con los SGBD-R. Como todo lenguaje de un SGBD, está formado por tres componentes claramente diferenciados:

- Lenguaje de definición de datos: CREATE, ALTER y DROP.
- Lenguaje de manipulación de datos: INSERT, UPDATE, DELETE y SELECT.
- Lenguaje de control de datos: GRANT, REVOKE, COMMIT y ROLLBACK.

Existen dos estándares de conectividad para SQL:

- El estándar de conectividad abierta de BD (ODBC, Open Data Base Connectivity) definido por Microsoft para el uso con cualquier lenguaje de programación.

- El estándar de conectividad de Java con BD (JDBC, Java Data Base Connectivity) que proporciona las características correspondientes para el lenguaje Java.

¿Qué es ODBC?

- ODBC son las siglas de Open Database Connectivity.
- Es un interface estándar de programas de aplicación (API) para acceso a BD.
- Impulsado por SQL Access Group, principalmente por Microsoft, desde 1992.
- Usando sentencias ODBC en un programa, se puede acceder a las tablas de diferentes BD, tales como: Access, dBase, DB2, etc.
- Permite a los programas utilizar sentencias SQL que acceden a las BD, sin tener que conocer los interfaces propietarios de dichas BD.
- ODBC maneja la sentencia SQL requerida y la convierte en una petición a la BD.
- No soporta COMMIT en dos fases, para coordinar el acceso simultáneo a varias BD.

¿Qué es JDBC?

- JDBC son las siglas de Java Database Connectivity.
- Es un Java API, para conectar programas escritos en Java a datos almacenados en SGBDR.
- Consiste en un conjunto de clases e interfaces escritos en el lenguaje de programación Java.
- Suministra un API estándar para los programadores, haciendo posible desarrollar aplicaciones con acceso a BD usando un "puro" Java API.
- Este estándar es definido por Sun Microsystems, y permitiendo a los diversos suministradores de BD, implementar y extender dicho estándar con sus propios JDBC drivers.
- JDBC establece una conexión con la BD, envía las sentencias SQL y procesa los resultados.
- Con un pequeño programa "puente" se puede utilizar el interface JDBC, para acceder a las BD a través de ODBC.

Arquitectura de sistemas cliente-servidor y multicapas: tipología. Componentes. Interoperabilidad de componentes. Ventajas e inconvenientes. Arquitectura de servicios web (WS).

Arquitecturas Cliente/Servidor

Concepto de Arquitectura Cliente/Servidor

La tecnología Cliente/Servidor es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales.

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información de forma transparente aún en entornos multiplataforma. Se trata pues, de la arquitectura más extendida en la realización de Sistemas Distribuidos.

Un sistema Cliente/Servidor es un Sistema de Información distribuido basado en las siguientes características:

- Servicio: unidad básica de diseño. El servidor los proporciona y el cliente los utiliza.
- Recursos Compartidos: Muchos clientes utilizan los mismos servidores y, a través de ellos, comparten tanto recursos lógicos como físicos.
- Protocolos asimétricos: Los clientes inician “conversaciones”. Los servidores esperan su establecimiento pasivamente.
- Transparencia de localización física de los servidores y clientes: El cliente no tiene por qué saber dónde se encuentra situado el recurso que desea utilizar.
- Independencia de la plataforma Hardware y Software que se emplee.
- Sistemas débilmente acoplados. Interacción basada en envío de mensajes.
- Encapsulamiento de servicios. Los detalles de la implementación de un servicio son transparentes al cliente.
- Escalabilidad horizontal (añadir clientes) y vertical (ampliar potencia de los servidores).
- Integridad: Datos y programas centralizados en servidores facilitan su integridad y mantenimiento.

En el modelo Cliente/Servidor, un servidor (daemon en la terminología sajona basada en sistemas UNIX/LINUX, traducido como “demonio”) se activa y espera las solicitudes de los clientes. Habitualmente, programas cliente múltiples comparten los servicios de un programa servidor común. Tanto los programas clientes como los servidores son con frecuencia parte de un programa o aplicación mayores.

El Esquema de funcionamiento de un Sistema Cliente/Servidor sería:

1. El cliente solicita una información al servidor.
2. El servidor recibe la petición del cliente.
3. El servidor procesa dicha solicitud.
4. El servidor envía el resultado obtenido al cliente.
5. El cliente recibe el resultado y lo procesa.

Componentes de la arquitectura Cliente/Servidor

El modelo Cliente/Servidor es un modelo basado en la idea del servicio, en el que el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios. Además esta relación está establecida en función del intercambio de mensajes que es el únicos elemento de acoplamiento entre ambos.

Cliente

Un cliente es todo proceso que reclama servicios de otro. Una definición un poco más elaborada podría ser la siguiente: cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Se lo conoce con el término front-end.

Éste normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

La funcionalidad del proceso cliente marca la operativa de la aplicación (flujo de información o lógica de negocio). De este modo el cliente se puede clasificar en:

- Cliente basado en aplicación de usuario. Si los datos son de baja interacción y están fuertemente relacionados con la actividad de los usuarios de esos clientes.
- Cliente basado en lógica de negocio. Toma datos suministrados por el usuario y/o la base de datos y efectúa los cálculos necesarios según los requerimientos del usuario.

Servidor

Un servidor es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor se enumeran a continuación:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Puede darse el caso que un servidor actúe a su vez como cliente de otro servidor. Existen numerosos tipos de servidores, cada uno de los cuales da lugar a un tipo de arquitectura Cliente/Servidor diferente.

El término “servidor” se suele utilizar también para designar el hardware, de gran potencia, capacidad y prestaciones, utilizado para albergar servicios que atienden a un gran número de usuarios concurrentes. Desde el punto de vista de la arquitectura cliente/servidor y del procesamiento cooperativo un servidor es un servicio software que atiende las peticiones de procesos software clientes.

Para conectar cliente con servidor y viceversa, existe un software llamado middleware y se ejecuta en ambas partes.

Características

Las características básicas de una arquitectura Cliente/Servidor son:

- Combinación de un cliente que interactúa con el usuario, y un servidor que lo hace con los recursos compartidos. El proceso del cliente facilita la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como si fuese un motor de software que maneja recursos compartidos como bases de datos, impresoras, módems, etc.
- Las tareas del cliente y del servidor tienen distintos requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco y dispositivos de E/S.
- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Existe un clara distinción de funciones basada en el concepto de “servicio”, que se establece entre clientes y servidores.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.

- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.
- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente/Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

Arquitectura multicapa

La *arquitectura multicapa* es también conocida como *arquitectura de procesamiento distribuido*. En este caso, el sistema se descompone en varias capas, de ellas cada una lleva un tipo de procesamiento específico.

Pondremos un ejemplo orientado a Bases de Datos, en la capa más cercana al usuario, se podría tener un programa con interfaces gráficas poderosas para facilitar la actuación de la información de la base de datos a través de ventanas. En la siguiente capa, se podría tener un servidor de internet que llevara el control de todas las páginas de internet que se mostrarían al usuario como interfaces de la aplicación. La tercera capa podría ser un servidor de aplicaciones que contendría las aplicaciones que implementan la lógica de la organización. Por último, una capa contendría al servidor de bases de datos.

Esta arquitectura puede presentar muchas variantes, tanto en la lógica de procesamiento que puede existir en cada capa, como en la distribución que se puede hacer de los programas que implementan estas lógicas en diferentes equipos conectados a la red.

La arquitectura más común en los sistemas de información abarca una interfaz para el usuario y el almacenamiento persistente de datos que se conoce con el nombre de **arquitectura de tres capas. PRESENTACIÓN, LÓGICA y ALMACENAMIENTO.**

La calidad tan especial de la arquitectura de tres capas consiste en aislar la lógica de la aplicación y en convertirla en una capa intermedia bien definida y lógica del software.

- En la capa de presentación se realiza relativamente poco procesamiento de la aplicación.
- Las ventanas envían a la capa intermedia peticiones de trabajo y éste se comunica con la capa de almacenamiento del extremo posterior.
- Esta arquitectura contrasta con el diseño de dos capas, donde, por ejemplo, colocamos la lógica de aplicaciones dentro de las definiciones de ventana, que leen y escriben directamente en una base de datos; no hay una capa intermedia que separe la lógica.

Arquitectura de Servicios Web (WS)

SOA son las siglas de *Service Oriented Architecture* (Arquitectura Orientada a Servicios de cliente) y es el concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

Ofrece una manera bien definida de exposición e invocación de servicios que normalmente se orientan a servicios web, lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

La W3C define “Servicio Web” como un sistema de software diseñado para permitir interoperabilidad máquina a máquina en la red. En general, los servicios web son sólo APIs (Interfaz de Programación de Aplicaciones) que pueden ser accedidas en una red, como internet, y ejecutadas en un sistema de hosting remoto.

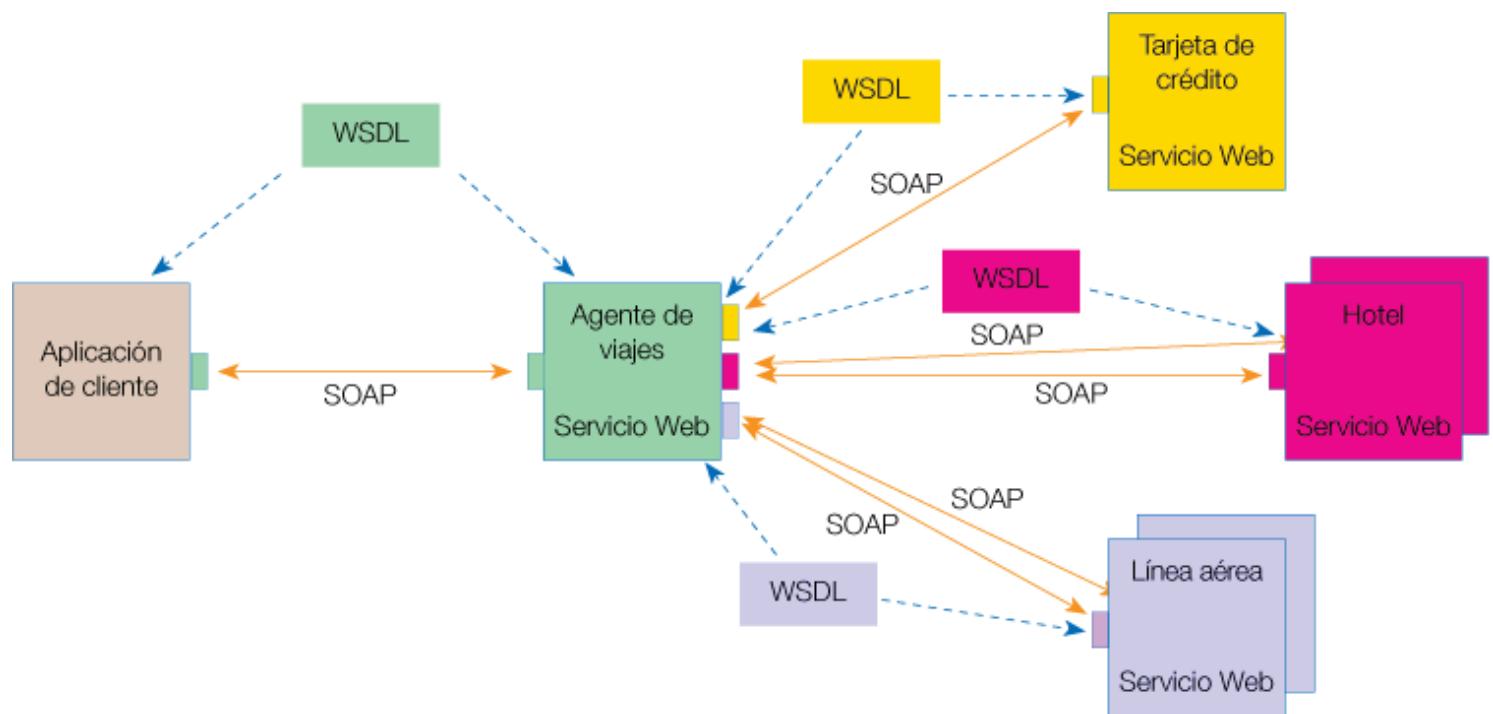
Un servicio web es cualquier sistema de software que se diseña para soportar interacción máquina a máquina sobre una red.

Esta definición es muy amplia y abarca multitud de sistemas muy diferentes, pero en general “servicio web” se suele referir a clientes y servidores que se comunican usando mensajes XML que siguen el estándar SOAP.

En definitiva, permite comunicación entre diferentes máquinas, con diferentes plataformas y entre programas distintos. Esta comunicación se consigue adoptando diversos estándares abiertos.

Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios web se ha creado el organismo WS-I, que se encarga de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

Funcionamiento de los Servicios Web.



Si nos fijamos en la imagen, un usuario (cliente), a través de una aplicación, pide información sobre un viaje que desea realizar haciendo una petición a una agencia de viajes que ofrece sus servicios a través de internet. La agencia de viajes ofrecerá a su cliente (usuario) la información requerida. Para proporcionar al cliente la información que necesita, esta agencia de viajes solicita a su vez información a otros recursos (otros Servicios Web) en relación con el hotel y la compañía aérea. La agencia de viajes obtendrá información de estos recursos, lo que la convierte a su vez en cliente de esos otros Servicios Web que le van a proporcionar la información solicitada sobre el hotel y la línea aérea. Por último, el usuario realizará el pago del viaje a través de la agencia de viajes que servirá de intermediario entre el usuario y el Servicio Web que gestionará el pago.

Durante todo el proceso intervienen una serie de tecnologías que hacen posible esta circulación de información. Por un lado, estaría SOAP (Protocolo Simple de Acceso a Objetos). Este protocolo está basado en XML, y permite la interacción entre varios dispositivos además de ser capaz de transmitir información compleja. Los datos pueden ser

transmitidos a través de HTTP, SMTP, etc. SOAP especifica el formato de los mensajes. El mensaje SOAP está compuesto por un **envelope** (sobre), cuya estructura está formada por los siguientes elementos: **header** (cabecera) y **body** (cuerpo).

INTERNA. El modelo TCP/IP y el modelo de referencia de interconexión de sistemas abiertos (OSI): arquitectura, capas, interfaces, protocolos, direccionamiento y encaminamiento.

El modelo OSI y los protocolos de red

OSI, la pila teórica de protocolos de red

A finales de la década de los setenta, la Organización Internacional para la Normalización (ISO) empezó a desarrollar un modelo conceptual para la conexión en red al que bautizó con el nombre de *Open Systems Interconnection Reference Model* o Modelo de Referencia de Interconexión de Sistemas Abiertos. En los entornos de trabajo con redes se les conoce más comúnmente como el modelo OSI. En 1984 este modelo pasó a ser el estándar internacional para las comunicaciones en red al ofrecer un marco de trabajo conceptual que permitía explicar el modo en que los datos se desplazan dentro de una red.

El modelo OSI divide en siete capas el proceso de transmisión de la información entre equipos informáticos, donde cada capa se encarga de ejecutar una determinada parte del proceso global. Este marco de trabajo estructurado en capas, aun siendo puramente conceptual, puede utilizarse para describir y explicar el conjunto de protocolos reales que, como veremos, se utilizan para la conexión de sistemas. Por ejemplo, TCP/IP y AppleTalk son dos de las pilas de protocolos que se utilizan en el mundo real para transmitir datos; los protocolos que, de hecho, sirven como capas o niveles dentro de un conjunto de protocolos como TCP/IP pueden, por tanto, explicarse de acuerdo con su correlación con el modelo teórico de capas o niveles de red que conforma OSI.

¿Qué es exactamente una pila de protocolos?

Las pilas o suite (o capas) de protocolos no son más que una jerarquía de pequeños protocolos que trabajan juntos para llevar a cabo la transmisión de los datos de un nodo a otro de la red. Las pilas de protocolos se asemejan mucho a las carreras de relevos, pero, en vez de pasarse un testigo se transmiten paquetes de datos de un protocolo a otro hasta que éstos revisten la forma adecuada (una secuencia única de bits) para transmitirse por el entorno físico de la red.

El modelo OSI abarca una serie de eventos importantes que se producen durante la comunicación entre sistemas. Proporciona las normas básicas empíricas para un serie de procesos distintos de conexión en red:

- El modo en que los datos se traducen a un formato apropiado para la arquitectura de red que se está utilizando. Cuando se envía un mensaje de correo electrónico o un archivo a otra computadora, se está trabajando, en realidad, con una determinada aplicación, como un cliente de correo electrónico o un cliente FTP. Los datos que se transmiten utilizando dicha aplicación tienen que convertirse a un formato más genérico si van a viajar por la red hasta llegar a su destino.

- El modo en que los PC u otro tipo de dispositivos de la red se comunican. Cuando se envían datos desde un PC, tiene que existir algún tipo de mecanismo que proporcione un canal de comunicación entre el remitente y el destinatario. Lo mismo que cuando se desea hablar por teléfono, para lo cual hay que descolgar el teléfono y marcar el número.
- El modo en que los datos se transmiten entre los distintos dispositivos y la forma en que se resuelve la secuenciación y comprobación de errores. Una vez establecida la sesión de comunicación entre dos computadoras, tiene que existir un conjunto de reglas que controlen la forma en que los datos van de una a otra.
- El modo en que el direccionamiento lógico de los paquetes pasa a convertirse en el direccionamiento físico que proporciona la red. Las redes informáticas utilizan esquemas de direccionamiento lógico, como direcciones IP. Por tanto, dichas direcciones lógicas tienen que convertirse en las direcciones relaes de hardware que determinan las NIC instaladas en las distintas computadoras.

El modelo OSI ofrece los mecanismos y reglas que permiten resolver todas las cuestiones que acabamos de referir. Comprender las distintas capas del modelo OSI no sólo permite internarse en los conjuntos de protocolos de red que actualmente se utilizan, sino que también proporciona un marco de trabajo conceptual del que puede servirse cualquiera para comprender el funcionamiento de dispositivos de red complejos, como commutadores, puentes y routers.

Las capas OSI

Las capas del modelo OSI describen el proceso de transmisión de los datos dentro de una red. Las dos únicas capas del modelo con las que, de hecho, interactúa el usuario son la primera capa, la capa Física, y la última capa, la capa de Aplicación.

- La **capa física** abarca los aspectos físicos de la red (es decir, los cables, *hubs* y el resto de dispositivos que conforman el entorno físico de la red). Seguramente ya habrá interactuado más de una vez con la capa Física, por ejemplo al ajustar un cable mal conectado.
- La **capa de aplicación** proporciona la interfaz que utiliza el usuario en su computadora para enviar mensajes de correo electrónico o ubicar un archivo en la red.

La figura presenta la estructura de capas que conforman el modelo OSI de arriba abajo. La pirámide invertida es uno de los modos que mejor ilustran la estructura de este modelo, en el que los datos con un formato bastante complejo pasan a convertirse en una secuencia simple de bits cuando alcanzan el cable de la red. Como verá, las capas vienen numeradas de abajo arriba, cuando lo lógico sería que vinieran numeradas de arriba abajo. Éste es el sistema adoptado y, de hecho, muchas veces se alude al mismo para referirse a una de las capas de la red. Pero, tanto si se usa el nombre como el número, lo importante es que recuerde siempre el papel que desarrollan cada una de las capas en el proceso global de transmisión de los datos.

7 Aplicación

6 Presentación

5 Sesión

4 Transporte

3 Red

2 Enlace de Datos

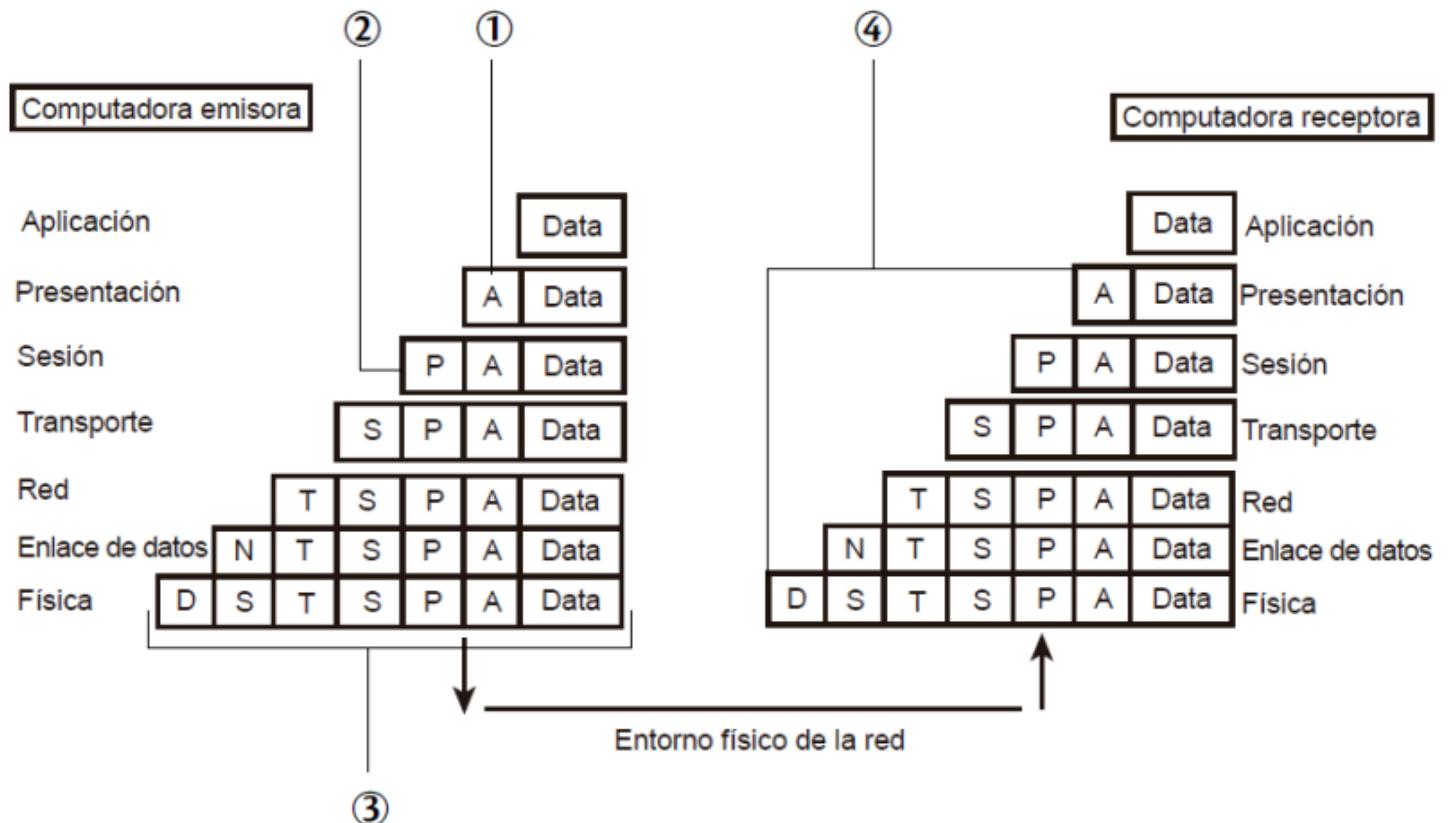
1 Física

El modelo OSI ofrece un modelo teórico que explica el modo en que se desplazan los datos desde una computadora emisora a otra computadora receptora.

Antes de explicar cada una de las capas que componen la pila, conviene hacerse una idea general de lo que ocurre cuando los datos se mueven por el modelo OSI. Supongamos que un usuario decide enviar un mensaje de correo electrónico a otro usuario de la red. El usuario que envía el mensaje utilizará un cliente o programa de correo (como Outlook o Eudora) como herramienta de interfaz para escribir y enviar el mensaje. Esta actividad del usuario se produce en la capa de aplicación.

Cuando los datos abandonan la capa de aplicación (la capa insertará un encabezado de capa de aplicación en el paquete de datos), éstos pasan por las restantes capas del modelo OSI. Cada capa proporcionará servicios específicos relacionados con el enlace de comunicación que debe establecerse, o bien formateará los datos de una determinada forma.

Al margen de la función específica que tenga asignada cada capa, todas adjuntan un encabezado a los datos. Puesto que la capa física está integrada por dispositivos de hardware (un cable, por ejemplo) nunca añade un encabezado de datos.



1. Encabezado de la capa de aplicación.
2. Encabezado de la capa de presentación.
3. Paquete con todos los encabezados de las capas OSI.
4. Los encabezados se van suprimiendo a medida que los datos suben por la capa OSI.

Los datos bajan por la pila OSI de la computadora emisora y suben por la pila OSI de la computadora receptora.

Los datos llegan así a la capa física (en entorno tangible de la red, como los cables de par trenzado y hubs que conectan las computadoras entre sí) de la computadora del destinatario, desplazándose por el entorno físico de la red hasta alcanzar su destino final, el usuario al que iba dirigido el mensaje de correo electrónico.

Los datos se reciben en la capa física de la computadora del destinatario y pasan a subir por la pila OSI. A medida que los datos van pasando por cada una de las capas, el encabezado pertinente se va suprimiendo de los datos. Cuando los datos finalmente alcanzan la capa de aplicación, el destinatario puede utilizar su cliente de correo electrónico para leer el mensaje que ha recibido.

A continuación pasamos a explicar cada una de las capas que componen el modelo OSI, de arriba abajo (es decir, desde la capa de aplicación hasta la capa física).

La capa de aplicación

La capa de aplicación proporciona la interfaz y servicios que soportan las aplicaciones de usuario. También se encarga de ofrecer acceso general a la red.

Esta capa suministra las herramientas que el usuario, de hecho ve. También ofrece los servicios de red relacionados con estas aplicaciones de usuario, como la gestión de mensajes, la transferencia de archivos y las consultas de bases de datos. La capa de aplicación suministra cada uno de estos servicios a los distintos programas de aplicación con los que cuenta el usuario en su computadora. Entre los servicios de intercambio de información que gestiona la capa de aplicación se encuentran la Web, los servicios de

correo electrónico (como el Protocolo Simple de Transferencia de Correo, comúnmente conocido como SMTP - *Simple Mail Transfer Protocol* - incluido en TCP/IP), así como aplicaciones especiales de bases de datos clientes/servidor.

La capa de presentación

La capa de presentación puede considerarse el traductor del modelo OSI. Esta capa toma los paquetes (la creación del paquete para la transmisión de los datos por la red empieza en realidad en la capa de aplicación) de la capa de aplicación y los convierte a un formato genérico que pueden leer todas las computadoras. Por ejemplo, los datos escritos en caracteres ASCII se traducirán a un formato más básico y genérico.

La capa de presentación también se encarga de cifrar los datos (si así lo requiere la aplicación utilizada en la capa de aplicación) así como de comprimirlos para reducir su tamaño. El paquete que crea la capa de presentación contiene los datos prácticamente con el formato con el que viajarán por las restantes capas de la pila OSI (aunque las capas siguientes irán añadiendo elementos al paquete, lo cual puede dividir los datos en paquetes más pequeños).

La comunicación se produce directamente entre capas

A medida que los datos bajan por la pila de protocolos de la computadora emisora (por ejemplo, un mensaje de correo electrónico) hasta llegar al cable físico y de ahí pasan a subir por la pila de protocolos de la computadora receptora, la comunicación entre ambas máquinas se está produciendo en realidad entre capas complementarias. Por ejemplo, cuando se envía un mensaje entre dos computadoras existe entre ellas una comunicación virtual en la capa de sesión. Lo cual es del todo lógico, ya que ésta es la capa que controla la comunicación entre ambas computadoras por el entorno físico de la red (ya sean cables coaxiales, de par trenzado o de fibra óptica).

La capa de sesión

La capa de sesión es la encargada de establecer el enlace de comunicación o sesión entre las computadoras emisora y receptora. Esta capa también gestiona la sesión que se establece entre ambos nodos.



Una vez establecida la sesión entre los nodos participantes, la capa de sesión pasa a encargarse de ubicar puntos de control en la secuencia de datos. De esta forma, se proporciona cierta tolerancia a fallos dentro de la sesión de comunicación. Si una sesión falla y se pierde la comunicación entre los nodos, cuando después se restablezca la sesión sólo tendrán que volver a enviarse los datos situados detrás del último punto de control

recibido. Así se evita el tener que enviar de nuevo todos los paquetes que incluía la sesión.

Los protocolos que operan en la capa de sesión pueden proporcionar dos tipos distintos de enfoques para que los datos vayan del emisor al receptor: la comunicación orientada a la conexión y la comunicación sin conexión.

Para comunicarse, los usuarios tienen que ejecutar el mismo conjunto de protocolos

En el ejemplo anterior del envío y recepción de un mensaje de correo electrónico, dimos por sentado que tanto el remitente como el destinatario estaban ejecutando la misma pila de protocolos (la pila teórica OSI) en sus computadoras clientes. De hecho, las computadoras que ejecuten sistemas operativos distintos pueden comunicarse entre sí si utilizan el mismo conjunto de protocolos de red. Esto es lo que explica que una máquina UNIX, un Macintosh o un PC que esté ejecutando Windows utilicen el TCP/IP para comunicarse en Internet. Un ejemplo en el que dos computadoras no podrían comunicarse sería aquél en que una computadora ejecutara TCP/IP y la otra IPX/SPX. Estos dos protocolos de red del mundo real utilizan reglas distintas y formatos de datos diferentes que hacen que la comunicación resulte imposible.

Los protocolos orientados a la conexión que operan en la capa de sesión proporcionan un entorno donde las computadoras conectadas se ponen de acuerdo sobre los parámetros relativos a la creación de los puntos de control en los datos, mantienen un diálogo durante la transferencia de los mismos, y después terminan de forma simultánea la sesión de transferencia.

Los protocolos orientados a la conexión operan de forma parecida a una llamada telefónica: en este caso, la sesión se establece llamando a la persona con la que se desea hablar. La persona que llama y la que se encuentra al otro lado del teléfono mantienen una conexión directa. Y, cuando la conversación termina, ambos se ponen de acuerdo para dar por terminada la sesión y cuelgan el teléfono a la par.

El funcionamiento de los protocolos sin conexión se parece más bien a un sistema de correo regular. Proporciona las direcciones pertinentes para el envío de los paquetes y éstos pasan a enviarse como si se echaran a un buzón de correos. Se supone que la dirección que incluyen permitirá que los paquetes lleguen a su destino, sin necesidad de un permiso previo de la computadora que va a recibirlas.

La capa de transporte

La capa de transporte es la encargada de controlar el flujo de datos entre los nodos que establecen una comunicación; los datos no sólo deben entregarse sin errores, sino además en la secuencia que proceda. La capa de transporte se ocupa también de evaluar el tamaño de los paquetes con el fin de que éstos tengan el tamaño requerido por las capas inferiores del conjunto de protocolos. El tamaño de los paquetes lo dicta la arquitectura de red que se utilice.

La comunicación también se establece entre computadoras del mismo nivel (el emisor y el receptor); la aceptación por parte del nodo receptor se recibe cuando el nodo emisor ha enviado el número acordado de paquetes. Por ejemplo, el nodo emisor puede enviar de un solo golpe tres paquetes al nodo receptor y después recibir la aceptación por parte del nodo receptor. El emisor puede entonces volver a enviar otros tres paquetes de datos de una sola vez.

Esta comunicación en la capa de transporte resulta muy útil cuando la computadora emisora manda demasiados datos a la computadora receptora. En este caso, el nodo receptor tomará todos los datos que pueda aceptar de una sola vez y pasará a enviar una

señal de “ocupado” si se envían más datos. Una vez que la computadora receptora haya procesado los datos y esté lista para recibir más paquetes, enviará a la computadora emisora un mensaje de “luz verde” para que envíe los restantes.

Cada capa ejecuta funciones de entrada y salida de datos

No debe olvidarse que cada capa del modelo OSI (o de un conjunto real de protocolos de red, como IPX/SPX o TCP/IP) ejecutan funciones relativas a la entrada y salida de información. Cuando los datos bajan por la pila de protocolos en una computadora emisora, la capa de presentación convierte la información procedente de una determinada aplicación a un formato más genérico. En la computadora receptora, la capa de presentación se ocupará de tomar dicha información genérica y de convertirla al formato que utilice el programa que se esté ejecutando en la capa de aplicación de la computadora receptora.

La capa de red

La capa de red encamina los paquetes además de ocuparse de entregarlos. La determinación de la ruta que deben seguir los datos se produce en esta capa, lo mismo que el intercambio efectivo de los mismos dentro de dicha ruta. La Capa 3 es donde las direcciones lógicas (como las direcciones IP de una computadora de red) pasan a convertirse en direcciones físicas (las direcciones de hardware de la NIC, la Tarjeta de Interfaz para Red, para esa computadora específica).

Los *routers* operan precisamente en la capa de red y utilizan los protocolos de encaminamiento de la Capa 3 para determinar la ruta que deben seguir los paquetes de datos.

La capa de enlace de datos

Cuando los paquetes de datos llegan a la capa de enlace de datos, éstos pasan a ubicarse en tramas (unidades de datos), que vienen definidas por la arquitectura de red que se está utilizando (como Ethernet, Token Ring, etc). La capa de enlace de datos se encarga de desplazar los datos por el enlace físico de comunicación hasta el nodo receptor, e identifica cada computadora incluida en la red de acuerdo con su dirección de hardware, que viene codificada en la NIC.

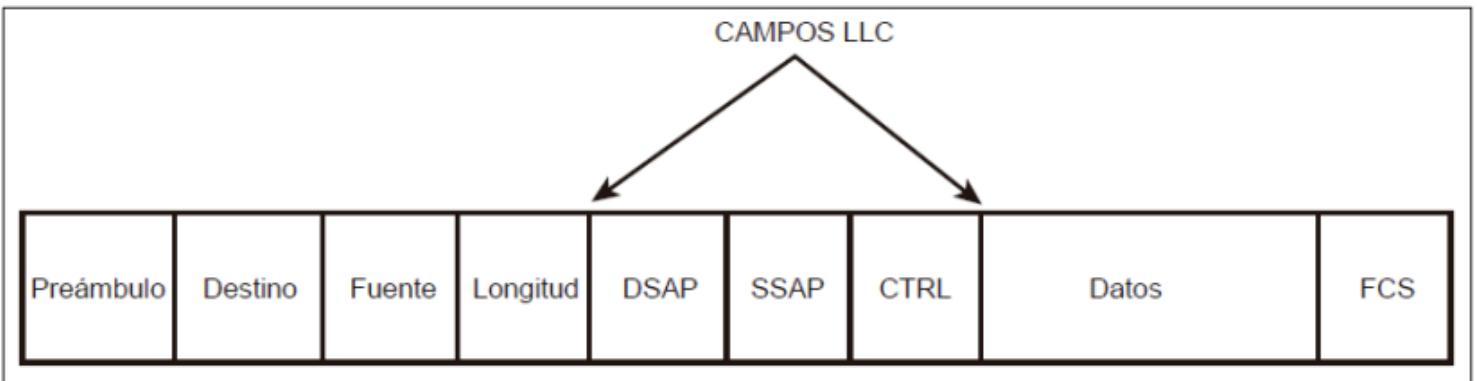
Los protocolos reales utilizan ambos métodos de comunicación: sin conexión y orientados a la conexión

En los conjuntos de protocolos de red, como TCP/IP e IPX/SPX, se utilizan ambas estrategias de comunicación, la que precisa de una conexión y la que no, para desplazar los datos por la red. Por lo general, en la capa de sesión opera más de un protocolo para gestionar estas estrategias distintas de comunicación.

La información de encabezamiento se añade a cada trama que contenga las direcciones de envío y recepción. La capa de enlace de datos también se asegura de que las tramas enviadas por el enlace físico se reciben sin error alguno. Por ello, los protocolos que operan en esta capa adjuntarán un Chequeo de Redundancia Cíclica (*Cyclical Redundancy Check* o CRC) al final de cada trama. El CRC es básicamente un valor que se calcula tanto en la computadora emisora como en la receptora. Si los dos valores CRC coinciden, significa que la trama se recibió correcta e íntegramente, y no sufrió error alguno durante su transferencia.

Una vez más, y tal y como dijimos anteriormente, el tipo de trama que genera la capa de enlace de datos dependerá de la arquitectura de red que se esté utilizando, como Ethernet, Token Ring de IBM o FDDI.

La siguiente figura muestra una trama Ethernet 802.2:



Descripción de cada uno de los componentes:

Segmento	Función
Preámbulo	Bits de alternación (1 y 0) que indican que se ha enviado una trama.
Destino	La dirección de destino.
Fuente	La dirección de origen.
Longitud	Especifica el número de bytes de datos incluidos en la trama.
DSAP	<i>Destination Service Access Point</i> o Punto de Acceso al Servicio de Destino: indica a la tarjeta de red de la computadora receptora dónde tiene que ubicar la trama dentro de la memoria intermedia.
SSAP	Proporciona la información de Punto de Acceso al Servicio (<i>Service Access Point</i>) para la trama (los Puntos de Acceso al Servicio se tratan en más detalle en el apartado “Las subcapas del enlace de datos” incluido en este mismo capítulo).
CTRL	Un campo del Control Lógico del Enlace. (El enlace lógico se explica en más detalle en el apartado “Las subcapas del enlace de datos” incluido en este mismo capítulo.)
Datos	Este segmento de la trama mantiene los datos que se han enviado.
FCS	El campo de Secuencia de Comprobación de la Trama (<i>Frame Check Sequence</i>) contiene el valor CRC para la trama.

La trama se compone básicamente de un encabezado que la describe, de los datos que incluye, y de la información referente a la capa de enlace de datos (como los Puntos de Acceso al Servicio de Destino, *Destination Service Access Points*, y Puntos de Acceso al Servicio, *Service Access Points*), que no sólo definen el tipo de trama de que se trata (en este caso, Ethernet), sino que también contribuyen a que la trama llegue a la computadora receptora.

La capa de enlace de datos también controla la forma en que las computadoras acceden a las conexiones físicas de la red.

La capa física

En la capa física las tramas procedentes de la capa de enlace de datos se convierten en una secuencia única de bits que puede transmitirse por el entorno físico de la red. La capa física también determina los aspectos físicos sobre la forma en que el cableado está enganchado a la NIC de la computadora. En la computadora receptora de datos, la capa

física es la encargada de recibir la secuencia única de bits (es decir, información formada por 1 y 0).

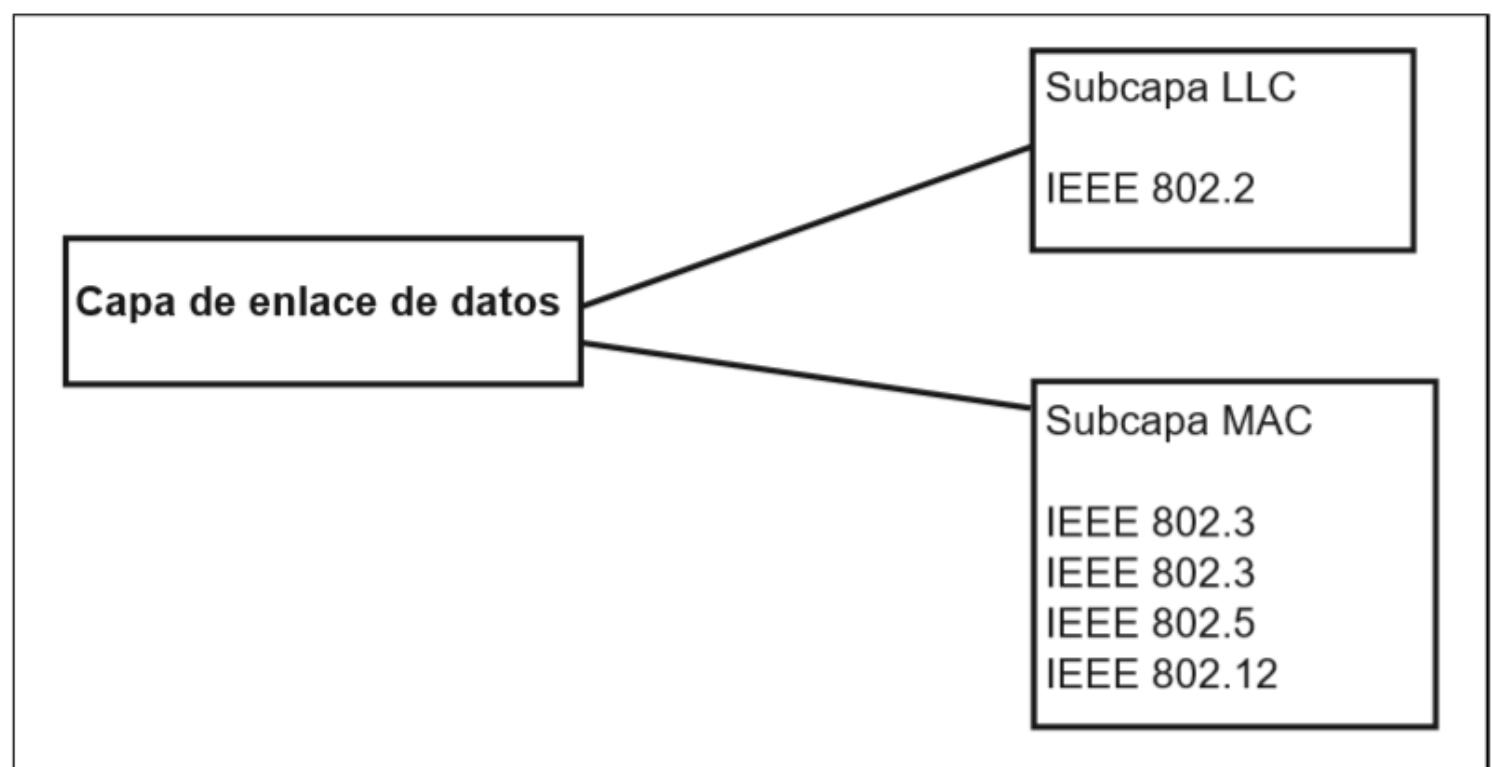
Las subcapas del enlace de datos

La especificación IEEE 802 dividía la capa de enlace de datos en dos subcapas, el Control Lógico del Enlace (*Logical Link Control* o LLC) y el Control de Acceso al Medio (*Media Access Control* o MAC).

La subcapa de Control Lógico del Enlace establece y mantiene el enlace entre las computadoras emisora y receptora cuando los datos se desplazan por el entorno físico de la red. La subcapa LLC también proporciona Puntos de Acceso al Servicio (*Service Access Point* o SAP), que no son más que puntos de referencia a los que otras computadoras que envíen información pueden referirse y utilizar para comunicarse con las capas superiores del conjunto de protocolos OSI dentro de un determinado nodo receptor. La especificación IEEE que define la capa LLC es la 802.2.

La subcapa de Control de Acceso al Medio determina la forma en que las computadoras se comunican dentro de la red, y cómo y dónde una computadora puede acceder, de hecho, al entorno físico de la red y enviar datos. La especificación 802 divide a su vez la subcapa MAC en una serie de categorías (que no son más que formas de acceder al entorno físico de la red), directamente relacionadas con la arquitectura específica de la red, como Ethernet y Token Ring.

La capa de enlace de datos está compuesta por dos subcapas: la subcapa LLC y la subcapa MAC:



Protocolos de red del mundo real

Después de repasar el modelo teórico que determina la forma en que los datos van de una computadora a otra dentro de una red, pasando por las distintas capas que conforman el modelo OSI, podemos pasar a explicar algunos de los conjuntos de protocolos de red más utilizados hoy en día y cotejar las capas que los integran con las del modelo OSI. De esta forma, lograremos una visión clara y sencilla del modo en que operan estas pilas de protocolos reales y de la forma en que transportan los datos por la red.

También veremos qué protocolos de un determinado conjunto participan en la capa de red del modelo OSI. Estos protocolos son de suma importancia ya que contribuyen a encaminar los paquetes en una conexión entre redes.

NetBEUI

NetBEUI (*NetBios Extended User Interface* o Interfaz Ampliada de Usuario para NetBIOS) es un protocolo de red rápido y sencillo que fue diseñado para ser utilizado junto con el protocolo NetBIOS (*Network Basic Input Output System* o Sistema Básico de Entrada/Salida para Red) desarrollado por Microsoft e IBM para redes pequeñas. NetBEUI opera en las capas de transporte y red del modelo OSI.

Puesto que NetBEUI sólo proporciona los servicios que se requieren en las capas de transporte y red de OSI, necesita funcionar con NetBIOS, que opera en la capa de sesión del modelo OSI, y se encarga de establecer la sesión de comunicación entre las dos computadoras conectadas a la red. Las redes Microsoft incluyen además otros dos componentes: el redirector y el Bloque de Mensajes del Servidor (*Server Message Block*). El redirector opera en la capa de aplicación y hace que una computadora cliente perciba todos los recursos de la red como si fueran locales. El Bloque de Mensajes del Servidor (*Server Message Block* o SMB), por su parte, proporciona comunicación de mismo nivel entre los redirectores incluidos en las máquinas cliente y servidor de la red. El Bloque de Mensajes del Servidor opera en la capa de presentación del modelo OSI.

Aunque resulta un excelente protocolo de transporte de bajo coste, NetBEUI no es un protocolo que pueda encaminarse por medio de routers, por lo que no puede utilizarse en las interconexiones de redes. Por tanto, si bien NetBEUI es una opción de protocolo de red para redes pequeñas y sencillas, no resulta válida para redes más amplias que requieren el uso de routers.

Un apunte sobre direcciones hardware

Las direcciones NIC de hardware también se denominan **direcciones MAC**. Esta sigla procede de la expresión ingles *Media Access Control* o Control de Acceso al Medio, y es una de las subcapas de la capa de enlace de datos. Las direcciones de hardware están grabadas en los chips de la memoria ROM en las tarjetas de interfaz para red y cada una de ellas proporciona una dirección única. El esquema de direccionamiento lo desarrolló en su día el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). De acuerdo con este esquema, cada dirección reviste la forma de una cadena de 48 bits escrita en formato hexadecimal. Un ejemplo de dirección MAC sería 00-00-B3-83-B3-3F.

Tramas Ethernet

La trama Ethernet que utilizaban las primeras versiones de NetWare de Novell (NetWare 2.x y 3.x) se creó antes de que el IEEE completara sus especificaciones. Esto hace que el tipo de trama Ethernet 802.3 no se ajuste estrictamente a las normas que ha dictado el IEEE. Las versiones más recientes de NetWare y de otros sistemas operativos de red utilizan la trama Ethernet 802.2, que cumple con todos los requisitos especificados por el IEEE.

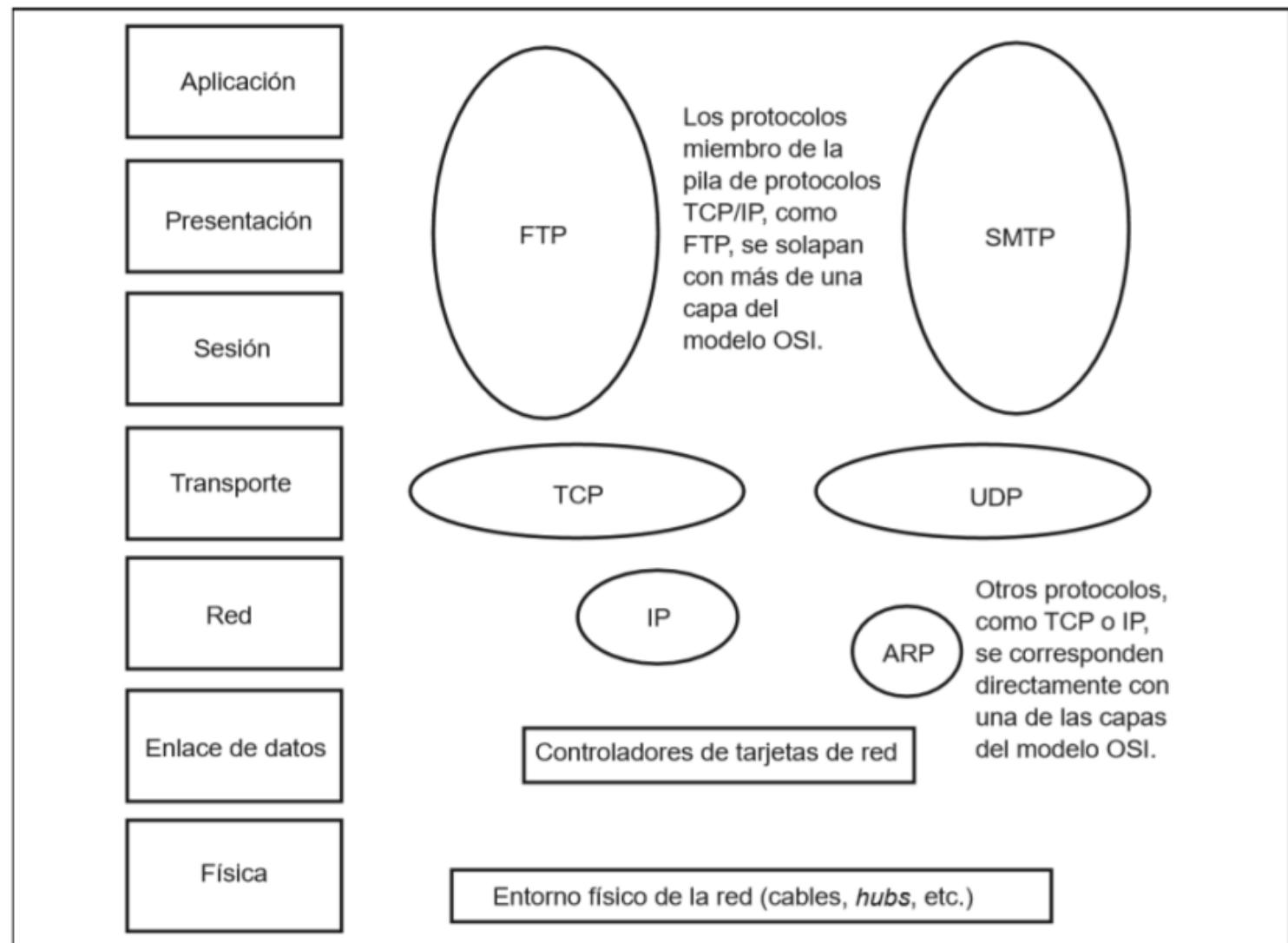
TCP/IP

A menudo referido como el “protocolo de baja puja”, TCP/IP se ha convertido en el estándar *de-facto* para la conexión en red corporativa. Las redes TCP/IP son ampliamente escalables, por lo que TCP/IP puede utilizarse tanto para redes pequeñas como grandes.

TCP/IP es un conjunto de protocolos encaminados que puede ejecutarse en distintas plataformas de software (Windows, UNIX, etc.) y casi todos los sistemas operativos de red lo soportan como protocolo de red predeterminado. TCP/IP consta de una serie de protocolos "miembro" que componen de hecho la pila TCP/IP. Y puesto que el conjunto de protocolos TCP/IP se desarrolló antes de que terminara de desarrollarse el modelo de referencia OSI, los protocolos que lo conforman no se corresponden perfectamente con las distintas capas del modelo.

TCP/IP es un amplio conjunto de protocolos que utiliza una serie de protocolos miembro en varias de las capas del modelo OSI.

Correlación entre el conjunto de protocolos TCP/IP y las capas OSI:



La siguiente tabla describe los protocolos que aparecen en la figura:

Protocolos miembro de la pila TCP/IP.

Protocolo	Función
FTP	El <i>File Transfer Protocol</i> o Protocolo de Transferencia de Archivos proporciona una interfaz y servicios para la transferencia de archivos en la red.
SMTP	El <i>Simple Mail Transport Protocol</i> o Protocolo Simple de Transferencia de Correo proporciona servicios de correo electrónico en las redes Internet e IP.
TCP	El <i>Transport Control Protocol</i> o Protocolo de Control de Transporte es un protocolo de transporte orientado a la conexión. TCP gestiona la conexión entre las computadoras emisora y receptora de forma parecida al desarrollo de las llamadas telefónicas.
UDP	El <i>User Datagram Protocol</i> o Protocolo de Datagrama de Usuario es un protocolo de transporte sin conexión que proporciona servicios en colaboración con TCP.
IP	El <i>Internet Protocol</i> o Protocolo Internet es la base para todo el direccionamiento que se produce en las redes TCP/IP y proporciona un protocolo orientado a la capa de red sin conexión. Funciona de forma semejante a una carta con remite echada al buzón y después entregada a su destinatario.
ARP	El <i>Address Resolution Protocol</i> o Protocolo de Resolución de Direcciones hace corresponder las direcciones IP con las direcciones MAC de hardware. ARP se explica en más detalle en el Capítulo 10.

TCP/IP no sólo proporciona un amplio conjunto de características referidas a la conexión en red (lo cual significa que TCP/IP requiere de una gran carga general para ejecutarse) sino también un sistema de direccionamiento lógico y único. Cualquier usuario que se conecte a Internet estará familiarizado con las direcciones IP de 32 bits, que normalmente se escriben en 4 octetos (un octeto equivale a 8 bits de información). El formato de una dirección es del tipo 129.30.20.4, donde cada uno de los cuatro valores decimales separados por un punto representa 8 bits de información binaria.

Especificaciones 802 del IEEE

Las especificaciones IEEE 802 proporcionan categorías que definen la Capa del Enlace Lógico así como las distintas arquitecturas de red que puede utilizar la subcapa MAC. A continuación se incluye el listado de las categorías 802:

- 802.1 Conexión entre redes.
- 802.2 Control del Enlace Lógico.
- 802.3 LAN Ethernet (CSMA/CD).
- 802.4 LAN Token Bus.
- 802.5 LAN Token Ring.
- 802.6 Red de Área Metropolitana.
- 802.7 Grupo Técnico Asesor sobre Banda Ancha.
- 802.8 Grupo Técnico Asesor sobre Fibra Óptica.
- 802.9 Redes Integradas de Voz y Datos.
- 802.10 Seguridad de Red.
- 802.11 Redes Inalámbricas.
- 802.12 LAN de Demanda de Prioridad.

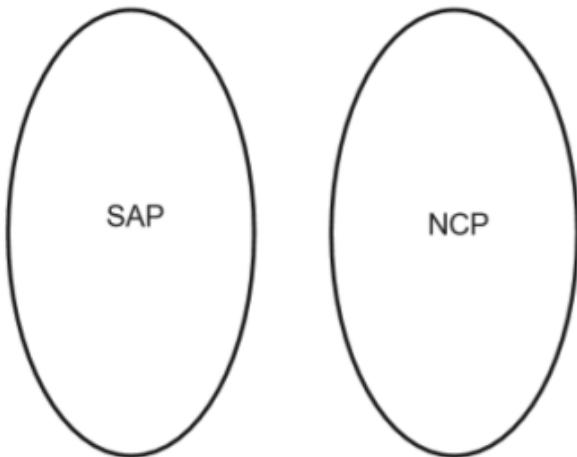
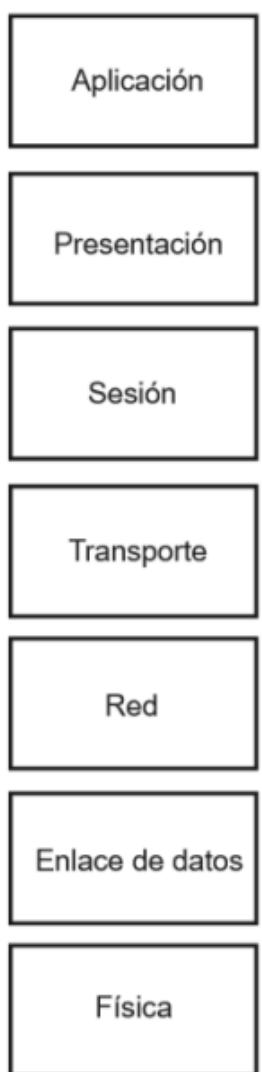
Orígenes de TCP/IP

TCP/IP lo desarrolló la Agencia de Defensa de Proyectos Avanzados de Investigación (DARPA) a petición del Departamento de Defensa de Estados Unidos. Dicho departamento necesitaba un conjunto de protocolos que pudieran utilizarse en cualquier sistema operativo, ya que no existía uniformidad alguna entre los sistemas informáticos de sus oficinas. Y ello por la forma misma en que funcionaba el Departamento, que licitaba todos sus proyectos y servicios. De ahí que de forma coloquial se conozca TCP/IP como protocolo de baja puga, ya que surgió a raíz de la práctica del gobierno estadounidense por pujar.

IPX/SPX

IPX/SPX (*Internetwork Packet Exchange/Sequenced Packet Exchange* o Intercambio de Paquetes entre Redes/Intercambio Secuenciado de Paquetes) es un conjunto de protocolos de red desarrollado por Novell para ser utilizado en su sistema operativo de red NewWare. IPX/SPX agrupa menos protocolos que TCP/IP, por lo que no requiere la misma carga general que TCP/IP necesita. IPX/SPX puede utilizarse tanto en redes pequeñas como grandes y también permite el encaminamiento de datos.

Correlación entre la pila IPX/SPX y las capas del modelo OSI:



Los protocolos SAP y NCP del conjunto IPX/SPX gestionan, de hecho, las tareas de tres capas OSI: aplicación, presentación y sesión.



IPX es un protocolo sin conexión y opera en las capas de transporte y red. SPX, que es un protocolo orientado a la conexión, opera en la capa de transporte.

Controladores de interfaz para red

Entorno físico

IPX/SPX es un conjunto de protocolos eficaz que se utiliza tanto en redes grandes como pequeñas.

Descripción breve de cada uno de los protocolos que lo componen:

Protocolo	Función
SAP	El <i>Service Advertising Protocol</i> o Protocolo de Anuncio de Servicio lo utilizan los servidores de archivo y los servidores de impresora de NetWare para anunciar la dirección del servidor.
NCP	El <i>NetWare Core Protocol</i> o Protocolo de Núcleo NetWare gestiona las funciones de red en las capas de aplicación, presentación y sesión. Gestiona además la creación de paquetes y se encarga de proporcionar servicios de conexión entre los clientes y servidores.
SPX	El <i>Sequenced Packet Exchange Protocol</i> o Protocolo de Intercambio Secuenciado de Paquetes es un protocolo de transporte orientado a la conexión.
IPX	El <i>Internetwork Packet Exchange Protocol</i> o Protocolo de Intercambio de Paquetes entre Redes es un protocolo de transporte sin conexión que gestiona el direccionamiento y encaminamiento de los datos en la red.

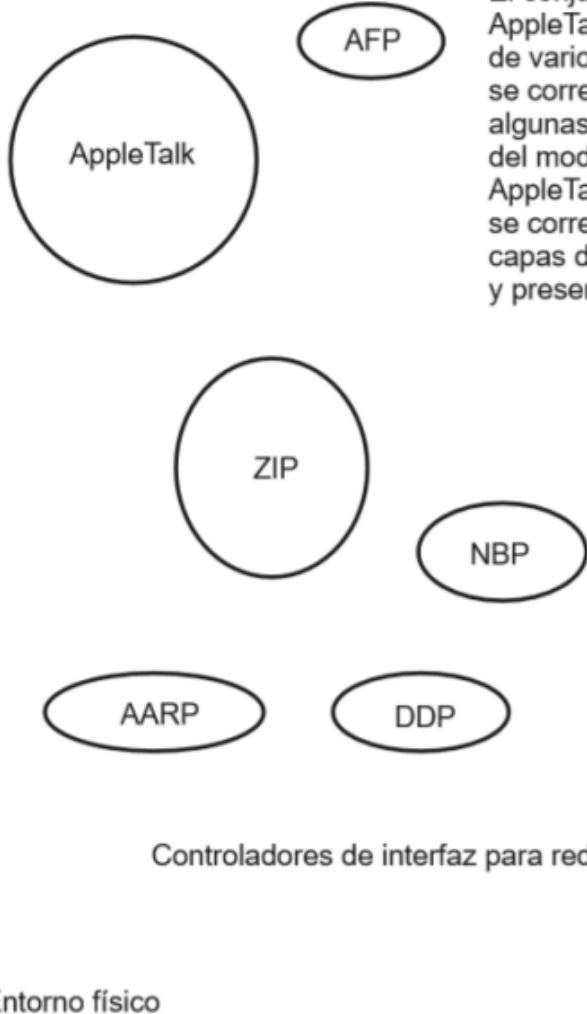
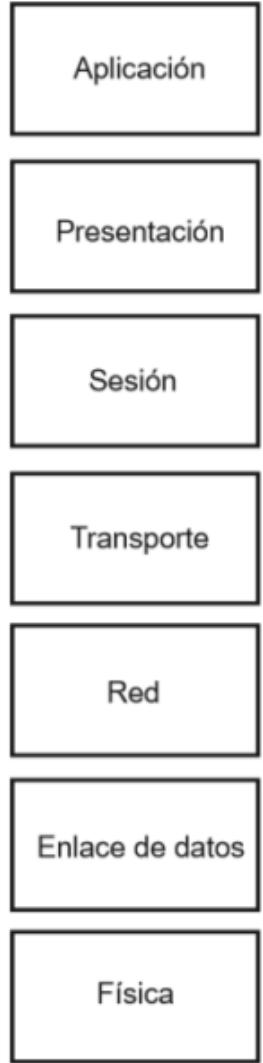
Lo que más nos interesa acerca de IPX/SPX es la forma en que debe encaminarse este conjunto de protocolos dentro de una conexión entre redes.

AppleTalk

Aunque muchos administradores de red no consideran AppleTalk un protocolo de red corporativo o de interconexión, AppleTalk permite el encaminamiento de datos mediante *routers*. De hecho, con el tipo apropiado de NIC (los Macintosh de Apple pueden conectarse a una red Ethernet si cuentan con tarjetas EtherTalk u otro tipo de adaptadores) AppleTalk puede soportar arquitecturas Ethernet, Token Ring y FDDI. Las computadoras Macintosh suelen utilizarse en los entornos empresariales para la manipulación de gráficos y otras tareas de tipo multimedia, por lo que no resulta nada descabellado incluir AppleTalk como otro protocolo encaminado a la red corporativa.

AppleTalk es una arquitectura de red, pero lo cierto es que también se trata de un conjunto de protocolos.

Correlación entre los protocolos que integra AppleTalk y las capas del modelo OSI:



El conjunto de protocolos AppleTalk se compone de varios protocolos, que se corresponden con algunas de las capas del modelo OSI. AppleTalk, por ejemplo, se corresponde con las capas de aplicación y presentación de OSI.

AppleTalk es un conjunto de protocolos encaminados para las redes Macintosh que pueden comunicarse con rede Ethernet, Token Ring y FDDI.

La siguiente tabla describe brevemente cada uno de estos protocolos:

Protocolo	Función
AppleShare	AppleShare proporciona servicios en la capa de aplicación.
AFP	El <i>AppleTalk Filing Protocol</i> o Protocolo de Archivo AppleTalk proporciona y gestiona la compartición de archivos entre nodos de una red.
ATP	El <i>AppleTalk Transaction Protocol</i> o Protocolo de Transacción AppleTalk proporciona la conexión de capa de transporte entre computadoras.
NBP	El <i>Name Binding Protocol</i> o Protocolo de Enlace de Nombre hace corresponder los nombres de servidores de red con las direcciones de la capa de red.
ZIP	El <i>Zone Information Protocol</i> o Protocolo de Información de Zona controla las zonas AppleTalk y hace corresponder los nombres de zonas con las direcciones de red.
AARP	El <i>AppleTalk Address Resolution Protocol</i> o Protocolo de Resolución de Direcciones AppleTalk hace corresponder las direcciones de la capa de red con las direcciones del hardware de enlace de datos.
DDP	El <i>Datagram Delivery Protocol</i> o Protocolo de Entrega de Datagramas proporciona el sistema de direccionamiento para la red AppleTalk, así como el transporte sin conexión de los datagramas entre las distintas computadoras.

Lenguajes de marca o etiqueta. Características y funcionalidades. SGML, HTML, XML y sus derivaciones. Lenguajes de script.

Lenguajes de Marca o Etiqueta

La idea de separar en un documento el contenido del formato apareció por primera vez en los años 60. Empieza a utilizarse el concepto de “marca” o “etiqueta” como texto que se añade a los datos y que proporciona información sobre ellos. Un lenguaje de marca sería un modo formalizado de proporcionar estas marcas.

La definición de un lenguaje de marca debe proporcionar:

- Qué marcas están permitidas.
- Qué marcas son obligatorias.
- Como se distinguen las marcas del texto.
- El significado de cada marca.

El primer modo de utilización de marcas que se exploró fueron las marcas procedurales. Se preocupaban sobre todo de la presentación y del formato del texto. Con las marcas procedurales se instruye al agente en qué hacer con el texto y especifica como procesarlo. Entre los lenguajes de marca procedurales se encuentran RTF y PostScript.

A finales de los años 60 la GCA, Graphic Communication Association, crea un comité de estudio que llega a la conclusión de que las marcas deberían ser más descriptivas que procedurales y deberían tener en cuenta la estructura del documento. Las marcas descriptivas, también llamadas generalizadas, además de identificar la estructura del documento se centran en qué es el texto y no especifican los procedimientos que hay que aplicarle.

A finales de los 60 IBM retoma los trabajos de la GCA y crea un lenguaje de marca al que llamaron GML (Generalized Markup Language, que además coincide con las iniciales de sus creadores Goldfarb, Mosher y Lorie). GML estaba basado en las ideas de codificación genérica, pero en vez de tener un simple esquema de etiquetas, introduce el concepto de tipo de documento formalmente definido con una estructura explícita de elementos anidados.

A finales de los 70 ANSI crea un comité para elaborar una norma basada en el lenguaje GML. El primer borrador aparece en 1980. En 1984 ISO se une al grupo de trabajo de ANSI y en 1986 SGML (Standard Generalized Markup Language) se convierte en estándar internacional (ISO 8879/1986). “SGML es el estándar internacional para la definición de métodos de representación de textos en forma electrónica, independientes del dispositivo e independientes del sistema”.

SGML, más que un lenguaje, es un metalenguaje para la definición y la estandarización de la estructura de los documentos. Define una gramática con la que se pueden crear otros lenguajes de marca. Los lenguajes de marca específicos escritos de acuerdo con los requerimientos de SGML se llaman aplicaciones SGML.

Uno de esos lenguajes derivados de SGML es HTML, Hypertext Markup Language. En 1989, Tim Berners-Lee creó una propuesta de un sistema de documentos hipertexto para ser usado en la comunidad del CERN (Conseil Européen pour la Recherche Nucléaire). Este sistema al que denominó posteriormente “The World Wide Web” perfilaba los componentes básicos necesarios que definen la WWW hoy:

- Ser independiente del sistema.
- Ser capaz de utilizar muchos de los recursos de información existentes y permitir añadir de una manera sencilla nueva información.
- Contar con un mecanismo de transporte de los documentos entre diferentes redes (HTTP).
- Contar con un esquema de identificación para el direccionamiento de documentos de hipertexto tanto local como remoto.

Berners-Lee definió y desarrolló el lenguaje HTML, usando SGML, durante el desarrollo del primer navegador Web. En 1991 puso el código y las especificaciones del sistema, incluyendo HTML, en Internet. Rápidamente empezó a haber navegadores para una amplia variedad de plataformas y según crecía el número de implementaciones, así crecía la variedad de las mismas. El HTML especificado inicialmente (versión 1) se desarrolló más allá de su forma inicial sin que se hubiera creado todavía un estándar real.

HTML 2.0 se convirtió en 1995 en el estándar oficial del lenguaje y fue considerado, en general, una decepción. En aquel año ya estaba en el mercado la versión 2.0 de Navigator, el navegador de Netscape, con implementaciones de marcos y referencias multimedia, muy por encima de las restricciones que imponía el estándar. La especificación estándar de HTML llegó a ser la versión 4.

Sin embargo pronto se puso de manifiesto, sobre todo después de la guerra de etiquetas entre Microsoft y Netscape y con la llegada de las nuevas aplicaciones Web, que eran necesarios medios más flexibles y mejores que los proporcionados por HTML para describir los datos.

En 1996, el W3C, patrocinó un grupo de expertos en SGML para definir un lenguaje de marca con la potencia de SGML y la simplicidad de HTML. Se eliminaron las partes más críticas de SGML y lo considerado no esencial. El resultado fue la especificación de XML. XML es un subconjunto de SGML y al igual que él, más que un lenguaje de marca es un metalenguaje, es un instrumento para crear otros lenguajes de marca.

Características y Funcionalidades

Podemos considerar dos grupos en los lenguajes de marca: los procedurales y los descriptivos o generalizados.

Los lenguajes de marca procedurales se caracterizan por:

- Contener instrucciones claras para el programa visualizador o impresor sobre la forma en que se debe mostrar el contenido del documento, con un determinado estilo y formato.
- Dependencia de las instrucciones de formateado del medio de presentación, de forma que el documento final, el formado por el contenido original más las marcas, no es portable a otros medios.

Por otra parte, los lenguajes de marca generalizados se caracterizan por:

- Marcar las estructuras del texto: las marcas se dirigen más a describir el contenido del documento, su estructura lógica, que a expresar instrucciones detalladas de formateado.
- Separar la estructura del documento de su aspecto.
- Ser independientes del software que procesa el documento.
- Facilitar el procesamiento automático de la extracción de la información contenida en los documentos.
- Facilitar la generación de visualizaciones y presentaciones del documento: permiten la presentación de diferentes vistas del documento dependiendo del dispositivo de visualización y de las preferencias del usuario.
- Utilizar hojas de estilo: la especificación de cómo presentar las estructuras lógicas, aunque a veces es implícita, generalmente es explícita mediante hojas de estilo del documento.
- Tener soporte para lenguajes de script: permiten el uso de lenguajes de script para añadir funcionalidades no proporcionadas por el lenguaje, generalmente comportamiento interactivo y dinámico.
- Proporcionar medios para determinar si un documento es válido o no, es decir si se ajusta a las reglas gramaticales del lenguaje.

En cuanto a las funcionalidades de los lenguajes de marca generalizados se pueden citar:

- Permiten la publicación de documentos electrónicos en Web.
- Enlace de información a través de enlaces hipertexto, permiten recuperar la información on-line.
- Inclusión de otras aplicaciones, como hojas de cálculo, música, ... etc en un documento.
- Diseño de formularios para transacciones electrónicas con otros servicios remotos.

SGML

Dos principios significativos que caracterizan a SGML son:

- Independencia de la representación de la información respecto de los programas y sistemas que la crean y procesan. De aquí su énfasis en las marcas generalizadas o descriptivas, y no procedurales. Con la utilización de marcas descriptivas el mismo

documento se puede procesar por software diferente, aplicando diferentes instrucciones de procesamiento a cada una de las partes del documento o el mismo procesamiento a diferentes partes.

- Cubrir la necesidad de tener más de una representación para la misma información: una representación abstracta o lógica de la información y una representación de almacenamiento.

SGML proporciona una herramienta conceptual para el modelado de información estructurada, el documento, y una notación para la representación de estos documentos. Los documentos se representan por medio de tres constructores básicos:

- Los **elementos** : son bloques estructurales, vistos desde un punto de vista lógico o abstracto, que forman parte del documento (por ejemplo cabeceras, párrafos, tablas, enlaces de hipertexto) y contienen datos, otros elementos subordinados o ambas cosas al mismo tiempo.
- Los **atributos** : son propiedades asociadas con un tipo de elemento dado cuyo valor describe a un elemento de ese tipo pero que no forman parte de su contenido (por ejemplo la longitud de una tabla).
- Las **entidades** : son unidades de almacenamiento virtual que contienen una parte del documento o varias partes o el documento completo. SGML define una entidad como una cadena de caracteres que puede ser referenciada como una unidad.

Para obtener la representación abstracta de la información SGML permite definir la estructura de un documento basándose en la relación lógica entre sus partes. Para marcar esta estructura del documento, en vez de considerar un simple esquema de marcas, introduce los conceptos de tipo de documento y definición de tipo de documento (DTD). Lo mismo que cualquier otra clase de objeto procesada por un ordenador, SGML considera que los documentos tienen tipos. El tipo de documento se define formalmente por su estructura lógica, por las partes que lo constituyen. Una definición de tipo de documento es una especificación formal de la estructura de elementos anidados que componen un documento SGML.

La distinción entre la representación lógica y la representación de almacenamiento de la información es inherente al procesamiento de la información (si un documento se representa mediante un único volumen o como varios más pequeños es independiente de su representación lógica; y por otra parte hay propiedades del documento, como su localización, que no se ven afectadas por un cambio en su estructura lógica).

El proporcionar una representación de almacenamiento de la información entra en conflicto con la independencia respecto al sistema. Para resolver este conflicto SGML proporciona la estructura de entidades del documento como un sistema de almacenamiento virtual que interpone entre la estructura de elementos y el sistema real de almacenamiento (el sistema de ficheros, la BD o cualquiera que sea el sistema real).

Como todo lenguaje SGML tiene una sintaxis. El estándar impone unas reglas formales para distinguir las marcas del contenido del documento, y para el uso y colocación de las marcas. Estas reglas constituyen la sintaxis de SGML. Pero el estándar no impone nombres particulares para los elementos del documento, sólo el concepto de elemento (en realidad se le denomina tipo de elemento), ni impone ninguna estructura particular, sólo una definición formal de la estructura cualquiera que sea ésta.

Tipos de sintaxis de SGML. Sintaxis concreta de referencia.

En SGML se han definido dos tipos de sintaxis:

- Una **sintaxis abstracta** , que en términos de conceptos abstractos, tales como los roles de los delimitadores y de los conjuntos de caracteres, define como se deben construir las marcas SGML.

- Una **sintaxis concreta**, que define como se han codificado estos conceptos abstractos en una clase específica de documentos SGML.

Dentro del estándar ISO de SGML se ha establecido formalmente una sintaxis concreta particular, las sintaxis concretas de referencia.

La definición formal es la siguiente:

SYNTAX	SHUNCHAR	CONTROLS	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127 255		
	BASESET	"ISO 646-1983//CHARSET International Reference Version (IRV)//ESC 2/5 4/0"			
	DESCSET	0 128 0			
	FUNCTION	RE 13 RS 10 SPACE 32 TAB SEPCHAR 9			
	NAMING	LCNMSTR UCNMSTR LCNMCHAR UCNMCHAR NAMECASE	"" "" "-." "-." GENERAL ENTITY YES SGMLREF NO		
	DELIM	GENERAL SHORTREF	SGMLREF SGMLREF		
	NAMES	SGMLREF			
	QUANTITY	SGMLREF			

Contiene 8 subcláusulas que definen:

- Los números decimales de los códigos que se van a ignorar porque son caracteres de control (SUNCHAR).
- El conjunto de caracteres de la sintaxis, que consiste en la declaración del conjunto base de caracteres (BASESET), seguido de cómo se van a usar estos caracteres para definir la sintaxis correcta (DECSET). El conjunto base de caracteres es el definido en ISO 646 y el DECSET muestra que los 128 caracteres, empezando en la posición 0 de la lista, deberían asociarse a posiciones idénticas en la sintaxis concreta de referencia.
- Los códigos que representan caracteres de función requeridos por la sintaxis (FUNCTION). Se definen 4 códigos: el fin e inicio de registro, el espacio y el tabulador horizontal.
- Las reglas que se van a aplicar cuando se definen nombres de elementos, atributos o entidades (NAMING). La sintaxis concreta de referencia sólo permite utilizar los caracteres az, A-Z, 0-9, . (punto) y - (guión) para los nombres y tienen que comenzar con un carácter alfabético. Por defecto SGML suponer que los nombres empiezan con un carácter alfabético, seguido de caracteres alfanuméricos. Las entradas en LCNMSTR y UCNMSTR permiten indicar qué otros caracteres se van a permitir como carácter inicial de los nombres y las entradas en LCNMCHAR y UCNMCHAR qué caracteres no alfanuméricos se pueden utilizar después del carácter inicial. Las entradas en NAMECASE indican que se permite la sustitución de caracteres en

minúsculas por caracteres en mayúsculas en los nombres de elementos y marcas relacionadas, pero no en los nombres de entidad, es decir, los nombres de entidad son sensibles al uso de mayúsculas y minúsculas.

- Los delimitadores de marcas que se van a usar en el documento (DELIM), en este caso, al igual que en las dos subcláusulas siguientes, son los que proporciona SGML por defecto. Algunos de los delimitadores de marca generales proporcionados por defecto se muestran en la tabla siguiente.
- Las palabras reservadas utilizadas en las declaraciones de marcas (NAMES).
- El “conjunto cantidad” (quantity set) requerido por el documento (QUANTITY).

Delimitadores de la sintaxis concreta de referencia:

Carácter	Nombre	Propósito
&	ERO	Apertura de referencia de entidad
&	AND	Conector Y
%	PERO	Apertura de referencia de entidad de parámetro
;	REFC	Cierre de referencia de entidad
<	STAGO	Apertura de etiqueta de inicio
</	ETAGO	Apertura de etiqueta de fin
<!	MDO	Apertura de declaración de marca
>	TAGC	Cierre de etiqueta
>	MDC	Cierre de declaración de marca
(GRPO	Apertura de grupo (en una declaración)
)	GRPC	Cierre de grupo (en una declaración)
[DSO	Apertura de subconjunto de declaraciones
]	DSC	Cierre de subconjunto de declaraciones
"	LIT	Inicio y fin de un literal
'	LITA	Inicio y fin de un literal (alternativo al anterior)
=	VI	Indicador de valor (en los atributos)
--	COM	Inicio y fin de un comentario
+	PLUS	Indicador de ocurrencia: requerido y repetible
*	REP	Indicador de ocurrencia: opcional y repetible
?	OPT	Indicador de ocurrencia: opcional
	OR	Conector O
,	SEQ	Conector de secuencia

Estructura de elementos

Marcado genérico. El rol de la DTD.

El principio de marcado genérico o lógico consiste en marcar la estructura de un documento y comprende dos fases:

- La definición del conjunto de marcas que identifican todos los elementos de un documento, junto con la definición del conjunto de reglas que expresan las relaciones entre los elementos y su estructura (esto es el rol de la DTD).
- La introducción de las etiquetas en el contenido del documento de acuerdo a las reglas establecidas en la DTD.

Varias instancias de documento pueden pertenecer a la misma clase de documento, es decir, tener la misma estructura lógica. Para describir la estructura formal de todas las instancias de documento de una determinada clase se construye la Definición de Tipo de Documento o DTD para esa determinada clase.

SGML crea los mecanismos necesarios para describir la estructura de una clase de documentos usando unas declaraciones formales de marcas. Las declaraciones de marcas establecen las marcas que se pueden usar en el documento para delimitar de manera clara y sin ambigüedades su estructura.

Las declaraciones SGML tienen en general la forma siguiente:

```
<!Palabra_clave Param Param_asociados>
```

“<!” es el delimitador de apertura de la declaración de marca y “>” es el delimitador de cierre.

Un caso especial son las declaraciones de comentarios que son declaraciones que sólo contienen comentarios:

```
<!-- Texto del comentario -->
```

Los dos guiones (-) que son los delimitadores de comentario deben ir inmediatamente después del delimitador de apertura de la declaración de marca e inmediatamente antes del delimitador de cierre, sin espacios en blanco entre ellos.

Palabra_clave puede ser:

- DOCTYPE: mediante esta declaración se define el tipo de documento, que asigna el nombre en Param al conjunto de declaraciones. Este conjunto de declaraciones puede venir justo aquí a continuación encerrado entre [], o estar en otro fichero, que se identifica en Param_asociados, o en una combinación de ambos lugares. Este conjunto de declaraciones engloba a la DTD y puede incluir cualquier otra declaración de marcas que sea necesaria.
- ELEMENT: para declarar un tipo de elemento en la estructura lógica del documento.
- ATTLIST: para asociar un tipo de elemento con un conjunto de características. Estas características se pueden aplicar a una instancia específica de ese tipo de elemento.
- ENTITY: entre otras cosas permite declarar una cadena corta de texto que signifique otra cadena más larga, para sustituir la primera cadena por la segunda cada vez que sea referenciada en una instancia de documento.
- NOTATION: se utiliza para especificar cualquier técnica especial que se necesite cuando se procesa un documento que contiene datos no SGML como por ejemplo un fichero de gráficos.
- SHORTREF: da nombre a un conjunto de asociaciones entre cadenas cortas de caracteres y marcas.
- USEMAP: para activar el conjunto de SHORTREF nombrado en Param con los elementos nombrados en Param_asociados.

Una DTD se expresa en el metalenguaje definido por el estándar SGML y en ella se define la estructura del documento en términos de los elementos que puede contener y el orden en el que estos elementos se pueden presentar. La DTD asigna un nombre a cada uno de esos elementos estructurales, el identificador genérico, por medio del cual se puede reconocer el rol del elemento. Cuando estos identificadores genéricos se colocan entre los delimitadores de marcas forman las etiquetas utilizadas para identificar el inicio y el final de cada elemento.

Una vez definida la DTD se puede marcar la instancia de documento siguiendo las reglas definidas. Se podría marcar un documento sin una DTD formal, simplemente sería necesario un conjunto de etiquetas, pero no se puede comprobar la validez de un documento sin una DTD.

Para poder validar un documento es necesario algún modo de asegurar que se ha marcado sin errores y que la estructura es coherente. Para poder cumplir estos objetivos, en la DTD

a parte de dar el nombre de los elementos que se pueden utilizar, sus posibles atributos y valores por defecto para estos atributos, se define también el contenido de cada elemento, el orden en que deben ocurrir estos elementos y cuantas ocurrencias pueden tener, el nombre de las referencias de entidad que se pueden utilizar y si se pueden omitir las etiquetas de inicio o final.

Los documentos SGML se pueden validar utilizando un software especial de análisis denominado parser SGML. Un parser SGML realiza tres tareas:

- Valida la estructura y la validez sintáctica de la propia DTD antes de analizar la instancia del documento para garantizar que no hay ambigüedades en la DTD.
- Analiza la estructura del documento.
- Comprueba si hay errores de marcado en el documento. Un documento sin errores se dice que ha sido validado.

Elementos. Declaración de tipos de elementos.

El rol de un elemento SGML depende del contexto en el que se encuentre. Tenemos dos roles principales:

- Elemento documento base. Es el primer elemento especificado en cualquier documento SGML. Este elemento debe ser formalmente declarado mediante una entrada de declaración de tipo de elemento que tenga el mismo nombre que el de la declaración de tipo de documento.
- Elementos anidados. Se pueden anidar otros elementos entre las dos etiquetas que identifican los límites del elemento documento base. El nivel de anidamientos que se pueden tener depende de la sintaxis concreta que se esté siguiendo. En la sintaxis concreta de referencia se pueden usar hasta 24 niveles de anidamiento.

Una declaración de tipo de elemento tiene la forma:

```
<!ELEMENT nombre_elem n m modelo>
```

donde:

- nombre_elem: es el nombre del tipo de elemento (su identificador genérico) que lo identifica de manera única.
- n m: son los modificadores de minimización. Son reglas para incluir u omitir las etiquetas de inicio y fin respectivamente. Un - (guión) indica que la etiqueta (de inicio o fin) es obligatoria y una o (mayúscula o minúscula) que es opcional.
- modelo: es bien una declaración formal del tipo de datos que puede contener el elemento, bien un modelo de contenido, que muestra que subelementos pueden o deben formar parte del tipo de elemento que se está declarando.

Cuando varios tipos de elementos comparten el mismo contenido y modificadores de minimización se puede sustituir el nombre del tipo de elemento por un grupo de nombres que es un conjunto de nombre de tipo de elemento conectados, generalmente por conectores OR, y encerrados entre paréntesis:

```
<!ELEMENT (nombre_elem1 | ... | nombre_elemN) n m modelo>
```

Cuando un elemento puede contener subelementos se debe definir el modelo de contenido como un grupo de modelo. Los grupos de modelo son uno o más nombres de tipos de elementos unidos mediante conectores de ordenación y encerrados entre paréntesis.

Los conectores de ordenación se utilizan para especificar un orden en un grupo y son:

- , (coma) conector de secuencia: especifica una secuencia estricta.
- & conector AND: se permite cualquier orden.
- | conector OR: puede aparecer uno y sólo uno (o exclusivo).

Los grupos de modelo utilizan además indicadores de ocurrencia. Los indicadores de ocurrencia modifican a un grupo o a un elemento individual y son:

- ?: opcional, 0 ó 1.
- +: requerido y puede repetirse, 1 ó más.
- *: opcional y puede repetirse, 0 ó más.

La ausencia de uno de estos indicadores implica que el elemento, o grupo de elementos, tiene que ocurrir una vez y sólo una.

Los indicadores de ocurrencia tienen una precedencia mayor que los conectores de ordenación.

Para indicar en los modelos de grupo un punto en el cual el elemento puede contener texto se utiliza el indicador de nombre reservado, #, seguido de PCDATA (parsed character data). #PCDATA indica que en ese lugar del modelo el elemento contiene texto que ha sido comprobado por el parser SGML para asegurar que se han identificado todas las marcas y las referencias de entidad.

Cuando un elemento no tiene subelementos, su contenido se puede declarar como uno de los siguientes tipos de contenidos declarados:

- RCDATA (replaceable character data): puede contener texto, referencias de carácter (una referencia que es reemplazada por un único carácter) o referencias de entidad que se resuelven en caracteres válidos SGML.
- CDATA (character data): contiene sólo caracteres válidos SGML.
- EMPTY: no tiene contenido.

Para los elementos con contenido declarado, la palabra clave sustituye al grupo de modelo, incluyendo los paréntesis.

Los grupos de modelo se pueden calificar añadiendo listas de excepciones. Hay dos tipos de excepciones:

- Excepciones de exclusión: identifican elementos que no se pueden utilizar mientras el elemento actual esté sin cerrar.
- Excepciones de inclusión: definen elementos que se pueden incluir en cualquier punto en el grupo de modelo.

Atributos. Declaración de listas de definición de atributos.

Un atributo es un parámetro nominado que se utiliza para calificar un elemento. Se introduce en la etiqueta de inicio del elemento.

Hay dos partes en la especificación de un atributo, su nombre y su valor, que se unen mediante un indicador de valor (=):

```
<nom_elemen nom_atrib="valor_atrib">
```

Los atributos se declaran en declaraciones de listas de definición de atributos. Su formato es:

```
<!ATTLIST elem_asoc definicion_atrb1  
.....  
definicion_atrbN>
```

donde elem_asoc es el elemento al que se asocia la lista de atributos y cada definicion_atrib es la definición de un atributo.

Cada definición de atributo consiste en un nombre de atributo, un valor declarado y un valor por defecto, separados entre ellos por carácter separador.

Una valor declarado puede ser un nombre reservado que identifica el tipo de valores que se pueden introducir o una lista de valores de atributo encerrada entre paréntesis.

Los nombres reservados que se pueden utilizar para los tipos de valores son:

CDATA	El valor consiste en caracteres válidos SGML
ENTITY	El valor puede ser cualquier nombre de entidad general declarada
ENTITIES	El valor es una lista de nombres de entidad general
ID	El valor es un identificador único para el elemento, es decir, se especifica un nombre único para el elemento cuando éste ocurre en el elemento
IDREF	El valor es un valor de referencia de ID, es decir, una referencia al nombre entrado como ID para un elemento
IDREFS	El valor es una lista de IDREF
NAME	El valor es cualquier nombre válido SGML
NAMES	El valor es una lista de NAME
NMTOKEN	Es igual que NAME pero en este caso se permite como primer carácter del nombre dígitos y cualquier otro carácter aceptado (en la sintaxis concreta de referencia dígitos, . (punto) y – (guión))
NMTOKENS	El valor es una lista de TOKEN
NUMBER	El valor es un número
NUMBERS	El valor es una lista de números
NUTOKEN	El valor es un nombre que empieza por un dígito
NUTOKENS	El valor es una lista de NUTOKEN

Un valor por defecto es o bien un valor específico o uno de los nombres reservados siguientes:

#FIXED	El valor que se especifica a continuación es un valor por defecto fijo
#REQUIRED	El valor del atributo es obligatorio
#CURRENT	Si no se especifica valor para el atributo, se utiliza el valor introducido para este atributo en el elemento anterior más próximo que comparta esta declaración de lista de definición de atributo
#IMPLIED	Si no se especifica valor, el programa puede deducir uno
#CONREF	El elemento puede contener o bien una referencia cruzada específica o un atributo cuyo valor es IDREF

Estructura de Entidades

La estructura de entidades es un modelo virtual de almacenamiento que permite dividir un documento arbitrariamente para facilitar su gestión. Es virtual porque no hay relación uno a uno entre las entidades y los objetos reales de almacenamiento.

La estructura de entidades es independiente de la estructura de elementos.

Entidades. Tipos de entidades.

SGML define una entidad como una cadena de caracteres que puede ser referenciada como una unidad.

Un documento SGML completo es una entidad llamada entidad documento SGML y que pueden contener referencias a otras entidades.

Cada entidad embebida en una entidad de documento SGML tiene dos componentes: una declaración de entidad y una o más referencias de entidad. La declaración de entidad define el nombre y el contenido de la entidad. Las referencias de entidad identifican los puntos en los que el contenido se va a introducir en el documento.

Hay dos tipos principales de entidades:

- Entidades generales: contienen datos que van a formar parte de un documento. Se pueden referenciar en la instancia de documento o en el texto de sustitución de las declaraciones de otras entidades generales.
- Entidades de parámetro: contienen caracteres que se necesitan como parte de alguna declaración SGML. Se pueden referenciar sólo en las declaraciones de marcas.

Ambas categorías se pueden subdividir en:

- Entidades externas: el texto de sustitución está definido por referencia a un identificador SYSTEM (datos que especifican el identificador del fichero, su localización y cualquier otra información que permitan localizar la entidad) o PUBLIC (un literal que identifica texto público, es decir, que es conocido más allá del contexto de un único documento o de un entorno de un sistema), es decir, el texto está almacenado en un fichero separado.
- Entidades internas: el texto de sustitución está definido dentro del prólogo.

En la sintaxis concreta de referencia las referencias de entidad general tienen la forma &nombre_entidad; o simplemente &nombre_entidad si va seguida de un espacio en blanco de un código de final de registro; y las referencias de entidad de parámetro %nombre_entidad; o %nombre_entidad.

Entidad documento SGML

Una entidad documento SGML tiene tres secciones:

- Una **declaración SGML** (opcional). Es la parte de un documento SGML donde se define el esquema de codificación utilizado en su preparación. En la declaración se especifica, entre otras cosas, el conjunto de caracteres que se está utilizando y los caracteres usados para delimitar las marcas. Para los documentos que se transmiten sin la declaración SGML, llamados documentos básicos SGML, se asume que la declaración es la proporcionada por el estándar ISO de SGML para un documento básico típico.
- Un **prólogo de documento**. Contiene la definición de las estructuras del documento (la DTD) y otra información relacionada. La indicación de qué DTD se está usando en el documento se hace por medio de una declaración de tipo de documento (declaración DOCTYPE).
- Una **instancia de documento**. Es el contenido, más las marcas, del documento.

Un ejemplo muy simple de documento SGML es el siguiente:

```

<!DOCTYPE tema
[<!ELEMENT tema - - (titulotem,indice,(seccion)+)>
<!ELEMENT seccion - - (titulosec,(parrafo|subsec)+)>
<!ELEMENT subsec - - (titulossec,(parrafo)+)>
<!ELEMENT (titulotem, indice, titulosec, titulossec, parrafo) - - (#PCDATA)>
]>
<tema>
  <titulotem> LENGUAJES DE MARCA </titulotem>
  <indice> 1. LENGUAJES DE MARCA O ETIQUETA
    2. CARACTERISTICAS Y FUNCIONALIDADES
    3. SGML
      3.1. SINTAXIS CONCRETA DE REFERENCIA
      3.2. ESTRUCTURA DE ELEMENTOS
  </indice>
  <seccion> <titulosec> 1. LENGUAJES DE MARCA O ETIQUETA </titulosec>
    <parrafo> La idea de separar .... </parrafo>
    <parrafo> El primer modo .... </parrafo>
  </seccion>
  <seccion> <titulosec> 2. CARACTERISTICAS Y FUNCIONALIDADES </titulosec>

```

```

    <parrafo> Los lenguajes de marca procedurales .... </parrafo>
  </seccion>
  <seccion> <titulosec> 3. SGML </titulosec>
    <parrafo> Los tres conceptos .... </parrafo>
    <subsec> <titulossec> 3.1. SINTAXIS CONCRETA DE REFERENCIA </titulossec>
      <parrafo> SGML define una .... </parrafo>
    </subsec>
    <subsec> <titulossec> 3.2. MARCADO GENERICO </titulossec>
      <parrafo> El principio de marcado generico .... </parrafo>
    </subsec>
  </seccion>
</tema>
```

Declaración de entidades

Las declaraciones de entidades generales tienen la forma:

```
<!ENTITY nombre_entidad texto_de_sustitución>
```

El texto de sustitución puede tener códigos de marcas, referencias a otras entidades, referencias de carácter, etc. que se interpretan cuando se añade el texto de sustitución al documento.

Hay variaciones de esta declaración básica, entre ellas las que permiten especificar:

- Entidades que contienen sólo datos de carácter que no deberían incluirse en un análisis de marcado. Esto se hace incluyendo la palabra clave CDATA entre el nombre de la entidad y su texto de sustitución. Esto significa que los caracteres dentro de la cadena de sustitución que posiblemente podrían ser interpretados como marcas serán ignorados.

- Entidades que contienen texto de sustitución específico del sistema. Para ello se incluye la palabra clave SDATA entre el nombre de la entidad y su texto de sustitución.
- Una entidad por defecto cuyo texto se usará cuando se referencia a una entidad cuyo nombre no se reconoce como una de las entidades declaradas. Esto se hace sustituyendo en la declaración nombre_entidad por la palabra reservada #DEFAULT.

Las declaraciones de entidades de parámetro tienen la forma:

```
<!ENTITY % nombre_entidad texto_de_sustitución">
```

Generalmente el texto de sustitución está formado por una serie de nombres de tipos de elemento separados por conectores de ordenación.

Las entidades de parámetro tienen que declararse antes de que sean referenciadas.

DSSSL

DSSSL (Document Style Semantics and Specification Language) definido en el estándar ISO/IEC 10179:1996 proporciona un mecanismo de propósito general para transformar documentos y para asociar instrucciones de formateado a documentos codificados utilizando SGML, tanto on-line como off-line.

DSSSL tiene dos componentes principales:

- Un lenguaje de transformación. Se utiliza para especificar transformaciones estructurales sobre ficheros fuente SGML, por ejemplo, como se fusionan dos o más documentos, como se generan índices o tablas de contenido, etc.
- Un lenguaje de estilo. Se utiliza para especificar el formato de una manera independiente de la plataforma. Para hacer posible una implementación limitada del lenguaje de estilo, dentro de él se creó CQL (Core Query Language) y CEL (Core Expression Language), que no contienen ciertas características del lenguaje de estilo designadas como opcionales.

HTML

Como ya se ha comentado HTML es una aplicación de SGML que ha sido establecido como estándar internacional (ISO/EIC 15445:2000). HTML se utiliza para la publicación de documentos en Web y para definir la estructura de los elementos y los enlaces entre ellos.

Entre las características y reglas básicas de HTML están:

- Los documentos tienen una estructura bien definida.
- Los documentos se estructuran utilizando elementos.
- La mayoría de los elementos son pares de etiquetas: una etiqueta de inicio y una de cierre. Algunos elementos tienen etiquetas de cierre optionales y otros tienen una única etiqueta de apertura.
- Las etiquetas tienen la forma <nom_elem> o <nom_elem></nom_elem>
- Las etiquetas deben estar anidadas, no cruzadas.
- Los elementos pueden tener atributos: <nom_elem nom_atributo="valor">. El valor del atributo debería ir entre comillas.
- Los nombres de elementos no son sensibles al uso de mayúsculas o minúsculas pero sí podría serlo el valor de los atributos.
- Los navegadores ignoran los elementos o atributos desconocidos.
- Los navegadores colapsan los espacios en blanco a un único espacio, esto incluye a los tabuladores, saltos de línea y retorno de carro.

Estructura de un documento HTML

De la DTD de cualquier versión de HTML se puede derivar una plantilla básica para un documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.1 Transitional//EN">
<html>
  <head>
    <tittle> El título del documento va aquí</TITLE>
    ...Información que describe el documento y proporciona otra información complementaria
    va aquí ...
  </head>
  <body>
    ...El contenido del documento, junto con las marcas va aquí...
  </body>
</html>
```

La primera línea, la declaración DOCTYPE, muestra la DTD que se sigue en el documento. Dentro de la etiqueta `<html>` se incluyen las dos secciones principales que tiene un documento HTML: la cabecera y el cuerpo. La cabecera contiene el título y otra información complementaria sobre el documento. En el cuerpo es donde va el contenido del documento con las marcas asociadas a su estructura y quizá presentación.

Todo documento debe contener una etiqueta de inicio de documento al principio, `<html>` y una etiqueta de fin de documento, `</html>` al final del documento. Contiene un único elemento HEAD y un único elemento BODY.

Las etiquetas `<head>` y `</head>` marcan el inicio y el final de la cabecera.

Las etiquetas `<body>` y `</body>` marcan el inicio y fin del cuerpo. El elemento body delimita el contenido del documento y sus atributos se utilizan para efectuar cambios que afecten al documento entero como por ejemplo establecer imágenes o colores de fondo.

En el body están incluidos los elementos de bloque estructurado como encabezamientos, párrafos, listas y tablas. Además contiene también elementos de nivel de texto. Sin embargo, aunque la definición de la presentación física de esa estructura lógica debería estar definida en otro sitio (en hojas de estilo por ejemplo), HTML no es un lenguaje de marcado lógico puro y tiene etiquetas físicas que se pueden utilizar para formatear y dar un aspecto determinado al contenido del documento.

Elementos de la cabecera del documento

Según la especificación estricta, los elementos que pueden aparecer en la cabecera son: **base, link, meta, script, style y title**.

El elemento **title** da un título al documento. Cuando se visualiza el documento en un navegador el título aparece en la barra del mismo. Su sintaxis es:
`<title>titulo_documento</title>`.

Todo documento debe tener **exactamente un elemento title** en la cabecera. El título puede contener texto plano y referencias de carácter. No se permite ninguna otra marca.

El elemento **style** embebe una hoja de estilo en el documento y la cabecera puede contener cualquier número de ellos. Tiene un **atributo obligatorio, type**, más otros opcionales. El atributo TYPE especifica el tipo de contenido del lenguaje de estilo. Por ejemplo, para CSS (Cascading Style Sheets) el valor del atributo debería ser "text/css".

Las reglas de estilo se especifican entre la etiqueta de inicio `<style>` y la etiqueta de fin `</style>` como se fuera una declaración de comentario:

```
<style type="tipo_contenido">
<!--
    REGLAS DE ESTILO
-->
</style>
```

El elemento **script** incluye un script de cliente en el documento. La cabecera puede contener cualquier número de elementos script (se permiten también en el body).

Tiene un atributo obligatorio, **type**, para especificar el tipo de contenido del lenguaje de script, por ejemplo “text/javascript”. Este atributo ha sustituido al atributo **language** en versiones anteriores, que se mantiene en esta versión como “desaprobado” (deprecated), donde se especifica el nombre del lenguaje.

Un script embebido se da como el contenido del elemento script y debe ir como si fuera una declaración de comentario:

Mediante el atributo opcional **src** se permite especificar la localización de un script externo, ya sea local o remoto. En este caso si hubiera un script embebido se ignora.

El elemento **meta** especifica pares nombre-valor para proporcionar metainformación sobre el documento, **link** especifica una relación entre el documento actual y otros recursos y **base** proporciona una dirección base para interpretar URLs relativas.

Marcas Básicas

Antes de comenzar a examinar las principales marcas decir que en HTML 4 se han añadido tres conjuntos de atributos a prácticamente todos los elementos.

Un primer conjunto de cuatro atributos, que se usan sobre todo en las hojas de estilo y lenguajes de script:

- **id** : asigna un nombre único en el documento al elemento.
- **class** : indica la clase o clases a las que pertenece el elemento (usado en hojas de estilo).
- **style** : añade información de hoja de estilo directamente en la etiqueta.
- **title** : se utiliza para dar un texto de aviso sobre un elemento o su contenido.

Se ha añadido también un conjunto de atributos manejadores de eventos para poder añadir código de scripts que se definen en múltiples elementos. Los manejadores de eventos son: **onclick**, **ondblclick**, **onkeydown**, **onkeypress**, **onkeyup**, **onmousedown**, **onmousemove**, **onmouseup**, **onmouseover** y **onmouseout**.

El último grupo está relacionado con los distintos idiomas. Los dos atributos que contempla son **dir**, que permite indicar la dirección del texto como `ltr` (de izquierda a derecha) o `rtl` (de derecha a izquierda); y **lang** que permite indicar el idioma utilizado.

Elementos de nivel de bloque

Párrafos y saltos de línea

El elemento **<p>** indica un párrafo. Generalmente el navegador incluye una línea antes y después del párrafo. Como los encabezamientos, tiene un atributo **align** para indicar el alineamiento.

La etiqueta **
** inserta un salto de línea en el documento. **
** es un elemento vacío así que no tiene etiqueta de cierre.

Divisiones y texto preformatoado

La etiqueta **<div>** se utiliza para estructurar un documento en secciones o divisiones. Tiene el atributo align que puede valer right, left o centre para alinear el texto a la derecha a la izquierda o centrado. Existe una etiqueta que es un alias para el elemento div alineado en el centro: **<centre>** .

La etiqueta **<pre>** se utiliza para indicar texto preformatoado. El texto entre las etiquetas de inicio y fin de pre conservan todos los espacios en blanco, tabuladores y saltos de línea.

Listas

HTML tiene **tres tipos de listas** :

- **Ordenadas** : se marcan con **** y generalmente se presentan como un esquema numerado.
- **No ordenadas** : se marcan con **** . Los navegadores generalmente añaden algún símbolo a cada ítem (un círculo o un cuadrado) y añaden una sangría.
- **De definición** : se marcan con **<dl>** . Las listas de definición son listas de pares término-definición, es decir, un glosario.

Las listas ordenadas y no ordenadas se pueden anidar y los elementos en la lista se definen utilizando la etiqueta **** . En las listas de definición cada término se marca con **<dt>** y cada definición con **<dd>** . No se requiere etiqueta de cierre para estos elementos.

Las listas ordenadas tienen tres atributos:

- **compact** : sugiere al navegador que intente compactar la lista.
- **type** : establece el esquema de numeración para la lista. Toma los valores "a" para letras minúsculas, "A" para mayúsculas, "i" para números romanos en minúsculas, "I" para números romanos en mayúsculas y "1" para números (es el valor por defecto).
- **start** : indica el valor para comenzar la numeración de la lista. Este valor es numérico incluso aunque se hubiera indicado en type un numeral.

Tablas

Una tabla se marca con el par de etiquetas **<table></table>** . Una tabla está compuesta de filas y las filas de celdas. Cada fila se marca con **<tr></tr>** . Las celdas pueden ser de datos, **<td></td>** , o cabeceras, **<th></th>** . El número de filas de la tabla lo determina el número de elementos tr que contenga y el número de columnas el máximo entre los números de celdas de cada fila.

Por medio de los atributos **colspan** y **rowspan** de los elementos de la tabla se pueden indicar celdas que se expandan al número de columnas o filas indicado.

Se puede especificar la anchura de la tabla mediante el atributo **width** de table y el grosor del borde con **border** . Los atributos **cellpadding** y **cellspacing** controlan el espacio entre celdas y el espacio entre el borde de la celda y su contenido respectivamente.

Elementos de nivel de texto

Los elementos de nivel de texto pueden ser físicos y lógicos. Los físicos especifican como debería presentarse el texto, los lógicos se centran en lo qué es el texto y no en su aspecto.

Elementos físicos típicos son: **** para negrita; **<i></i>** para itálica, **** para subíndice.

Elementos lógicos son: **<cite></cite>** para citas (se suele mostrar en itálica); **** para énfasis (también se muestra en itálica); **** para mayor énfasis (se suele mostrar en negrita).

Otros elementos y entidades

Encabezados

Los elementos de encabezamiento se utilizan para crear cabeceras en el contenido del documento. Hay seis niveles desde **<h1>** a **<h6>**. Su principal atributo es **align** que permite indicar los valores left (por defecto), right, center o justify para alinear el texto a la izquierda, a la derecha, centrado o justificado respectivamente.

Imágenes

Para insertar una imagen se utiliza la etiqueta **** con el URL de la imagen en el atributo **src**. Mediante el atributo **alt** se puede especificar un texto alternativo que se muestra mientras se está cargando la imagen o si esta no puede cargarse. Con los atributos **height** y **width** se dan la altura y la anchura a la imagen; y con **hspace** y **vspace** se especifica un espacio horizontal a la derecha y a la izquierda de la imagen, y un espacio vertical encima y debajo de la imagen. También se puede especificar como se alinearán el texto alrededor de la imagen. Para esto se utiliza el atributo **align** con posibles valores top, middle, bottom, left y right.

Entidades de carácter

Para poner caracteres especiales en el texto (letras acentuadas, espacios en blanco que no se colapsen, etc) se utilizan las entidades de carácter. Se referencian mediante **&codigo;** donde código es un valor numérico o una palabra que indican el carácter real que se quiere poner en el contenido. En la siguiente tabla se dan algunas de estas entidades:

Valor	Nominado	Símbolo
"	"	"
&	&	&
<	<	<
>	>	>
 	 	Espacio en blanco
©	©	©
á	á	á

Formularios

Los formularios en HTML están contenidos en un elemento **form**. Un formulario está compuesto de campos y de las marcas necesarias para estructurarlo y controlar su presentación. Cada uno de los campos se identifica de manera única dentro del formulario por el valor en el atributo name o id.

Dentro de la etiqueta **<form>** se deben identificar dos cosas: la dirección del programa que lo va a manejar, por medio del atributo **action**, y el método que se va a utilizar para

pasar los datos a ese programa, mediante el atributo **method**. En action generalmente se especifica la URL del programa. El valor de method puede ser GET o POST (métodos de HTTP).

Los principales campos del formulario incluyen: campos de texto, campos de password, campos de texto multilínea, radio buttons, check box, menús despliegables y botones.

Los campos de texto se especifican por medio de una etiqueta **<input>** (el elemento input no tiene etiqueta de cierre) y el atributo **type** igual a "text". El tamaño del campo y el número máximo de caracteres que puede contener establecen con los atributos **size** y **maxlength** respectivamente. Además se puede establecer un valor por defecto para el campo con el atributo **value**. Si el atributo type se pone como "password" tenemos un campo de password en los que no se tiene eco de los caracteres tecleados.

Los campos de texto multilínea se definen con el elemento **textarea**. El número de líneas lo determina el valor del atributo **rows** y el número de caracteres por línea el atributo **cols**. El texto por defecto es el contenido del elemento, es decir, el texto que va entre la etiqueta de apertura **<textarea>** y de cierre **</textarea>**, que debe ser texto sin incluir ninguna marca HTML. En este texto sí se conservan todos los espacios en blanco, saltos de línea y otros caracteres de control.

Para crear un menú desplegable, que permitan seleccionar una opción entre varias posibles, se utiliza la etiqueta **<select>**. Las opciones se indican con elementos **option** dentro del elemento select. El valor enviado cuando se envía el formulario es el valor entre las etiquetas de inicio y fin de option, pero si se incluye el atributo **value**, su valor será el que se transmita. El número de opciones mostradas se establece con el atributo **size** de select que por defecto vale 1. Incluyendo el atributo **multiple** en select convertimos al menú en una lista en la que se puede elegir más de una opción.

La etiqueta input también se utiliza para crear check-boxes y radio buttons, estableciendo el valor del atributo type a checkbox o radio. En el caso de los radio buttons, todos los que se quiera que estén dentro del mismo grupo, es decir, que cuando se seleccione uno del grupo se elimine la selección anterior dentro del grupo, deben tener el mismo valor en el atributo name.

Los botones se definen con el elemento input y type="button". No tienen ninguna acción predefinida. Para definir una acción se utilizan los atributos manejadores de evento mencionados con anterioridad y los lenguajes de script. Por ejemplo:

```
<input type="button" value="Pincha" onclick="alert('Mensaje');"/>
```

Dentro de un campo del formulario también es posible indicar el nombre de un fichero que se quiere cargar en el servidor. Para ello se utiliza el elemento input con type="file". Esto crea un campo de texto donde introducir el nombre del fichero y un botón a la derecha del campo generalmente etiquetado como examinar (o browse) que al pulsarlo permite examinar el sistema de archivos local para encontrar el fichero que se quiere enviar al servidor.

Una vez que se ha llenado el formulario necesitamos un medio para decir al navegador que lo envíe, este medio se proporciona de nuevo con el elemento input. Estableciendo type a submit, se crea un botón que al ser pinchado hace que el navegador envíe los datos a la dirección especificada en action. En el atributo value se establece el texto que va a aparecer en el botón y además se envía al servidor. Si el valor de type es reset se crea también un botón pero su función es limpiar el formulario y establecer los valores por defecto que se hayan podido especificar.

Direccionamiento y enlaces

En HTML la forma de definir hiperenlaces es mediante la etiqueta `<a>` (en hipertexto los extremos de los enlaces generalmente se les denomina anclas – anchors) que tiene un atributo `href` para indicar la dirección del recurso que se quiere enlazar. El contenido entre las etiquetas de inicio y fin del elemento es el hiperenlace que se activa al pinchar en él. El contenido puede ser texto o una imagen.

Un posible destino del hiperenlace es una localización dentro del propio documento. Se necesita alguna forma de poder marcar esa localización para luego referenciarla en href. También se usa la etiqueta `<a>` para definir estas localizaciones mediante un uso especial de la misma denominado establecimiento de identificador de fragmento o simplemente marcador.

Para establecer un marcador se coloca una etiqueta `<a>` en la localización y va a ser el valor del atributo name el que determine el nombre simbólico para poder referenciarla. La forma de referenciar esta localización es mediante el valor “#marcador” en el href. Por ejemplo:

```
<p><a name="inicio"></a> Esto es el inicio</p>
.....
<p><a name="fin" href="#inicio">Ir a inicio</a></p>
```

XML y sus extensiones

Como ya se ha comentado, XML (Extensible Markup Language) es una recomendación del W3C desde 1998, creada como subconjunto de SGML que aprovecha su potencia y flexibilidad eliminando parte de su complejidad. Como metalenguaje se puede utilizar para la definición de otros lenguajes de marcas. Por ejemplo se ha escrito HTML basándose en XML, dando lugar a la especificación XHTML. Otros lenguajes derivados de XML son WML (Wireless Markup Languages), SOAP (Simple Object Access Protocol) o MathML (Mathematical Markup Language).

Los objetivos del diseño de XML, citados en la especificación, eran:

- Ser directamente utilizable en Internet.
- Soportar una amplia variedad de aplicaciones.
- Compatibilidad con SGML.
- Facilidad de creación de programas que procesen documentos XML.
- Mantener el número de características opcionales al mínimo, idealmente ninguna.
- Los documentos deberían ser legibles por humanos y razonablemente claros.
- El diseño debería ser preparado rápidamente.
- Diseño formal y conciso.
- Facilidad de creación de documentos XML.
- Importancia mínima de la concisión y refinamiento de las marcas.

Cada subconjunto de SGML, XML no contempla ciertas características de SGML. Podemos citar entre las diferencias:

- Respecto a la sintaxis, en XML:
 - Los nombres pueden utilizar caracteres Unicode y no se restringen a ASCII.
 - Se permiten _ y : en los nombres.
 - Los nombres son sensitivos al uso de mayúsculas y minúsculas.
 - El delimitador de cierre de las instrucciones de procesamiento es ?>.
 - No se permiten marcas de inicio sin cerrar.

- Respecto a la declaración de elementos, atributos y entidades impone ciertas restricciones no presentes en SGML, como son:
 - Las referencias de entidad y de carácter deben terminar en ;.
 - No se permiten referencias a entidades externas de datos en el contenido.
 - En las declaraciones de elementos no se permiten grupos de nombres, ni contenido declarado como CDATA o RDATA, si se pueden utilizar exclusiones o inclusiones. Además no se puede utilizar un grupo de nombres como tipo de elemento y en los modelos de contenido no pueden utilizar el conector AND.
 - En las declaraciones de listas de definición de atributos no se permiten atributos CURRENT, no se puede utilizar un nombre de grupo como elemento asociado y los valores especificados por defecto deben ser literales.

Hoy en día XML ha alcanzado un alto grado de utilización por los beneficios que ofrece, entre ellos:

- Simplicidad: la información codificada en XML es legible visualmente y puede ser procesada sin dificultad por los ordenadores. XML es fácil de entender, implementar y usar.
- Aceptabilidad de uso para transferencia de datos: XML no es un lenguaje de programación, es una forma estándar de poner la información en un formato que pueda ser procesado e intercambiado por diversos dispositivos de hardware, SO, aplicaciones de software y la Web.
- Uniformidad y conformidad: cuando se quiere integrar dos aplicaciones, la organización y los expertos técnicos deben decidir si se integran los sistemas o se modifica la arquitectura de las aplicaciones. Si los datos de ambas aplicaciones están de acuerdo en el formato y se pueden transformar fácilmente de uno a otro, los costes de desarrollo se pueden reducir. Si este formato común se puede basar en un estándar ampliamente aceptado entonces las interfaces entre aplicaciones se hacen menos costosas.
- Separación de los datos de su visualización: sin la separación de datos la reutilización de los mismos en múltiples interfaces de usuario sería difícil.
- Extensibilidad: como se deduce de su nombre, XML fue diseñado desde el principio para permitir extensiones.

Podemos considerar dos ámbitos fundamentales de aplicación de XML:

- Compartir información: el problema principal a la hora de compartir información entre dos organizaciones cualesquiera es la interfaz entre ellas. Los beneficios de tener un formato común para compartir información entre dos organizaciones cualesquiera son obvios. Sobre XML se han construido tecnologías y estándares. Consorcios y organizaciones empresariales han desarrollado formatos y estandarizaciones XML. En diciembre de 2000 la UN/CEFACT y OASIS se unieron para iniciar un proyecto para estandarizar especificaciones XML para negocios. La iniciativa, llamada XML para negocio electrónico (Electronic Business XML, ebXML) desarrolló un marco técnico que permite utilizar XML para todo el intercambio de datos de negocio electrónico.
- Almacenamiento, transmisión e interpretación de los datos: si el almacenamiento de la información se puede adaptar a varios medios, éste se dirigirá al medio que tiene menor coste. XML está basado en texto, que obviamente es admitido por todos los medios de almacenamiento, y su coste de almacenamiento es barato comparado con los necesarios para el almacenamiento de los gráficos. Al igual que el coste de almacenamiento de estos objetos basados en texto es barato, también lo es su transmisión. Además como es comúnmente aceptado, al adherirse a estándares internacionales, puede ser fácilmente interpretado.

La recomendación XML viene acompañada de otras especificaciones como son XLS, para la creación de hojas de estilo, XLink y XPointer para la creación de enlaces con otros recursos.

Documentos XML

Documentos válidos y documentos bien formados

XML distingue entre documentos bien formados y válidos.

Un documento es un documento XML bien formado si:

- Contiene uno o más elementos.
- Tiene exactamente un elemento raíz (o elemento documento) caracterizado por que ninguna de sus partes aparece en el contenido de ningún otro elemento, es decir, debe contener un par único de etiquetas de apertura y de cierre que contengan al documento completo.
- Para el resto de elementos si su etiqueta de inicio están en el contenido de otro elemento su etiqueta de cierre debe estar en el contenido del mismo, o lo que es lo mismo, los elementos deben estar anidados apropiadamente sin permitir cruces entre estos anidamientos.
- Obedece todas las restricciones de forma (well-formedness constraints) dadas en la especificación XML.
- El contenido de todas las entidades analizables (parsed entities) está bien formado.

Una entidad analizable es aquella cuyo contenido se considera parte integrante del documento. Una entidad no analizable es un recurso cuyo contenido puede o no ser texto, y si es texto puede ser otro que no sea XML.

Algunas de las restricciones son las siguientes:

- Todos los elementos, menos los vacíos, deben llevar etiquetas de apertura y cierre.
- Los elementos vacíos, sin contenido, se marcarán con la etiqueta <elem_sin_cont/>
- Los valores de atributos van siempre entre comillas simples ('') o dobles ("")
- Los nombres son sensibles al uso de mayúsculas y minúsculas. Deben comenzar por una letra, _ o :.

Según se define en la especificación, un documento es documento XML válido si:

- Tiene asociada una declaración de tipo de documento.
- Cumple todas las restricciones expresadas en la declaración de tipo de documento.

Para comprobar la validez de los documentos, o si están bien formados, se utilizan los procesadores XML, que son módulos de software que leen el documento XML y proporcionan acceso a su contenido y estructura a la aplicación XML para la que el procesador está haciendo el análisis.

En la especificación se distingue entre procesadores con validación y sin validación. Ambos deben informar de las violaciones a las restricciones de la especificación en la entidad documento y en cualquier otra entidad analizable que puedan leer. Los procesadores con validación deben además informar de las violaciones a las restricciones expresadas en las declaraciones de la DTD y a las restricciones de validación dadas en la especificación.

Componentes de un documento XML

Un documento XML consta de un prólogo que contiene toda la información relevante del documento que no sean marcas ni contenido, y el cuerpo del documento que contiene al menos el elemento raíz.

El prólogo contiene:

- Una declaración XML. Es la sentencia que declara al documento como un documento XML.
- Una declaración de tipo de documento. Enlaza el documento con su DTD o el DTD puede estar incluido en la propia declaración o ambas cosas al mismo tiempo.
- Uno o más comentarios e instrucciones de procesamiento.

Según la especificación el prólogo debe ir antes del primer elemento del documento pero puede estar vacío, es decir, todas las componentes nombradas anteriormente son opcionales, no son necesarias para tener un documento XML bien formado.

La **declaración de documento** es una instrucción de procesamiento (las instrucciones de procesamiento permiten incluir en un documento instrucciones para aplicaciones propietarias; sus delimitadores de inicio y fin de etiqueta son <? y ?> respectivamente, tiene la forma:

```
<?xml version="version_xml" encoding="tipo_codificacion" standalone="XX" ?>
```

Donde:

- version: describe la versión de XML que se está usando. Puede ser "1.0" o "1.1".
- encoding (opcional): permite especificar la codificación de caracteres que se está usando. Por ejemplo UTF-8.
- standalone (opcional): le dice al procesador XML si el documento puede ser leído como un documento único o si es necesario buscar fuera del documento por otras reglas. Puede valer NO o YES.

La **declaración tipo de documento** contiene un nombre que debe coincidir con el nombre del elemento raíz. Opcionalmente contiene una DTD interna, referencia a una DTD externa o ambas cosas al mismo tiempo:

```
<!DOCTYPE elemen_raiz declaracion_externa [dtd_interna]>
```

Las DTD externas pueden ser públicas o privadas. Las públicas se identifican por la palabra clave PUBLIC y tienen la forma siguiente:

```
<!DOCTYPE elemen_raiz PUBLIC "nombre_DTD" "localización_DTD">
```

Las privadas se identifican por la palabra clave SYSTEM y tienen la forma:

```
<!DOCTYPE elemen_raiz SYSTEM "localización_DTD">
```

Espacios de nombres en XML

Nos podemos encontrar aplicaciones XML en las que un documento contiene elementos y atributos definidos en varios lenguajes. En varios contextos, sobre todo en la Web debido a su naturaleza distribuida, esto podría presentar un problema de colisión y reconocimiento de marcas para el software que los procesa. Se requieren entonces mecanismos que permitan tener con nombres universales cuyo alcance se extienda más allá de los documentos que los contienen. En XML este mecanismo son los espacios de nombres (XML namespaces).

Un espacio de nombres XML es un conjunto de nombres referenciados por una referencia URI, que se usan en XML como nombres de elementos y atributos.

Los nombres de elementos y atributos pueden aparecer entonces como nombres cualificados que consisten en un prefijo y una parte local separadas por : (dos puntos). El

prefijo, que tiene una correspondencia con la referencia URI, selecciona un espacio de nombres.

Un espacio de nombres se declara como un atributo. El valor del atributo, una referencia URI, es el nombre del espacio de nombres. El nombre del atributo debe comenzar por el prefijo xmlns: o xmlns. En este último caso el nombre del espacio de nombres es el del espacio de nombres por defecto que existe en el contexto del elemento donde se está incluyendo la declaración. Si el prefijo es xmlns: el nombre puede ser cualquier nombre válido XML.

Un ejemplo de declaración que asocia el prefijo edi a <http://ecommerce.org.schema> es:

```
<nom_elem xmlns:edit='http://ecommerce.org.schema'>  
.....  
</nom_elem>
```

XSL

Básicamente XSL es un lenguaje para la expresión de hojas de estilo. Una hoja de estilo XSL es un fichero que describe como mostrar un documento XML de un tipo dado. Se basa en especificaciones anteriores como DSSSL y CSS, aunque es más sofisticado.

XSL Contiene tres especificaciones:

- XSLT, XSL Transformation, un lenguaje de transformación de documentos XML. Su intención inicial era permitir operaciones de estilo más complejas que las contempladas inicialmente tales como la generación de tablas de contenido o índices, pero ahora se utiliza como un lenguaje de procesamiento de XML de propósito general. Las hojas de estilo XSLT se utilizan por ejemplo para la generación de páginas (X)HTML a partir de datos XML.
- XSL-FO, XSL Formating Objects, un vocabulario para la especificación de formateado.
- XPath, XML Path Language, un lenguaje utilizado por XSLT para referenciar o acceder a partes de un documento XML. También se usa en la recomendación de XLing, XML Linking Language.

XSL especifica como dar estilo a un documento XML utilizando XSLT para describir como el documento se transforma en otro documento XML que utiliza el vocabulario de formateado.

XSLT

Cada documento bien formado XML se puede considerar un árbol cuyos nodos son los elementos y su contenido. Para los propósitos de XSL también deben considerarse nodos los atributos, las estructuras de procesamiento y los comentarios. Básicamente un procesador XML/XSLT toma como fuente un documento XML y construye un árbol denominado árbol fuente y construye otro árbol, el árbol resultado, a partir del árbol fuente. En este proceso puede reordenar nodos, duplicarlos y añadir nuevos objetos de elementos o texto. Los nodos del árbol resultado son objetos de flujo a los que se les ha aplicado estilo. Después se interpreta el árbol resultado para obtener los resultados formateados para su presentación.

Una transformación expresada en XSLT describe las reglas para transformar un árbol fuente en uno resultado y se la llama hoja de estilo pues cuando XSLT realiza una transformación al vocabulario de formateado de XSL (XSL-FO), la transformación funciona como una hoja de estilo.

Una hoja de estilo contiene un conjunto de **reglas de plantilla**, que dan las reglas para la transformación. Una regla de plantilla tiene dos partes; un patrón, que se compara con

los nodos del árbol fuente y una plantilla de la que se crea un ejemplar para que forme parte del árbol resultado.

Los patrones son expresiones XSL, que utilizan la sintaxis de XPath, cuya evaluación va a dar un conjunto de condiciones. Los nodos que cumplen las condiciones son los que se eligen del árbol de entrada.

Las plantillas pueden contener elementos que especifican un literal para incluirlo en el árbol resultado o elementos XSLT que son instrucciones que crearán un fragmento del árbol. Cuando se crea un ejemplar de una plantilla cada instrucción es ejecutada y reemplazada por el fragmento de árbol resultado que crea. Hay elementos que seleccionan y procesan los elementos del árbol fuente que son descendientes del elemento actual (<xsl:apply-templates/>). La construcción del árbol resultado comienza encontrando la regla de plantilla para el nodo raíz y creando una instancia de su plantilla.

Un ejemplo de regla de plantilla es el siguiente:

```
<xsl:template match="/">
    <pre><xsl:apply-templates/></pre>
</xsl:template>
```

El valor de match es el patrón de la regla de plantilla (/ se corresponde con el nodo raíz) y lo que va entre la apertura y cierre de template es la plantilla.

Una hoja de estilo se representa en un documento XML por un elemento stylesheet que contienen a los elementos template que especifican las reglas de plantilla, además de otros elementos.

El elemento stylesheet debe tener un atributo versión y puede tener como atributos prefijos de extensiones de elementos, como por ejemplo los del espacio de nombres.

Por ejemplo:

```
<xsl:stylesheet version="1.0" xmlns:xls=http://www.w3.org/1999/XSL/Transform>
.....
</xsl:stylesheet>
```

Algunos de los elementos XSLT son:

- <xsl:apply-templates>: mediante esta instrucción se le dice al procesador que compare cada elemento hijo del nodo actual con el patrón de las reglas de plantilla en la hoja de estilo y si encuentra uno que se ajuste cree un ejemplar de la plantilla de la regla. Puede llevar un atributo **select** para seleccionar por medio de una expresión los nodos que se van a procesar.
- <xsl:value-of>: tiene un atributo select donde se incluye una expresión. La instrucción crea un nodo de texto en el árbol resultado con la cadena de texto que se obtiene al evaluar la expresión de select y convertir el objeto resultante a una cadena.
- <xsl:if>: el contenido del elemento es una plantilla y tiene un atributo test con una expresión que se evalúa y se convierte el resultado a booleano. Si es cierto se crea una instancia de la plantilla en el contenido, en otro caso no se hace nada.
- <xsl:element>: permite crear un elemento. Tiene un atributo nombre donde se especifica el nombre del elemento. El contenido es una plantilla para los atributos y los hijos del elemento creado.

XSL-FO

XSL-FO especifica un vocabulario para el formateado a través de los llamados objetos de formateado. Utiliza y extiende el conjunto de propiedades comunes de formateado

desarrollado conjuntamente con el grupo de trabajo CSS&FP (Cascading Style Sheet and Formating Property).

Cuando un árbol de resultado utiliza este conjunto estandarizado de objetos de formateado, un procesador XSL procesa el resultado y obtiene la salida especificada.

Algunos objetos de formateado comunes son:

- Page-sequence: una parte principal en la que el esquema de diseño (layout) de la página básica puede ser diferente del de otras páginas.
- Flow: una división en una secuencia de página como puede ser un capítulo o una sección.
- Block: como por ejemplo un título, un párrafo, etc.
- Inline: por ejemplo un cambio de fuente en un párrafo.
- Wrapper: un objeto intercambiable, puede utilizarse como block o como inline. Se utiliza sólo para manejar propiedades heredables de los objetos.
- Graphic: referencia un objeto gráfico externo.
- Table-FO: similares a los modelos de tablas HTML.
- List-FO: son listas de objetos. Por ejemplo list-item, list-block, list-item-body.

Algunas propiedades básicas de formateado son:

- Propiedades relativas a márgenes y espaciado.
- Propiedades de fuentes.
- Sangría.
- Justificación.

XPath

El principal objetivo de XPath es el poder dirigirse a partes de un documento XML. Además también se ha diseñado para que pueda utilizarse para comprobar si un nodo se ajusta a un patrón.

El constructor sintáctico básico de XPath es la expresión, que al evaluarse proporciona un objeto de uno de los siguientes tipos básicos:

- Un conjunto de nodos.
- Un valor booleano.
- Un número (de coma flotante).
- Una cadena.

Una clase de expresión es un path de localización. El resultado de evaluar la expresión es un conjunto de nodos que contiene los nodos seleccionados por el path de localización.

Hay dos tipos de paths de localización:

- Relativos: consisten en uno o más pasos de localización separados por /. Los pasos se componen de izquierda a derecha. Por ejemplo child::div/child::xxx selecciona el elemento hijo xxx del elemento hijo div del nodo de contexto.
- Absolutos: consisten en una / seguida, opcionalmente, de un path relativo.

Algunos ejemplos de paths de localización son los siguientes:

- /: selecciona el documento raíz.
- /descendant::xxx: selecciona todos los elementos xxx en el mismo documento que el nodo de contexto.
- child::xxx[position()=1]: selecciona el primer hijo xxx del nodo de contexto.
- child::*: selecciona todos los hijos del nodo de contexto.

XLink y XPointer

XLink, XML Link Language, permite insertar elementos en un documento XML que crean y describen enlaces entre recursos. Su uso más común es para la creación de hiperenlaces. XPointer especifica un modo fácil de entender y conveniente para la localización de documentos XML.

XLink proporciona funcionalidades de enlace avanzadas como:

- Enlaces multidireccionales, se dirigen a varios recursos.
- Asociar metadatos con los enlaces.
- Expresar enlaces que residen en una localización separada de los recursos enlazados.

Los enlaces se implementan por medio de atributos y no por elementos.

XPointer proporciona mejores especificaciones de localización:

- Enlaces que apuntan a un punto específico dentro de un documento, incluso aunque no exista un ID justo en ese punto.
- Granularidad fina que permite apuntar a elementos, cadenas de texto.
- Sintaxis clara para tratar las localizaciones y las relaciones en jerarquías, de forma que es legible por las personas.

Lenguajes de Script

Aunque muchos lenguajes son calificados como lenguajes de script, no hay un consenso sobre el significado exacto del término ni ha sido formalmente definido. Se llaman lenguajes de script a lenguajes como awk, sh de UNIX, Javascript, VBScript, Perl, ...

Los lenguajes de script se diseñaron para realizar tareas diferentes de las que realizan otros tipos de lenguajes de programación, como C o C++, y esto lleva a diferencias fundamentales entre los dos tipos de lenguajes.

Los lenguajes de programación tradicionales se diseñaron para construir estructuras de datos y algoritmos empezando desde el principio, desde elementos de computación como las palabras de memoria. Para manejar esta complejidad suelen ser fuertemente tipados.

Frente a esto, los lenguajes de script **se diseñaron para unir componentes**. Asumen la existencia de un conjunto de potentes componentes y se centran generalmente en poner juntas estas componentes. Además suelen ser **débilmente tipados**, el tipo se determina en tiempo de ejecución, para simplificar la conexión entre componentes.

Otra diferencia entre los dos grupos de lenguajes, los de script frente a los que no reciben esta denominación, es que los lenguajes de script son generalmente **interpretados** en vez de compilados. Los lenguajes interpretados eliminan los tiempos de compilación durante el desarrollo y hacen a las aplicaciones más flexibles al permitir que se genere código fácilmente en tiempo de ejecución.

Estas características producen, generalmente, código que es menos eficiente. Sin embargo, el rendimiento no suele ser un aspecto fundamental en un lenguaje de script. Los lenguajes de script se dirigen fundamentalmente a áreas donde la flexibilidad y un desarrollo rápido son mucho más importantes que el puro rendimiento. Las aplicaciones en lenguajes de script suelen ser más pequeñas que las aplicaciones en otros lenguajes de programación y el rendimiento de una aplicación de scripts está dominado por el rendimiento de sus componentes, que generalmente están implementadas en algún otro tipo de lenguaje de programación.

Los dos grupos de lenguajes son complementarios, y las principales plataformas de computación han proporcionado tanto lenguajes de programación como de scripts desde los años 60. Mientras que los lenguajes de programación que podríamos denominar de

sistemas o tradicionales son buenos para producir implementaciones eficientes de funcionalidades críticas con respecto al tiempo, los lenguajes de script son buenos para poner juntas componentes de distinta funcionalidad.

Sin embargo la existencia de máquinas cada vez más rápidas, la existencia de mejores lenguajes de scripts, la importancia creciente de las interfaces gráficas de usuario y de las arquitecturas de componentes, y sobre todo el crecimiento de Internet, han incrementado enormemente la aplicabilidad de los lenguajes de script.

Podemos citar entre las características típicas de los lenguajes de script, aunque no siempre las tienen todas:

- Interpretados.
- Débilmente tipados.
- Asignación dinámica de memoria con liberación automática de la misma.
- Potentes capacidades para la manipulación de cadenas (expresiones regulares).
- Procedurales y con extensiones de orientación a objetos.
- No es necesario que el usuario del lenguaje defina clases o métodos.
- Utilizan un método de invocación de métodos muy simple, no es necesario especificar los paths de los objetos para acceder a los métodos.

Entre los lenguajes de script más comunes podemos citar Javascript, Perl, VBScript, ...

Aplicación de los lenguajes de script a internet

Los lenguajes de script se han utilizado en Internet para un aspecto fundamental, añadir interactividad y dinamismo a las páginas web.

El método más antiguo para añadir esta interactividad es mediante programas que utilizan la interfaz CGI (Common Gateway Interface). CGI es un protocolo estándar que especifica como pasar información desde una página web, a través de un servidor web, a un programa y devolver información desde el programa a la página en el formato apropiado. Los lenguajes de script, sobre todo Perl, son los más comúnmente utilizados para escribir estos programas CGI.

Pero CGI presenta ciertos problemas de ineficiencia. Por un lado los programas CGI se ejecutan fuera del servidor web y además están diseñados para manejar sólo una petición y terminar su ejecución después que han devuelto sus resultados al servidor.

Hoy en día se utiliza sistemas que toman la forma de componentes que se apoyan en APIs específicas del servidor para interactuar directamente con el proceso del servidor. Conectándose como un subprocesso al servidor Web se evita mucha de la sobrecarga de los programas CGI convencionales.

La generación de contenido dinámico del lado del servidor requiere que el servidor procese las peticiones en tiempo de ejecución para dar una respuesta apropiada a la petición, así que se necesitan instrucciones para ejecutar este procesamiento y por tanto alguna forma de programarlas.

Otra forma de añadir interactividad y dinamismo es mediante scripts del lado del cliente, fundamentalmente usando Javascript. El código de script se introduce directamente o se referencia en el documento HTML y se ejecuta cuando se carga la página o como respuesta a otro tipo de eventos.

Un ejemplo de uso de scripts en Web es la validación de formularios (uno de los motivos originales para el desarrollo de Javascript). La validación de formularios es el proceso de comprobar la validez de los datos introducidos por el usuario antes de enviarlos al servidor.

Con la aparición de la generación 4.x de los navegadores se introdujo el concepto DHTML, HTML dinámico, que describe la capacidad de manipular dinámicamente los elementos de una página a través de la interacción:

- HTML: la versión 4 introdujo dos cosas importantes, CSS y DOM.
- CSS: con CSS se tiene un modelo de estilo y layout para documentos HTML.
- DOM (Document Object Model): proporciona un modelo del contenido para documentos HTML.
- Javascript (y VBScript): permite codificar scripts que controlen los elementos HTML.

Análisis y gestión de riesgos de los sistemas de información. La metodología MAGERIT: método, elementos, técnicas.

Planes de Seguridad, Contingencias y Recuperación

La Información y los sistemas e instalaciones que la sustentan, son en la actualidad un activo fundamental para toda organización. Este activo está expuesto a una serie de amenazas más o menos destructivas (intrusiones, virus, fuego, desastres naturales, etc.), lo que hace necesario un instrumento capaz de gestionar su seguridad. La seguridad de la información se caracteriza por la preservación de la:

- **Confidencialidad** : Asegurando que la información es accesible sólo a aquellos que tienen autorización para ello.
- **Integridad** : Salvaguardando la completitud y precisión de la información y de los métodos de procesamiento.
- **Disponibilidad** : Asegurando que los usuarios autorizados tienen acceso a la información en el momento que este acceso es requerido.

La gestión de la seguridad de los sistemas de información incluye el análisis de los requerimientos de seguridad, el establecimiento de un plan para satisfacer estos requerimientos, la implementación de este plan y el mantenimiento y administración de la seguridad implementada. Este plan es el **Plan de Seguridad** .

El plan de seguridad se diseña para asegurar que se ha establecido de forma correcta el contexto y el alcance de la gestión de la seguridad, que se ha identificado y evaluado todos los riesgos de seguridad y que se han desarrollado estrategias apropiadas para el tratamiento de estos riesgos.

Pero incluso en las organizaciones bien preparadas, con buenos programas de seguridad, suceden incidentes, emergencias o desastres que afectan a la continuidad del servicio. Debido al avance de las tecnologías de la información, las organizaciones actuales dependen en gran medida de ellas. Con la aparición del e-business muchas organizaciones no pueden sobrevivir sin un modo de operación 24x7x365 (24 horas al día, 7 días a la semana, 365 días al año).

El Plan de Seguridad debe incluir tanto una estrategia proactiva como una reactiva:

- La **estrategia proactiva** es una estrategia de prevención que comprende el conjunto de pasos que ayudan a minimizar los puntos vulnerables de los activos (recursos de los sistemas de información).
- La **estrategia reactiva** es una estrategia de corrección que comprende aquellas medidas orientadas a reducir los daños ocasionados una vez que se ha producido un ataque o incidente.

Así que son necesarios planes extensos y rigurosos para alcanzar el estado de continuidad de las operaciones, en el que los sistemas críticos y las telecomunicaciones estén continuamente disponibles. La descripción de las acciones y decisiones necesarias para asegurar la continuidad de las operaciones de la Organización en el caso de un incidente, una emergencia o de un desastre se recogen en el **Plan de Contingencias**.

Mediante el Plan de Contingencias se diseñan y formulan los modos de actuación alternativos que van a permitir la operación de la Organización en condiciones adversas, la recuperación de sus funciones de la forma más rápida posible, y que van a proporcionar continuidad a los procesos críticos del negocio.

Como parte del Plan de Contingencias está el Plan de Recuperación de Desastres, en el que se desarrollan todas las normas de actuación para reiniciar las actividades de proceso, bien sea en el propio centro de proceso de datos o en un lugar alternativo (Centro de Respaldo) en el caso de producirse un desastre.

Gestión de la Seguridad. El Plan de Seguridad

La gestión de la seguridad es un proceso que comprende varias etapas. Comienza estableciendo los objetivos y las estrategias de seguridad y desarrollando una política corporativa de seguridad de tecnologías de la información. Como parte de esta política de seguridad corporativa está la creación de una estructura organizativa apropiada para asegurar que los objetivos definidos se pueden alcanzar.

La siguiente etapa sería el análisis y gestión de los riesgos, que incluye las siguientes actividades:

- Realización de un análisis y evaluación de riesgos.
- Selección de las salvaguardas para los sistemas de información basada en el resultado del análisis y evaluación de riesgos.
- Formulación de políticas de seguridad de sistemas de información.
- Elaboración del plan de seguridad de sistemas de información, basado en las políticas de seguridad anteriores, para implementar las salvaguardas.

Las dos últimas etapas son la implementación del plan de seguridad y su mantenimiento.

El propósito del Plan de Seguridad es dar una visión general de los requerimientos de seguridad del sistema de información y describir los controles necesarios para alcanzar esos requerimientos. Además delimita las responsabilidades y el comportamiento esperado de toda persona que acceda al sistema. Mediante el Plan de Seguridad se establecen las salvaguardas necesarias para proteger los recursos de tecnologías de la información, ya sean salvaguardas de gestión, técnicas u operativas.

Objetivos, Estrategias y Política de Seguridad Corporativa

El primer paso en el proceso de la gestión de la seguridad debería ser determinar cuál es el nivel de riesgo aceptable para la Organización y de aquí cuál es el nivel apropiado de seguridad. El nivel de seguridad apropiado está determinado por los objetivos o requerimientos de seguridad que la Organización necesita cumplir. Es esencial que una Organización identifique sus requerimientos de seguridad. Hay tres fuentes principales para esta identificación:

- El análisis y evaluación de riesgos: mediante éste se identifican las amenazas sobre los activos y las vulnerabilidades de éstos. Además se evalúa la posibilidad de ocurrencia de estas amenazas y se estima su impacto.
- El marco legal: en nuestro caso se deben tener en cuenta los requisitos establecidos por la Ley Orgánica 15/1999 de Protección de datos de carácter personal y el Real

Decreto 994/1999, por el que se aprueba el Reglamento de medidas de seguridad de los ficheros automatizados que contengan datos de carácter personal. Para las Organizaciones de la Administración Pública se tiene además el Real Decreto 263/1996 por el que se regula la utilización de técnicas electrónicas, informáticas y telemáticas por la Administración General del Estado, modificado por el Real Decreto 209/2003 por el que se regulan los registros y las notificaciones telemáticas, así como la utilización de medios telemáticos para la sustitución de la aportación de certificados por los ciudadanos.

- El conjunto de requerimientos, objetivos y principios particulares para el procesamiento de la información que la Organización ha desarrollado para dar soporte a sus operaciones.

Dependiendo de los objetivos de seguridad se eligen las estrategias para alcanzar estos objetivos. Cuatro son las estrategias básicas usadas para controlar el riesgo que resulta de las vulnerabilidades:

- Evitar el riesgo, mediante la aplicación de salvaguardas que eliminan o reducen los riesgos que permanecen para la vulnerabilidad considerada.
- Transferir el riesgo a otras áreas o a entidades externas, por ejemplo mediante la contratación de un seguro.
- Mitigar el riesgo, reduciendo los impactos en caso de que se explote la vulnerabilidad. La forma de mitigar el riesgo es mediante la Planificación de la Contingencia.
- Aceptar el riesgo sin control o mitigación, bien sea por su poca posibilidad de ocurrencia o por su bajo impacto.

Es importante el compromiso de la dirección y los altos niveles de gestión de la Organización con la seguridad de los sistemas de información. Este compromiso debería resultar en una política corporativa de seguridad de las tecnologías de la información formalmente consensuada y documentada. Esta política debería cubrir, como mínimo, los siguientes aspectos:

- Los objetivos de seguridad.
- Los aspectos organizativos de la seguridad, desde el punto de vista de las infraestructuras y de la asignación de roles y responsabilidades.
- Integración de la seguridad dentro del ciclo de desarrollo de sistemas.
- Directivas y procedimientos generales de seguridad.
- Definición de clases para la clasificación de la información y de los sistemas.
- Políticas de salvaguarda.
- Planificación de la contingencia.
- Consideraciones legales.

Análisis y Evaluación de Riesgos

Como ya se ha comentado el análisis y evaluación de riesgos (AER) es una de las fuentes fundamentales para la identificación de los requerimientos de seguridad. Los resultados de esta evaluación van a servir de guía para determinar las acciones de gestión adecuadas y las prioridades en el tratamiento de los riesgos de seguridad. Además van a ayudar a seleccionar e implementar las salvaguardas para protegerse de esos riesgos.

Mediante el AER se identifican las amenazas que soportan los activos del Sistema de Información (o los relacionados con él) y se determina cual es la vulnerabilidad de esos activos frente a esas amenazas. Con esto se estima el grado perjuicio (impacto) que pueden tener estas amenazas y vulnerabilidades y se calcula el riesgo que se corre.

Existen dos aproximaciones al Análisis de Riesgos, una cuantitativa y otra cualitativa. La primera se basa en dos parámetros fundamentales: la probabilidad de que ocurra un suceso y una estimación de las pérdidas en caso de que así sea. El producto de estos dos

factores es el denominado Coste Anual Estimado (EAC, Estimated Annual Cost) y aunque teóricamente es posible calcularlo para cualquier evento y tomar decisiones basadas en su valor, en la práctica la dificultad de la estimación o la inexactitud en el cálculo de parámetros hace que esta aproximación sea la menos usada.

La segunda aproximación, la cualitativa, es mucho más sencilla e intuitiva que la anterior, y es la más extendida hoy en día. Se basa en una simple estimación de pérdidas potenciales (ya no interviene el cálculo de probabilidades exactas). Para ello se interrelacionan cuatro elementos: las amenazas, las vulnerabilidades, los impactos asociados a las amenazas y las salvaguardas que se toman para minimizar los impactos o las vulnerabilidades. Con estos elementos se obtiene un indicador cualitativo del riesgo asociado a un activo, visto como la posibilidad de que una amenaza se materialice en un activo y produzca un impacto.

Los riesgos se pueden clasificar entonces según su nivel en:

- No aceptable: el riesgo pone en peligro los objetivos fijados en el proyecto de seguridad, causa un daño o pérdida irreparable. Se deben tomar acciones inmediatas para reducir su nivel.
- Crítico: el riesgo puede afectar los objetivos del proyecto o causar un daño o pérdida importante. El riesgo debe mitigarse.
- Mayor: el riesgo produce un impacto significativo en el proyecto (afecta a costos, calidad u a otras áreas). Se deben ejecutar acciones que reduzcan el impacto.
- Menor: No causa problemas significativos. Se debe monitorear de una manera regular para asegurar que no sube de nivel.

Selección de Salvaguardas

Se deben identificar y seleccionar las salvaguardas apropiadas para reducir el riesgo evaluado hasta niveles aceptables. Para seleccionar las salvaguardas se tienen que considerar: el resultado del análisis y evaluación de riesgos, las salvaguardas ya existentes o planificadas y las restricciones de varios tipos que puedan existir.

Mediante el análisis y evaluación de riesgos se han detectado las vulnerabilidades de los sistemas con amenazas asociadas que pueden explotar estas vulnerabilidades y cuál es el impacto que pueden causar. Así que los resultados de esta evaluación nos van a indicar dónde es necesario una protección adicional y de qué tipo.

Se deben tener también en cuenta las salvaguardas ya existentes o planificadas para evitar trabajo o costes innecesarios o para determinar si las nuevas salvaguardas son compatibles con aquellas. Además se tienen que examinar en términos de comparaciones de coste con vistas a eliminarlas (o no implementarlas) o mejorarlas en caso de que no sean suficientemente efectivas.

Hay varios tipos de restricciones que deben ser tenidos en cuenta a la hora de seleccionar las salvaguardas, entre ellos:

- Restricciones de tiempo: por ejemplo hay que considerar si el tiempo que va a tomar implementar la salvaguarda es un periodo de tiempo aceptable para dejar el sistema expuesto a un determinado riesgo.
- Restricciones financieras: el coste de la salvaguarda no debe ser mayor que el valor de los activos que va a proteger.
- Restricciones técnicas: como la compatibilidad de software o hardware.

Para la selección de salvaguardas se debe tener en cuenta que exista un equilibrio entre las salvaguardas de gestión, las salvaguardas operacionales y las salvaguardas técnicas.

Las salvaguardas de gestión, que se centran en la gestión de la seguridad del sistema, incluyen:

- Reglas de comportamiento: delimitan las responsabilidades y el comportamiento esperado de toda persona que acceda al sistema.
- Revisión de las salvaguardas: la seguridad de los sistemas puede degradarse con el tiempo, debido a los cambios tecnológicos, a la propia evolución del sistema o cambios en los procedimientos. Se deben realizar revisiones periódicas para garantizar que los controles funcionan de una manera efectiva y proporcionan los niveles adecuados de protección.
- Planificación de la seguridad durante todo el ciclo de vida del sistema: se deben tener en cuenta los requisitos de seguridad en todas las fases del ciclo de vida del sistema.

Las salvaguardas operacionales, salvaguardas no técnicas que proporcionan seguridad física, de personal y administrativa, incluyen:

- Controles de seguridad física y del entorno: entre ellas se encuentran el establecimiento de áreas seguras y equipamiento de seguridad.
- Controles de seguridad del personal: cubren comprobaciones en el reclutamiento del personal, especialmente del personal de confianza, programas de educación, entrenamiento y de compromiso con la seguridad.
- Controles del mantenimiento de aplicaciones de software: se utilizan para controlar la instalación o la actualización de aplicaciones de software. Entre ellos están el control de versiones, la gestión del cambio o la gestión de la configuración.
- Controles de validación e integridad de datos: se utilizan para proteger a los datos de alteraciones accidentales o malintencionadas, o para evitar su destrucción.
- Planificación de la contingencia que se puede considerar también como una salvaguarda.

Las salvaguardas técnicas, enfocadas a los controles que ejecutan los sistemas de computación, incluyen:

- Identificación y autenticación.
- Controles lógicos de acceso (autorización y controles de acceso).
- Registros de auditoría.

Política de Seguridad de Sistemas de Información

La política de seguridad del sistema de información contiene los detalles de las salvaguardas necesarias y el por qué son necesarias. Muchos sistemas, que introducen consideraciones especiales no contempladas en otros sistemas de la Organización, necesitan sus propias políticas de seguridad. Estas políticas deben ser compatibles con la política corporativa de seguridad de tecnologías de la información.

La política debe estar basada en los resultados del análisis de riesgos y debe contener las salvaguardas necesarias para alcanzar el nivel de seguridad apropiado para el sistema considerado. En concreto la política debe contener:

- Una descripción del sistema y de sus componentes.
- Una descripción de los servicios y funciones de negocio cubiertos por el sistema.
- Una identificación de los objetivos de seguridad para el sistema.
- El grado de dependencia de la Organización respecto al sistema.
- Los activos del sistema que se quieren proteger y una valoración de estos activos.
- Las vulnerabilidades del sistema y las amenazas a las que se enfrenta, incluyendo la relación entre los activos y las amenazas y la probabilidad de que éstas se materialicen.

- Los riesgos de seguridad del sistema resultantes de la combinación de la probabilidad de que se materialicen las amenazas, la facilidad de explotación de las vulnerabilidades y el impacto sobre el negocio.
- La lista de salvaguardas identificadas para proteger el sistema.
- El coste estimado de seguridad.

Elaboración e Implementación del Plan de Seguridad

El Plan de Seguridad del Sistema debe estar basado en la política de seguridad del sistema y describe las acciones que se van a tomar para mantener el nivel apropiado de seguridad y para implementar las salvaguardas requeridas para el sistema. Debe asegurar que las salvaguardas se implementan en el tiempo planificado, de acuerdo con las prioridades que se derivan del análisis de riesgos.

Se debe incluir:

- Los objetivos de seguridad en términos de confidencialidad, integridad, disponibilidad, autenticidad y fiabilidad.
- Una evaluación del riesgo residual aceptado después de implementar las salvaguardas identificadas.
- Una lista de las salvaguardas seleccionadas que se van a implementar, así como una lista de las salvaguardas ya existentes.
- Una identificación y definición de las acciones que se van a tomar, con sus prioridades, para implementar las salvaguardas.
- Una estimación de los recursos necesarios para la implementación de las salvaguardas y de los costes asociados.
- Un plan detallado de implementación del Plan.

Después de haber elaborado el plan de seguridad es necesario implementarlo. A la vez que se implementan las salvaguardas se debe desarrollar un programa para conseguir que todo el personal de tecnología de la información y los usuarios finales tengan un conocimiento suficiente de los sistemas y que entiendes porqué las salvaguardas son necesarias y como utilizarlas correctamente.

Durante la implementación deben tenerse en cuenta los siguientes puntos:

- El coste de las salvaguardas permanece dentro del rango aprobado.
- Las salvaguardas se implementan correctamente según lo establecido en el Plan de Seguridad.

Mantenimiento del Plan de Seguridad

El mantenimiento del Plan comprende la revisión y el análisis de las salvaguardas implementadas, la gestión del cambio y monitorización de la seguridad.

El objetivo de la revisión y análisis de las salvaguardas es comprobar que las salvaguardas establecidas en el plan están implementadas y que se usan correctamente.

Mediante la gestión del cambio se va a determinar que impacto tiene en la seguridad del sistema los cambios que se realizan, o está planificado realizar, en el mismo.

Por monitorización del Plan se entienden las actividades de comprobación de si el sistema, su entorno y sus usuarios mantienen el nivel de seguridad establecido en el Plan de Seguridad. La monitorización es un modo de detectar cambios relevantes en la seguridad.

Planificación de la Contingencia. El Plan de Contingencias

En el Plan de Contingencias se detalla la forma en que debe reaccionar la Organización y qué acciones emprender para garantizar la recuperación y continuidad y de las operaciones críticas, frente a determinados eventos (ya sean accidentales o malintencionados) que pueden provocar una interrupción del servicio. En el plan se tiene que incluir quien es la autoridad para iniciar las actividades descritas, cuales son las operaciones a las que se debe garantizar la continuidad y como restablecer dichas operaciones.

Un plan de Contingencias debería cubrir la ocurrencia de eventos como:

- Fallos del hardware.
- Fallos del software.
- Interrupción del suministro eléctrico o de los servicios de telecomunicaciones.
- Errores humanos o sabotajes.
- Ataques de software malicioso.
- Fuego.
- Desastres naturales.

La planificación de la contingencia es un proceso que, siguiendo la Guía para la Planificación de la Contingencia del NIST (National Institute of Standards and Technology), comprende los siguientes pasos:

1. Desarrollar una política de planificación de la contingencia.
2. Realizar un Análisis de Impacto en el Negocio.
3. Desarrollar estrategias de recuperación.
4. Desarrollar el Plan de Contingencias.
5. Realizar pruebas y entrenamientos.
6. Revisión y mantenimiento del plan.

Vamos a desarrollar a continuación cada una de estas etapas.

Desarrollo de Políticas de Planificación de Contingencias

Para que un plan de contingencias sea efectivo debe basarse en políticas claramente definidas que aseguren que todo el personal entiende de una forma completa los requerimientos de planificación de contingencias de la Organización. Estas políticas definen los objetivos y las responsabilidades, y establecen un marco para la planificación de la contingencia de los sistemas de información.

Análisis de Impacto en el Negocio

Una vez definidas las políticas, la primera fase en el desarrollo de un proceso de planificación de contingencias es realizar un análisis de impacto en el negocio (BIA, Business Impact Analysis). Un BIA es una investigación y una evaluación del impacto que pueden provocar en un sistema los diversos ataques potenciales a los que se enfrenta. Con él se establece el alcance del Plan y se determina cuales son los recursos críticos y los tiempos de recuperación aceptables.

El BIA arranca donde termina el Análisis y Gestión de Riesgos (AGR). Se asume que los controles que se establecieron como resultado del AGR han fallado, han conseguido ser sobrepasados o simplemente no han sido efectivos, y que el ataque ha tenido éxito.

Los tres pasos para realizar un BIA son: identificación de los recursos críticos del sistema, identificación de los impactos y de los tiempos asumibles de las interrupciones, y establecimiento de las prioridades de recuperación.

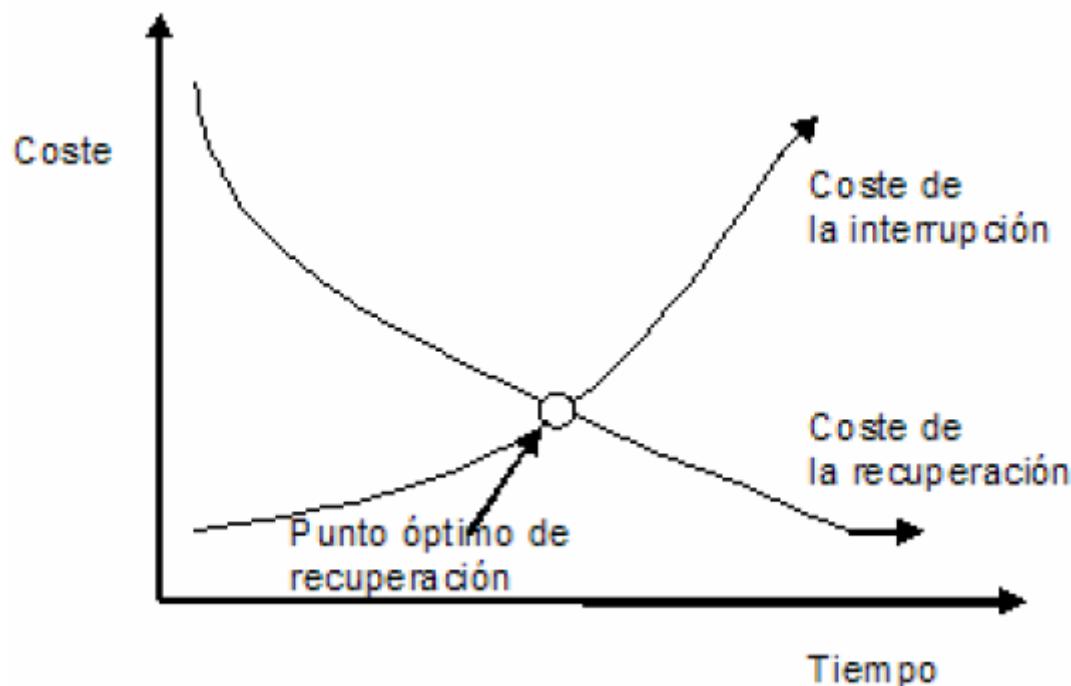
Los sistemas de información pueden ser extremadamente complejos y realizar múltiples servicios a través de numerosos componentes, procesos e interfaces. El primer paso del BIA es evaluar el sistema para determinar cuales son las funciones críticas y cuales son los recursos necesarios para realizarlas.

En el siguiente paso se analizan los recursos críticos identificados en la etapa anterior y se determina el impacto de un daño o de una interrupción en cada uno de dichos recursos. Los efectos de estos daños o interrupciones se deben estudiar teniendo en cuenta:

- Su duración en el tiempo, para poder identificar cual es el tiempo máximo que se puede asumir que un recurso estará fuera de servicio antes de que se produzca la negación de una función esencial.
- Los recursos relacionados y los sistemas dependientes, para identificar cualquier efecto en cascada que pudiera ocurrir.

Se debe determinar cual es el punto óptimo de recuperación del sistema teniendo en cuenta que tiene que haber un equilibrio entre el coste de la no operatividad del sistema y el coste de los recursos necesarios para su restauración.

Considerando la gráfica del coste de interrupción y recuperación como función del tiempo, como se muestra en la siguiente figura, el punto óptimo de recuperación es el punto en el que se cortan ambas funciones.



El último paso, establecimiento de las prioridades de recuperación, se basa en los resultados de la etapa anterior para priorizar las estrategias de recuperación que se establecerán durante la activación del Plan de Contingencias. Mediante esta priorización se pueden tomar decisiones bien fundadas respecto a la asignación de recursos y gastos de recuperación, ahorrando esfuerzos, tiempo y dinero.

Desarrollo de Estrategias de Recuperación

Las estrategias de recuperación proporcionan los medios para restaurar el sistema de una manera rápida y efectiva después de una interrupción de las operaciones.

Una estrategia básica e imprescindible la constituyen los respaldos o copias de seguridad (backups). Para conseguir una política de respaldo efectiva es necesario determinar aspectos como:

- Qué ficheros o sistemas de ficheros respaldar y dónde se encuentran.
- El tipo de backup que se va a realizar. Tenemos tres tipos de backups: totales, incrementales y diferenciales. En el total se copia el conjunto total de los datos que queremos respaldar. En el incremental se salvan sólo aquellos ficheros que han cambiado desde la última copia de seguridad, sea esta del tipo que sea. En el diferencial se salvan sólo aquellos ficheros que han cambiado desde el último backup total. Cada uno de los tipos tiene sus ventajas y sus inconvenientes y son más o menos efectivos dependiendo de las circunstancias. La mayoría de las estrategias de respaldo utilizan una combinación de backups totales con backups incrementales o diferenciales.
- La frecuencia con la que se va a realizar. La frecuencia de realización de los backups depende de dos factores claves: la tasa de cambio de los datos y de la sensibilidad de los mismos para la organización. Un caso típico es realizar backups completos una vez a la semana y backup incrementales o diferenciales diariamente.
- Cuál es el tiempo de retención de las copias. Se determina el periodo de tiempo durante el cual se tienen que mantener archivadas las copias de seguridad antiguas.
- Qué medios de almacenamiento se van a utilizar, cintas, discos magnéticos, discos ópticos, y cuál va a ser la estrategia de rotación de los conjuntos de medios que se utilicen. Un esquema de rotación típico es la rotación “abuelo-padre-hijo” (GFS, Grandfather-Father-Son). Este esquema utiliza conjuntos de medios diarios (hijo), semanales (padre) y mensuales (abuelo).

Una implementación típica del esquema GFS es considerar un periodo de rotación de tres meses y 12 cintas:

- Hijo: se asignan 4 cintas para el backup incremental diario [1-4]
- Padre: se asignan 5 cintas para el backup total semanal [5-9]
- Abuelo: se asignan 3 cintas para el backup total mensual [10-12]

	Lunes	Martes	Miércoles	Jueves	Viernes
Cinta	1	2	3	4	5
Cinta	1	2	3	4	6
Cinta	1	2	3	4	7
Cinta	1	2	3	4	8
Cinta	1	2	10-12 ¹	-	9 ²

Notas:

1. Sobre el último día hábil del mes
2. Para cuando hay un quinto viernes en el mes

Aunque interrupciones muy graves, con efectos a largo plazo, pueden ser que ocurran raramente se deben considerar en el plan de contingencia. El plan debe incluir entonces una estrategia de recuperación de las operaciones del sistema en instalaciones alternativas para operar en ellas durante un periodo de tiempo. Se pueden clasificar estos lugares alternativos en:

- “Cold site”: está localizado en el exterior de la Organización con toda la infraestructura necesaria preparada para la instalación en el caso de que ocurra un desastre, pero es la organización que utiliza el sitio frío la que debe instalar el equipamiento necesario. En un centro frío llevaría más de un día reiniciar las operaciones después del desastre.
- “Mutual Backup”: dos Organizaciones con una configuración similar de los sistemas se ponen de acuerdo para servir una a la otra como lugar de respaldo.

- “Hot site”: un lugar con instalaciones de telecomunicaciones, software y hardware compatibles con el lugar de producción. En un centro frío con ese equipamiento tomaría menos de un día el reiniciar las operaciones después del desastre.
- “Remote Journaling”: se refiere a transmisiones on-line de las transacciones de datos para salvaguardar de manera periódica el sistema de forma que se minimice la pérdida de datos y el tiempo de recuperación.
- “Mirrores Site”: un lugar equipado con un sistema idéntico al de producción con instalaciones de mirroring. Los datos se duplican en el sistema de salvaguarda de manera inmediata de forma que la recuperación es transparente a los usuarios.

Desarrollo del Plan de Contingencias

Podemos considerar cuatro componentes primarios en un Plan de Contingencias:

- Información de soporte.
- Fase de notificación/activación.
- Fase de recuperación.
- Fase de restauración.
- Apéndices del plan.

En la primera sección, información de soporte, generalmente se incluye cual es el propósito y los objetivos del Plan, y se define su alcance, identificando los sistemas y las localizaciones que van a estar cubiertos por el plan. Se describe además la estructura de los equipos de contingencias, incluyendo su jerarquía y mecanismos de coordinación, con sus roles y responsabilidades en una situación de contingencia.

En la sección fase de notificación/activación se definen las acciones que se van a tomar una vez que se detecta una situación de emergencia o una interrupción del servicio. Estas acciones incluyen la notificación al personal de recuperación, la evaluación de daños y activación del Plan. Los procedimientos de notificación, que describen los métodos para comunicarse con el equipo de recuperación, tienen que estar documentados en el Plan. La activación del Plan se realiza sólo cuando el análisis de daños indica que se cumplen uno o más de los criterios de activación que se han establecido en la declaración de la política de la planificación de la contingencia (fase 1 de la planificación). Una vez que se ha caracterizado el daño y establecido que se debe activar el Plan de Contingencias, el coordinador del plan puede seleccionar una estrategia de recuperación adecuada y se realiza la notificación al equipo de recuperación asociada con ella.

En la sección fase de recuperación se describen las actividades que se centran en las medidas para ejecutar las capacidades de procesamiento de una manera temporal y reparar el daño y recuperar las capacidades operativas del sistema original (o de uno nuevo si este fuera irrecuperable). Los procedimientos de recuperación deben reflejar las prioridades identificadas en el Análisis de Impacto en el Negocio y estar documentados en un formato secuencial, de paso a paso, de forma que los distintos componentes del sistema se puedan restaurar de una manera lógica.

En la fase de restauración las operaciones de recuperación han terminado y se transfieren de nuevo las operaciones normales al sistema original (o uno nuevo en el caso de que fuera irrecuperable). Las siguientes operaciones tienen lugar en esta fase:

- Asegurarse que las infraestructuras adecuadas, tales como energía eléctrica o telecomunicaciones están operativas.
- Instalar el hardware y el software del sistema.
- Establecer las conexiones y las interfaces con la red y con los sistemas externos.
- Probar las operaciones del sistema para probar su completa funcionalidad.
- Copiar los datos del sistema de contingencia y cargarlos en el sistema restaurado.
- Dar por terminadas las operaciones de contingencias.

Pruebas y Entrenamiento

Las pruebas del plan es un elemento crítico que va a permitir detectar sus deficiencias y evaluar la efectividad de cada procedimiento de recuperación y del plan como un todo.

Entre las pruebas a realizar están:

- Recuperación a partir de los backups.
- Restauración de las operaciones normales.
- Rendimiento del sistema al utilizar equipamientos alternativos.
- Los procedimientos de notificación de contingencias.

Para complementar las pruebas el personal debe entrenarse en la ejecución de los procedimientos bajo su responsabilidad con el objetivo de que sean capaces de realizarlos sin necesidad de la ayuda del documento real del plan de contingencias.

Mantenimiento del Plan

Los sistemas de información sufren cambios debidos a las necesidades cambiantes de la Organización, a los cambios tecnológicos o a nuevas políticas internas o externas. Para ser efectivo el Plan de Contingencias debe mantenerse para que se ajuste a los nuevos requerimientos, procedimientos o políticas. El plan debería revisarse con una base periódica, por ejemplo una vez al año, y siempre que se efectúen cambios significativos en cualquier elemento del plan.

Los cambios en el Plan deberían documentarse por medio de un registro de cambios en el que se especifique la fecha del cambio, la parte a la que afecta y un comentario sobre el mismo.

Plan de Recuperación de Desastres

El Plan de Recuperación de Desastres se refiere al plan enfocado a las tecnologías de la información diseñado para restaurar la operativa del sistema, de las aplicaciones o de las instalaciones de computación, generalmente en un lugar alternativo después de una emergencia o incidente grave.

Es la parte del Plan de Contingencias que debe incluir todas las acciones necesarias que van a permitir a la Organización operar durante el periodo de tiempo del desastre. Se definen en él los recursos, las acciones y los datos requeridos para reiniciar los procesos críticos del negocio, ya sea en el CPD o en Centro de Respaldo.

En el Plan de Recuperación de Desastres se incluyen:

- Los criterios usados para identificar cuando debería activarse el plan, quien es el responsable de la activación del plan y como se va a comunicar ésta.
- Tareas y responsabilidades específicas de los miembros del equipo de recuperación de desastres.
- Procedimientos para evaluar los daños provocados por el desastre.
- Uso de soportes de procesamiento alternativo.
- Acciones de mantenimiento o sustitución de equipos dañados.
- Uso de copias de seguridad.
- Para la recuperación en un Centro de Respaldo se debe especificar además la política de traslados y vuelta al centro inicial.

Una guía de las secciones que se incluyen en un Plan de Recuperación de Desastres es la siguiente:

- Propósito: incluye la declaración de los objetivos del plan.
- Alcance: identifica los sistemas críticos y las localizaciones específicas a las que se les aplica el plan.
- Bases: identifica las bases en las que se apoya el plan.
- Equipo: identifica al equipo que lidera las actividades del plan, así como los otros miembros involucrados en el proceso.
- Notificación: establece los métodos formales de comunicación para alertar al equipo de recuperación de la ocurrencia del incidente o desastre.
- Evaluación de daños: describe los procesos de análisis del alcance del daño.
- Activación: describe las situaciones y procesos que van a iniciar las actividades de recuperación.
- Operaciones de recuperación: describe todos los pasos para la recuperación de los sistemas y aplicaciones críticos, bien el propio centro o en Centro de Respaldo (cuando sea aplicable). Aquí se debería incluir la información sobre como recuperar los datos basándose en las copias de seguridad.
- Retorno a las operaciones normales: los procedimientos que debe seguir el equipo para la completa recuperación de todos los datos y la vuelta al procesamiento normal de todas las operaciones del negocio.

Políticas de Salvaguarda

Las políticas de salvaguarda son las que van a determinar como se van a seleccionar los controles y medidas de protección para alcanzar los requerimientos de seguridad que se hayan especificado para los sistemas.

Basándose en el resultado de revisiones de alto nivel en cuanto a seguridad, se pueden considerar tres opciones a la hora de hacer esta selección:

- Aproximación de línea base.
- Basada en un análisis de riesgos detallado.
- Basada en una combinación de las dos opciones anteriores.

Con la primera opción se selecciona un conjunto de salvaguardas para alcanzar un nivel básico de protección para todos los sistemas. Para establecer este conjunto de salvaguardas un buen punto de partida es considerar las que se basan en requerimientos legislativos o las consideradas como “buenas prácticas comunes” para la seguridad de la información.

Salvaguardas consideradas esenciales para una organización desde el punto de vista legislativo incluyen:

- Las salvaguardas para la protección y privacidad de los datos de carácter personal.
- Las salvaguardas para la protección de los derechos de la propiedad intelectual.
- Las salvaguardas para los registros de la organización que deben mantenerse por requerimientos normativos o legales.

Salvaguardas consideradas como “buenas prácticas” incluyen:

- Documentación de las políticas de seguridad.
- Asignación de responsabilidades de seguridad de la información.
- Educación y entrenamiento al personal en materia de seguridad.
- Información de incidentes de seguridad.
- Gestión de la contingencia.

El método MAGERIT de gestión de la seguridad, identifica 10 salvaguardas mínimas de seguridad en su guía de aproximación:

- Documentación de políticas de seguridad de la información. Como ya se ha visto en la sección del plan de seguridad, la organización debe publicar las normas de seguridad de los sistemas de información de una manera sencilla y comprensible para todos los empleados de la organización.
- Asignación de funciones y responsabilidades de seguridad. Se deben indicar de forma explícita las funciones y las responsabilidades de seguridad en la Organización. Para cada activo sujeto a medidas de seguridad deben identificarse los roles que actúan sobre ellos, y las personas que los ejercen.
- Responsabilidades del usuario en el acceso al sistema. De manera especial los usuarios deben conocer su responsabilidad en cuanto a los medios que se ponen a su cargo y a los controles de acceso para que éstos puedan mantener su eficacia.
- Educación y formación en la seguridad de la información. Para garantizar que los usuarios están preparados para seguir los procedimientos de seguridad de la organización y se minimizan posibles riesgos, deben recibir formación sobre los requerimientos de seguridad y sobre el uso correcto de los sistemas de información.
- Comportamiento ante incidentes de seguridad. Se deben establecer procedimientos, que deben ser conocidos por todos los empleados de la organización, para realizar y remitir informes sobre los diferentes tipos de incidentes, amenazas, vulnerabilidades o mal funcionamiento del hardware o del software.
- Controles físicos de seguridad. Deben existir controles que sólo permitan el acceso al personal autorizado a cada área de seguridad. Las autorizaciones de acceso se deben dar para propósitos específicos y controlados.
- Gestión de la seguridad del Equipamiento. Los aspectos que se consideran en la gestión de la seguridad del equipamiento son:
 - Instalación y protección del equipamiento. El equipamiento debe estar físicamente protegido para salvaguardarlo de pérdidas o daños.
 - Suministro eléctrico. Se debe garantizar un suministro eléctrico adecuado que cumpla las especificaciones de los fabricantes de equipos. Para los equipos que soporten operaciones críticas se recomienda tener un sistema de alimentación ininterrumpida.
 - Mantenimiento de equipos. Los equipos deben mantenerse en buen estado siguiendo las recomendaciones y especificaciones de los proveedores. Deben registrarse documentalmente los fallos identificados o las sospechas de fallos.
 - Movimientos de activos fuera de la organización. Cuando sea necesario sacar de la Organización equipos, datos o software se hará siempre con la correspondiente autorización.
- Cumplimiento de las obligaciones y restricciones jurídicas vigentes.
- Protección, transporte y destrucción de la Información. La Organización debe controlar de manera especial los intercambios de información con otras organizaciones para prevenir la modificación, pérdida o mal uso de la información en tránsito. Cuando se transmita información mediante medios de comunicación inalámbricos que utilicen radiofrecuencias de información debe ir cifrada.
- Gestión Externa de servicios. Ante una propuesta para cambiar a una gestión externa de servicios se deben incluir los controles de seguridad apropiados para las nuevas implicaciones de seguridad. En concreto deben estudiarse:
 - Identificación de las aplicaciones que se van a retener en la Organización por su criticidad o sensibilidad.
 - Implicaciones para los planes de contingencia de la Organización.
 - Especificación de las nuevas Normas de seguridad y del proceso de control de su cumplimiento.

Entre las ventajas de esta primera opción, aproximación de línea base, está que no se necesitan recursos para hacer un análisis de riesgos detallados y que el tiempo y esfuerzo

utilizado en la selección de salvaguardas es reducido. Además se pueden usar las mismas salvaguardas básicas para varios sistemas sin grandes esfuerzos.

Entre sus desventajas está que si este nivel de línea base es demasiado alto, podría proporcionar una seguridad demasiado restrictiva para algunos sistemas o demasiado cara, y si el nivel es demasiado bajo, para algunos sistemas podría no proporcionar una seguridad suficiente.

La segunda opción es realizar un análisis de riesgos detallado para cada sistema de la Organización. Basándose en los resultados obtenidos se identifican y se seleccionan las salvaguardas que reduzcan los riesgos identificados hasta los niveles considerados como aceptables.

La principal ventaja de esta opción es que se identifica un nivel de seguridad apropiado para las necesidades de cada sistema, pero tiene como desventaja que para obtener resultados viables se necesita una cantidad de tiempo y esfuerzo considerable, además de tener que ser llevada a cabo por equipos expertos.

La tercera opción es considerar una combinación de las dos anteriores. Se analizan los sistemas de la Organización utilizando un análisis de alto nivel para identificar aquellos sistemas que sustentan operaciones críticas para el negocio o que tienen un alto riesgo. Se clasifican entonces los sistemas en dos categorías: los que necesitan un análisis de riesgos detallado para alcanzar la protección adecuada, y los que es suficiente que tengan una protección basada en la aproximación de línea base.

Entre las ventajas de esta opción están que el esfuerzo y el dinero se aplicarán primero sobre los sistemas que tienen un riesgo mayor y donde es más beneficioso. Como desventaja está que si ese primer análisis de alto nivel produce resultados poco exactos, algunos sistemas que necesitan un análisis de riesgo detallado podrían pasarse por alto.

El Método MAGERIT de Gestión de la Seguridad

MAGERIT, Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información de las Administraciones Públicas, es un método formal para investigar los riesgos que afectan a los sistemas de información y para recomendar las medidas de control de esos riesgos.

Es una metodología de carácter público (su utilización no requiere autorización previa) elaborada por el Comité Técnico de Seguridad de los Sistemas de Información y Tratamiento Automatizado de Datos Personales, SSITAD, del Consejo Superior de Informática.

Los objetivos de MAGERIT son:

- Estudiar los riesgos que soporta un sistema de información y su entorno asociado (análisis de los riesgos).
- Recomendar las medidas que deben tomarse para prevenir, impedir, reducir o controlar esos riesgos (gestión de los riesgos).

Es decir MAGERIT se ocupa del Análisis y de la Gestión de Riesgos, punto de arranque del ciclo de gestión de seguridad.

MAGERIT se presenta como un conjunto de guías más unas herramientas de apoyo.

Las guías son:

- Guía de Aproximación. En ella se presentan los conceptos básicos de seguridad de los sistemas de información y una introducción al núcleo de MAGERIT, formado por la Guía de Procedimientos y la Guía de Técnicas.
- Guía de Procedimientos. Su finalidad es servir de marco de referencia para la identificación y la valoración de los activos y de las amenazas, para la evaluación de las vulnerabilidades y de los impactos, y como preparación para la toma de decisiones sobre políticas de salvaguardas a partir del cálculo de riesgos.
- Guía de Técnicas. Dan las bases para comprender y seleccionar las técnicas más apropiadas para los procedimientos de análisis y gestión de riesgos.
- Guía para Responsables del Dominio protegible. Determina cual es la participación de los directivos en la realización del Análisis y Gestión de Riesgos de los sistemas de información relacionados con su dominio funcional dentro de la organización.
- Guía para Desarrolladores de aplicaciones. Está diseñada para que pueda utilizarse como documento de referencia por los desarrolladores en la preparación de los mecanismos de seguridad adecuados durante el desarrollo de nuevas aplicaciones.
- Arquitectura de la información y especificaciones de la interfaz para el intercambio de datos. La arquitectura de la información permite a los técnicos informáticos estructurar la información que se ha de cargar en todo producto informatizado que sea semejante o esté relacionado con las herramientas de apoyo de MAGERIT. La interfaz para el intercambio de datos posibilita a los usuarios de MAGERIT establecer la comunicación con otras aplicaciones, facilitando la incorporación de otros productos a las herramientas de apoyo de MAGERIT o a la inversa.
- Referencia de normas legales y técnicas. Sirve como referencia a la documentación normativa en cuestiones de seguridad (como referencia a fecha 31/12/1996, pues no se han publicado actualizaciones de la guía que recojan la nueva normativa) y como instrumento para homogeneizar el intercambio de información entre las distintas entidades y personas implicadas en la seguridad de los sistemas de información.

Las herramientas de apoyo, que vienen junto con sus correspondientes guías de uso, son:

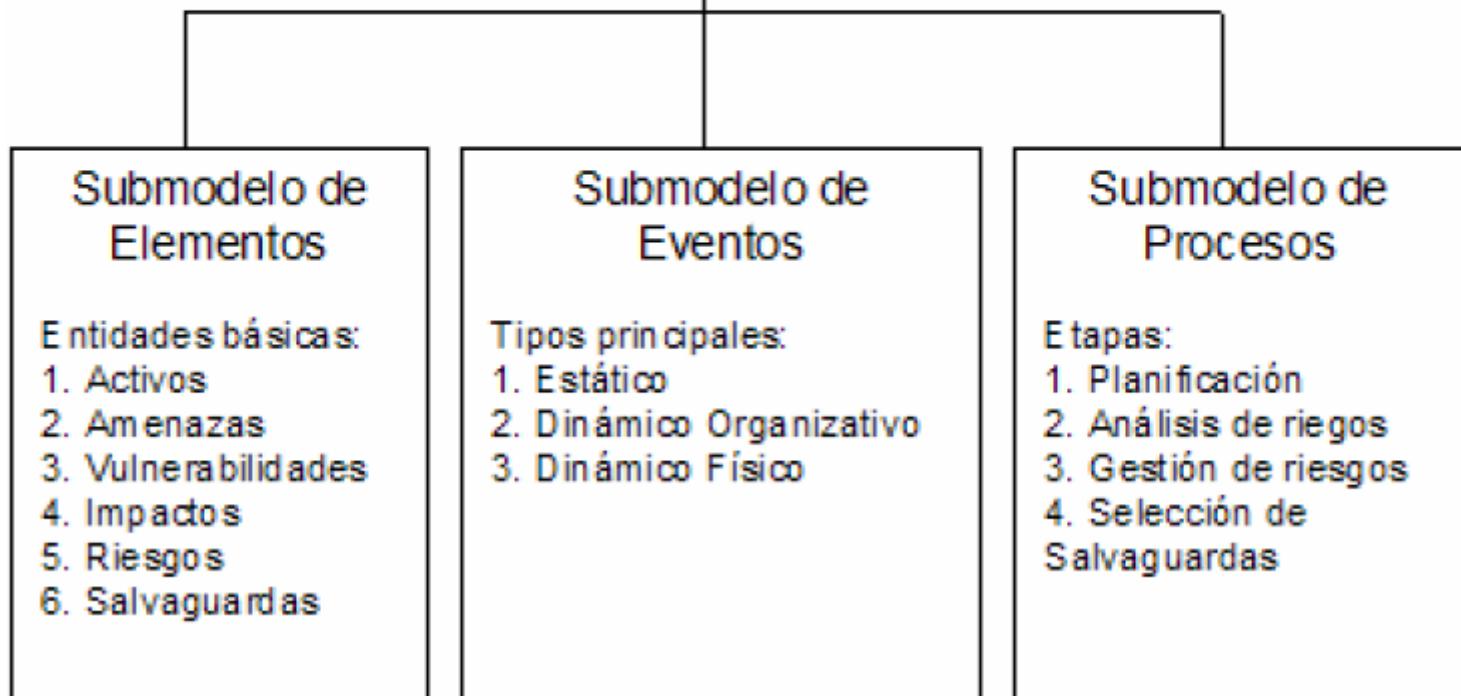
- Herramienta 1 Introductoria. Permite una primera aproximación al Análisis y Gestión de Riesgos. Es un apoyo para la identificación de:
 - Riesgos menores, a los que sencillamente se les puede aplicar medidas básicas de seguridad de una forma global.
 - Riesgos mayores, a cada uno de los cuales hay que aplicar un Análisis y Gestión de Riesgos más detallado.
- Herramienta 2 Avanzada. Permite realizar un Análisis y Gestión de Riesgos detallado con los que afrontar proyectos de complejidad media o alta en materia de seguridad. Utiliza técnicas algorítmicas, técnicas matriciales y técnicas de lógica difusa.

Existe además otra herramienta, “chinchón version 1.3”, para analizar cuantitativamente el riesgo de un sistema de información siguiendo la metodología de MAGERIT.

El modelo de MAGERIT está formado por tres submodelos:

- Submodelo de elementos: son los componentes del modelo.
- Submodelo de eventos: relaciona los componentes entre sí y con el tiempo.
- Submodelo de procesos: es la descripción funcional del proyecto de seguridad a construir.

MODELO DE MAGERIT



Vamos a ver a continuación cada uno de estos submodelos.

Submodelo de Elementos

Se basa en seis entidades que están relacionadas entre sí y cada una de ellas dotada de ciertos atributos.

Estas seis entidades son: Activos, Amenazas, Vulnerabilidades, Impactos, Riesgos y Salvaguardas.

Activos

Los activos se definen en la Guía de Procedimientos como “*los recursos del sistema de información o relacionados con éste, necesarios para que la organización funcione correctamente y alcance los objetivos propuestos por su dirección*”.

MAGERIT considera los activos clasificados en cinco grandes categorías:

- Entorno del Sistema de Información. Se incluye aquí cualquier elemento del entorno necesario para las siguientes categorías de activos. Por ejemplo, equipamientos y suministros, personal, instalación física.
- Sistema de Información. Incluye hardware, software, comunicaciones.
- Información. Datos, meta-information, soportes.
- Funcionalidades de la organización. Se refiere a las funcionalidades que justifican la existencia de los Sistemas de Información y les dan finalidad. Incluye:
 - Objetivos y misión de la organización.
 - Bienes y servicios producidos.
 - Personal usuario/destinatario de los bienes o servicios producidos.
- Otros activos. Cualquier activo no incluido en las categorías anteriores. Por ejemplo, credibilidad, conocimiento acumulado, independencia de criterio, etc.

Las tres primeras categorías constituyen el dominio estricto de todo proyecto de Seguridad de Sistemas de Información.

El fallo de un Activo de una categoría puede provocar cadenas de fallos: fallos en Activos del Entorno (1) generaría otros fallos en el Sistema de Información (2), que repercutirían en la Información (3) y así sucesivamente.

Cada activo tiene como atributos esenciales dos indicadores sobre dos tipos de valoraciones:

- La valoración intrínseca del activo: este atributo permite determinar para qué sirve un activo. En él se basa la clasificación dada anteriormente de los tipos de activos. Se centra en el aspecto cualitativo del activo como valor de uso.
- La valoración del estado de seguridad del activo: cada activo se caracteriza por su estado de seguridad. Este estado es el resultado de la estimación de cuatro subestados definidos por MAGERIT: Autenticación, Confidencialidad, Integridad y Disponibilidad.

Amenazas

Se definen en MAGERIT como “*los eventos que pueden desencadenar un incidente en la organización, produciendo daños materiales o pérdidas inmateriales en sus activos*”.

Se ven las amenazas bajo un enfoque dinámico, como un evento de tipo potencial que si se materializa en una agresión hace pasar al activo de un estado de seguridad antes del evento a otro posterior.

Se consideran cuatro tipos de amenazas:

- Grupo A de Accidentes: son amenazas no humanas.
- Grupo E de Errores: amenazas humanas pero involuntarias.
- Grupo P de Amenazas Intencionales Presenciales: amenazas humanas intencionales que necesitan presencia física.
- Grupo T de Amenazas Intencionales de Origen Remoto: amenazas humanas intencionales que proceden de un origen remoto.

Vulnerabilidades

Se define la vulnerabilidad como la “*potencialidad o posibilidad de ocurrencia de la materialización de una Amenaza sobre un Activo*”.

La Vulnerabilidad tiene dos aspectos:

- el estático, realiza una función de mediación entre un Activo, como objeto de cambio del estado de seguridad, y una Amenaza como acción.
- el dinámico, es el mecanismo que convierte a la Amenaza en una agresión materializada.

Mediante el término Vulnerabilidad se cubren dos acepciones distintas:

- Vulnerabilidad intrínseca del Activo respecto al tipo de Amenaza, en la que sólo se tienen en cuenta el Activo y la Amenaza.
- Vulnerabilidad efectiva, en la que se tienen también en cuenta las Salvaguardas aplicadas en cada momento al Activo.

Siempre que sea factible se mide la Vulnerabilidad por la posibilidad de la materialización de la Amenaza en agresión o por su frecuencia histórica.

Impactos

Según MAGERIT “ *el Impacto en un Activo es la consecuencia sobre éste de la materialización de una Amenaza* ”.

El impacto mide la diferencia entre el estado de seguridad del activo antes de la materialización de la amenaza y el estado posterior a dicha materialización. Por tanto una simple disfunción en un activo no constituye necesariamente un Impacto, a no ser que se produzca un perjuicio apreciable como cambio del estado de seguridad.

MAGERIT considera tres grandes grupos de Impactos:

- Cuantitativos: si representan pérdidas que se pueden cuantificar monetariamente, ya sea directa o indirectamente.
- Cualitativos con pérdidas funcionales: si son reductoras directamente de los subestados de seguridad (Autenticación, Confidencialidad, Integridad y Disponibilidad).
- Cualitativos con pérdidas orgánicas: como la pérdida de fondos patrimoniales intangibles, el daño a personas, la responsabilidad penal por incumplimiento de obligaciones legales, etc.

Riesgos

El riesgo es “ *la posibilidad de que se produzca un Impacto determinado en un Activo, en un Dominio o en toda la Organización* ”.

MAGERIT distingue entre dos tipos de riesgo:

- Riesgo intrínseco: el que se define o calcula antes de aplicar Salvaguardas.
- Riesgo residual: el que queda después de aplicar las salvaguardas dispuestas.

Además define el umbral de riesgo como el valor establecido como base para decidir, por comparación, si el riesgo calculado es asumible.

Salvaguardas

Se define en MAGERIT la Función o Servicio de Salvaguarda como “ *la acción que reduce el riesgo* ”.

Se hace una distinción entre la Función o Servicio de Salvaguarda que acabamos de definir y el mecanismo de salvaguarda que es el “ *dispositivo físico o lógico capaz de reducir el riesgo* ”.

Una Función o Servicio de Salvaguarda representa por tanto una actuación para reducir un riesgo, actuación que se concreta en un mecanismo de salvaguarda.

Tipificando las salvaguardas según su forma de actuación, distingue entre:

- Preventivas: actúan sobre la Vulnerabilidad y reducen la posibilidad de materialización de la Amenaza, es decir, actúan con anterioridad a la agresión. Entre ellas están la formación, información y concienciación del personal, la disuasión, la detección preventiva, etc.
- Curativas (o restablecedoras): actúan sobre el Impacto reduciendo su gravedad, es decir, actúan con posterioridad a la agresión. Entre ellas están la corrección, la recuperación, etc.

Submodelo de Eventos

Para presentar este modelo de forma intuitiva se ha utilizado la metáfora de la “ciudad amurallada”:

- Las salvaguardas son la muralla y las vulnerabilidades las brechas de la muralla.
- Las amenazas son el enemigo exterior y los activos quedan dentro del recinto amurallado.

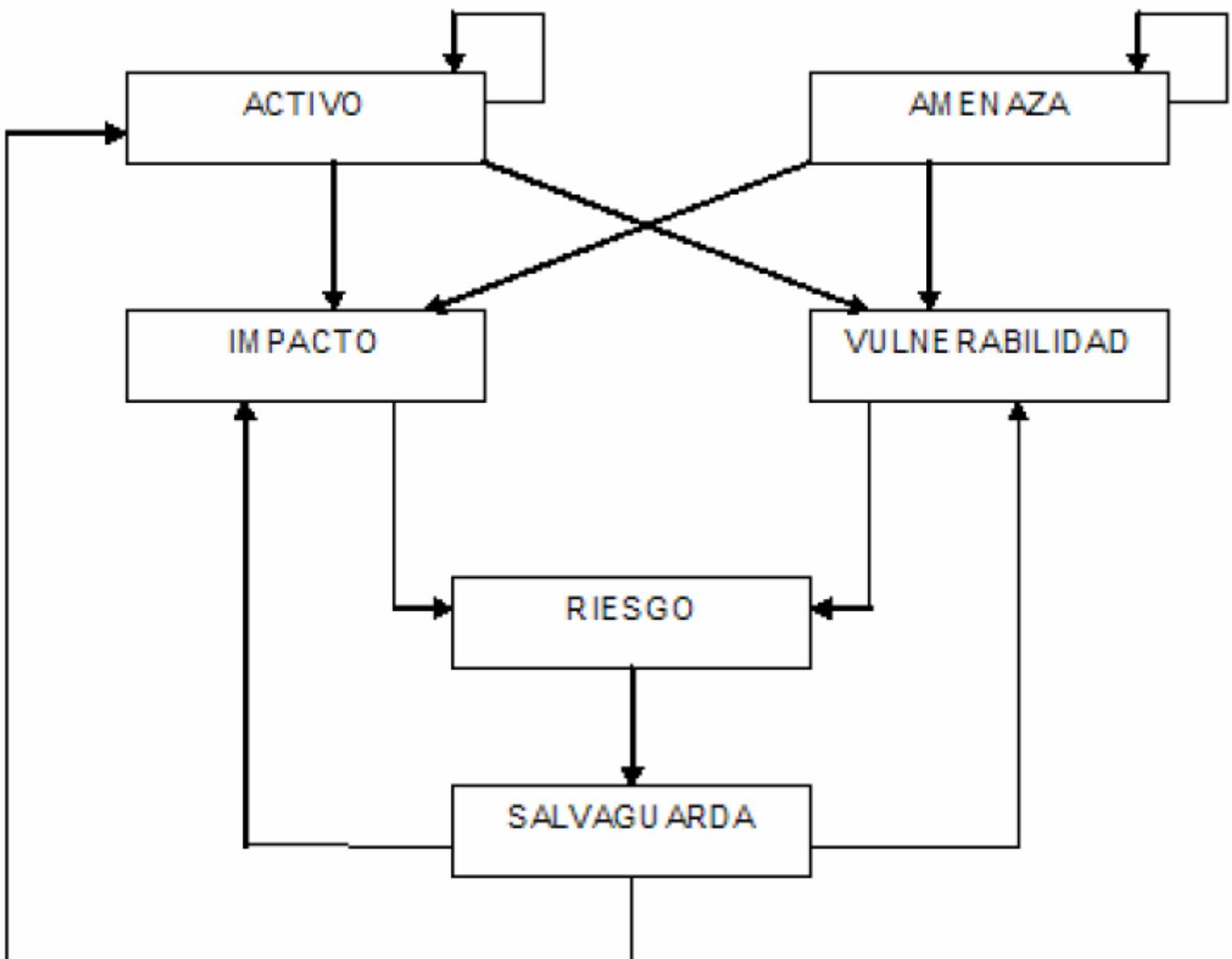
Esta visión estática de la seguridad se ha comprobado que es ineficiente en los Sistemas de Información actuales que son cada vez más “ciudades abiertas” sujetas a nuevos tipos de amenazas intencionales.

MAGERIT ofrece un submodelo de eventos de la metodología con tres puntos de vista:

- Vista estática relacional. En esta vista se ponen de manifiesto las relaciones existentes entre las entidades vistas en el Submodelo de Elementos. Es necesaria para establecer el Modelo Lógico de Datos que requerirán toda herramienta de apoyo de MAGERIT.
- Vista dinámica de tipo organizativo. Recoge de forma detallada la forma que tienen de interactuar entre sí las entidades del Submodelo de Elementos. Es necesaria para la construcción de herramientas de apoyo de MAGERIT, para determinar las técnicas de cálculo de riesgos y de selección de salvaguardas, y como soporte al Submodelo Lógico de Procesos.
- Vista dinámica de tipo físico. Recoge otra forma de funcionamiento de la interactuación entre los Elementos, con un nivel intermedio de detalle entre las dos vistas anteriores. Esta visión no es imprescindible para la comprensión de la metodología y será necesaria sólo en determinadas situaciones. Es útil para dar apoyo a ciertas técnicas de cálculo de riesgos y de selección de salvaguardas como las técnicas de simulación.

Vista Estática Relacional

La vista estática relacional recoge las relaciones directas entre las entidades del Submodelo de Elementos. El esquema básico se muestra en la siguiente figura:



El Activo se relaciona directamente con las entidades:

- Activo: un activo puede ser componente de otro activo o depender de él.
- Vulnerabilidad: un activo puede tener vulnerabilidades respecto a diversas amenazas.
- Impacto: un activo se puede ver afectado por varios impactos procedentes de diversas amenazas.
- Salvaguarda: una salvaguarda puede proteger un activo.

La Amenaza (si se materializa) se relaciona con:

- Amenaza: una amenaza puede ser componente de otra amenaza o depender de ella.
- Vulnerabilidad: una amenaza tiene que aprovecharse de una vulnerabilidad para afectar a un activo.
- Impacto: una amenaza puede causar un impacto sobre un activo.

El Impacto se relaciona con:

- Activo: un impacto se produce sobre un activo.
- Amenaza: un impacto puede causarlo la materialización de una amenaza.
- Salvaguarda: un impacto puede ser influenciado por un servicio de salvaguarda.

La Vulnerabilidad se relaciona con:

- Activo: una vulnerabilidad se produce sobre un activo.
- Amenaza: una amenaza puede sacar provecho de una vulnerabilidad.
- Salvaguarda: una vulnerabilidad puede ser influenciada por un servicio de salvaguarda.

- Riesgo: una vulnerabilidad contribuye al riesgo.

El riesgo se relaciona con:

- Vulnerabilidad: un riesgo se refiere a la vulnerabilidad de un activo.
- Impacto: un riesgo se refiere al impacto propiciado por la vulnerabilidad de un activo.
- Salvaguarda: un riesgo puede ser influenciado por una salvaguarda.

La salvaguarda se relaciona con:

- Activo: una salvaguarda protege un activo.
- Impacto: una salvaguarda puede afectar al impacto de una amenaza.
- Vulnerabilidad: una salvaguarda puede afectar a la vulnerabilidad una amenaza.
- Riesgo: una salvaguarda influye en un riesgo.

Vista Dinámica Organizativa

En esta vista se parte primero de una distinción entre el Dominio de Información que MAGERIT está ayudando a analizar dentro de la Organización, y el Entorno del Dominio delimitado. Dentro de este último se diferencia entre una parte interna a la Organización y otra externa.

El dominio está formado por activos y tiene como sujeto principal al Responsable del Dominio o de los Activos protegibles, que es quien establece su valor y las necesidades de seguridad mediante objetivos y decisiones.

El Dominio, definido en cada momento por su estado de seguridad, está sujeto a cambios en su estado por las acciones de las amenazas materializables y también por las salvaguardas. A partir de esta materialización desencadenante, el Submodelo de Eventos funciona dinámicamente como un escenario en el que se establecen acciones para obtener la seguridad.

Se distinguen dos subescenarios:

- Subescenario de análisis de las amenazas (o de ataque): es el análisis de riesgos. Parte de un ataque y analiza las consecuencias que tiene sobre los activos del dominio.
- Subescenario de análisis y gestión de salvaguardas (o de defensa): es la gestión de riesgos. Describe para cada ataque la gestión de las funciones de salvaguarda que sean apropiadas.

El subescenario de análisis de las amenazas considera que cada Activo del dominio, si la Amenaza se ha materializado en agresión aprovechando la Vulnerabilidad del activo con respecto a esa Amenaza, la Vulnerabilidad y su Impacto sobre el Activo determinan el **Riesgo calculado**. Su contribución es asimétrica, es decir, se considera que el Impacto influye más en el nivel de Riesgo que la Vulnerabilidad. La determinación del Riesgo calculado cierra este subescenario de ataque y abre el subescenario de defensa.

El subescenario de análisis y gestión de las salvaguardas se caracteriza por la toma de decisiones. Se comienza por comparar el riesgo calculado con el umbral de riesgo asumible. Tras la implantación de las salvaguardas se recalcula el riesgo calculado obteniéndose el llamado **Riesgo residual**. Si este es inferior al umbral de riesgo asumible, se considera que el subescenario de defensa ha cumplido su objetivo y se puede pasar a otras fases de Gestión de Seguridad de los Sistemas de Información. Si el riesgo residual es superior al umbral de riesgo asumible, el subescenario de defensa no ha cumplido su objetivo y MAGERIT propone incorporar nuevas salvaguardas y repetir todo el proceso hasta que el riesgo residual sea menor al umbral de riesgo aceptable.

Las salvaguardas se incorporan en distintos momentos y con distintas formas. Unas son anteriores a la materialización de la amenaza y otras posteriores. En el subescenario de defensa se incorporan las salvaguardas en dos niveles:

- Incorporación funcional de Servicios de Salvaguarda.
- Incorporación material de Mecanismos de Salvaguarda.

Vista Dinámica Física

Esta vista puede resultar útil para los especialistas en seguridad que utilizan MAGERIT en sistemas críticos que necesitan modelos específicos de simulación de procesos con el propósito de observar el funcionamiento y determinar los parámetros óptimos a efectos de seguridad. También es útil para dar soporte a herramientas sofisticadas de selección de salvaguardas y cálculo de riesgos.

Las Entidades de Seguridad pueden identificarse con las variables y parámetros de nivel (de estado) y de flujo (de acción) que caracterizan la simulación de los modelos de sistemas dinámicos.

Bajo este enfoque, cada agresión se muestra como un flujo de la acción que modifica el nivel de seguridad de un activo, es decir, es una acción que lo hace pasar de un estado de seguridad a otro. El impacto es la medida del resultado de la agresión sobre el activo, es decir, es la medida del flujo de la acción de cambio de nivel de seguridad del activo. El Riesgo es un indicador del nivel de seguridad.

Submodelo de Procesos

En el submodelo de procesos se ordenan las acciones que se van a realizar durante un proyecto de Análisis y Gestión de Riesgos. Tiene una estructura jerárquica de tres niveles: etapas, actividades y tareas. Las etapas se componen de actividades y éstas de tareas.

Las tareas tienen la misma acepción que en METRICA V3. Para describirlas se especifican los siguientes puntos:

- Acciones que se van a realizar.
- Actores que intervienen en la tarea o que están afectados por ésta.
- Productos y documentos que se van a obtener.
- Validaciones que se van a realizar de los resultados obtenidos.
- Técnicas que se van a utilizar.

Las actividades agrupan a un conjunto de tareas siguiendo generalmente criterios funcionales. Tienen la misma acepción que en METRICA V3.

Las etapas agrupan a un conjunto de actividades y se corresponden con los procesos de METRICA V3 (fases en METRICA V2.1). Las etapas marcan los puntos de toma de decisiones y entrega de productos. Es necesario que al final de cada etapa haya una aceptación formal de sus resultados ya que se utiliza el producto final de cada etapa como inicio de la siguiente.

MAGERIT propone las cuatro etapas siguientes:

- Etapa 1. Planificación del Análisis y Gestión de Riesgos. Su objetivo es definir el marco de referencia para la realización del proyecto de Análisis y Gestión de Riesgos.
- Etapa 2. Análisis de Riesgos. Se identifican las entidades y se valoran, obteniendo una evaluación del riesgo y una estimación del umbral de riesgo aceptable.
- Etapa 3. Gestión de Riesgos. Se identifican las funciones de salvaguarda para los riesgos detectados en la etapa anterior y se seleccionan las que son aceptables basándose en las ya existentes y en las restricciones.

- Etapa 4. Selección de salvaguardas. Se seleccionan los mecanismos de salvaguardas que se van a implantar y los procedimientos de seguimiento de dicha implantación.

MAGERIT tiene interfaces de enlace con METRICA V3, enlazando con los procesos de Planificación, Análisis, Diseño, Construcción e Implementación. En cada enlace se recogen los productos, los trata con procedimientos de aseguramiento y devuelve las salvaguardas.

Etapa 1. Planificación del Análisis y Gestión de Riesgos

En esta etapa se establecen las condiciones necesarias para iniciar el proyecto de Análisis y Gestión de Riesgos: se definen los objetivos del proyecto y el ámbito de actuación (dominio), se planifican los medios materiales y humanos para su ejecución y se particularizan las técnicas que se van a utilizar en las actividades del proyecto.

MAGERIT incluye en esta etapa las siguientes actividades:

- Oportunidad de realización.
- Definición de dominio y objetivos.
- Organización y planificación del proyecto.
- Lanzamiento del proyecto.

La primera actividad, oportunidad de realización, tiene por objetivo despertar el interés de la Dirección en el proyecto de Análisis y Gestión de Riesgos y consta de una única tarea, clarificar la oportunidad de realización.

La iniciativa de la realización del proyecto parte de un promotor (interno o externo a la organización) que es consciente de los problemas de seguridad que existen. Este sujeto elabora un cuestionario sobre aspectos de seguridad para ser respondido tanto por los responsables de informática como del resto de unidades de la Organización con el objetivo de recabar información y sensibilizar a la Organización sobre las cuestiones de seguridad. Con estos elementos realiza un informe preliminar en el que se incluyen los argumentos básicos para la realización del proyecto de Análisis y Gestión de Riesgos y una primera aproximación del Dominio del proyecto y de los medios materiales y humanos necesarios para hacerlo.

En la segunda actividad, definición de dominio y objetivos, se definen los límites del Dominio y los objetivos finales del proyecto. Además se identifican las restricciones que hay que considerar y las personas de las que se va a recoger información. Consta de las siguientes tareas:

- Especificar los objetivos del proyecto.
- Definir el dominio y los límites del proyecto.
- Identificar el entorno y las restricciones generales.
- Estimar dimensión, coste y retornos del proyecto.

En la tercera actividad, organización y planificación del proyecto, se calcula la carga de trabajo que va a suponer el proyecto y se establece el grupo y plan de trabajo. Las tareas de que consta son:

- Evaluar cargas y planificar entrevistas.
- Organizar a los participantes.
- Planificar el trabajo.

En la cuarta actividad, lanzamiento del proyecto, se asignan los recursos requeridos para el comienzo del proyecto y se eligen las técnicas primordiales de Análisis y Gestión de Riesgos. Incluye las siguientes tareas:

- Adaptar los cuestionarios.
- Seleccionar criterios de evaluación y técnicas para el proyecto.
- Asignar los recursos necesarios.

- Sensibilizar (campaña informativa).

Etapa 2. Análisis de Riesgos

Esta etapa constituye el núcleo de MAGERIT, la utilidad y validez de todo el proyecto depende de su correcta aplicación. El objetivo principal es la evaluación del riesgo del sistema en estudio, tanto del intrínseco como del efectivo, al que se añaden la presentación al Comité de Dirección de las áreas de mayor riesgo y la aprobación de los umbrales de riesgo asumibles.

La estimación de los Impactos y Vulnerabilidades es una tarea compleja y con un cierto grado de incertidumbre, lo que unido a su enorme influencia en la determinación del Riesgo trae consigo que deba ser dirigida por un grupo de expertos.

La etapa de Análisis de Riesgos comprende las siguientes actividades:

- Recogida de información.
- Identificación y agrupación de activos.
- Identificación y evaluación de amenazas.
- Identificación y estimación de vulnerabilidades.
- Identificación y valoración de impactos.
- Evaluación del Riesgo.

Durante la primera actividad, recogida de información, se recoge la información del sistema y de los factores que influyen en la seguridad. Comprende las siguientes tareas:

- Preparar la información.
- Realizar las entrevistas.
- Analizar la información recogida.

En la segunda actividad, identificación y agrupación de activos, se identifican los activos y sus relaciones, y se profundiza en sus características a partir de la información recogida en la actividad anterior. Incluye las tareas:

- Identificar y agrupar activos.
- Identificar los mecanismos de salvaguarda existentes.
- Valorar activos.

En la tercera actividad, identificación y evaluación de amenazas, se incluyen las tareas:

- Identificar y agrupar amenazas.
- Establecer los árboles de fallos generados por amenazas.

La cuarta actividad, identificación y estimación de vulnerabilidades, incluye las tareas:

- Identificar vulnerabilidades.
- Estimar vulnerabilidades.

La quinta actividad, identificación y valoración de impactos, incluye las tareas:

- Identificar impactos.
- Tipificar impactos.
- Valorar impactos.

En la sexta etapa, evaluación del riesgo, se incluyen las tareas:

- Evaluar el riesgo intrínseco.
- Analizar las funciones de salvaguarda existentes.
- Evaluar el riesgo efectivo.

Etapa 3. Gestión del Riesgo

En esta etapa se seleccionan los servicios de salvaguarda más apropiados con el objetivo de reducir el riesgo hasta niveles que se consideren aceptables.

Comprende las siguientes actividades:

- Interpretación del riesgo.
- Identificación de servicios de salvaguarda y estimación de su efectividad.
- Selección de los servicios de salvaguarda.
- Cumplimiento de objetivos.

La primera actividad, interpretación del riesgo, tiene una única tarea, interpretar y manejar los riesgos. El riesgo efectivo calculado en la etapa anterior se compara con el umbral de riesgo para determinar si es asumible. Si no lo es, MAGERIT propone continuar con las dos siguientes actividades de la etapa: identificación de servicios de salvaguarda y estimación de su efectividad, y selección de los servicios de salvaguarda.

La segunda actividad, Identificación de servicios de salvaguarda y estimación de su efectividad, comprende las tareas:

- Identificar funciones y servicios de salvaguarda.
- Estimar la efectividad de las funciones y servicios de salvaguarda.

La tercera actividad, selección de servicios de salvaguarda, incluye las tareas:

- Aplicar los parámetros de selección (ordena la lista de funciones desalvaguarda según su efectividad para reducir el riesgo).
- Reevaluar el riesgo.

La última actividad, cumplimiento de objetivos, tiene una única tarea: determinar el cumplimiento de objetivos. Si los riesgos efectivos calculados en la tarea anterior no están por debajo de los umbrales de riesgo fijados se conservan provisionalmente los resultados parciales alcanzados y se vuelve a repetir toda la actividad de selección de los servicios de salvaguarda.

Etapa 4. Selección de Salvaguardas

En esta etapa se eligen los mecanismos de salvaguarda que materializan las funciones y servicios de salvaguarda seleccionados previamente, tomando como base su efectividad. Se estudian sus tipos, costes y relaciones y se analiza si existen contraindicaciones en su aplicación. Finalmente se establece un orden para su implantación.

Las actividades de esta etapa, y las tareas de cada una de ellas, son:

- Identificación de los mecanismos de salvaguarda.
 - Identificar posibles mecanismos.
 - Estudiar mecanismos implantados.
 - Incorporar restricciones.
- Selección de los mecanismos.
 - Identificar mecanismos a implantar.
 - Evaluar el riesgo con los mecanismos elegidos.
 - Seleccionar mecanismos a implantar.
- Especificación de los mecanismos a implantar.
 - Especificar los mecanismos a implantar.
- Orientación a la planificación de la implantación.
 - Priorizar mecanismos.
 - Evaluar los recursos necesarios.
 - Evaluar cronogramas tentativos.

- Integración de resultados.
 - Integrar los resultados.

Auditoría Informática: objetivos, alcance y metodología. Técnicas y herramientas.

Normas y estándares. Auditoría del ENS y de protección de datos. Auditoría de seguridad física.

Auditoria Informática

La auditoria informática es un proceso llevado a cabo por profesionales especialmente capacitados para el efecto, y que consiste en recoger, agrupar y evaluar evidencias para determinar si un sistema de información salvaguarda el activo empresarial, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización, utiliza eficientemente los recursos, y cumple con las leyes y regulaciones establecidas. Permiten detectar de forma sistemática el uso de los recursos y los flujos de información dentro de una organización y determinar qué información es crítica para el cumplimiento de su misión y objetivos, identificando necesidades, duplicidades, costes, valor y barreras, que obstaculizan flujos de información eficientes.

La auditoria informática es el proceso de recoger, agrupar y evaluar evidencias para determinar si un sistema de información salvaguarda el activo empresarial, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización y utiliza eficientemente los recursos.

Auditar consiste principalmente en estudiar los mecanismos de control que están implantados en una empresa u organización, determinando si los mismos son adecuados y cumplen unos determinados objetivos o estrategias, estableciendo los cambios que se deberían realizar para la consecución de los mismos. Los mecanismos de control pueden ser directivos, preventivos, de detección, correctivos o de recuperación ante una contingencia.

Los objetivos de la auditoria Informática son:

- El control de la función informática.
- El análisis de la eficiencia de los Sistemas Informáticos.
- La verificación del cumplimiento de la Normativa en este ámbito.
- La revisión de la eficaz gestión de los recursos informáticos.

La auditoria informática sirve para mejorar ciertas características en la empresa como:

- Desempeño
- Fiabilidad
- Eficacia
- Rentabilidad
- Seguridad
- Privacidad

Generalmente se puede desarrollar en alguna o combinación de las siguientes áreas:

- Gobierno corporativo
- Administración del Ciclo de vida de los sistemas

- Servicio de Entrega y Soporte
- Protección y Seguridad
- Planes de continuidad y Recuperación de desastres

La necesidad de contar con lineamientos y herramientas estándar para el ejercicio de la auditoria informática ha promovido la creación y desarrollo de mejores prácticas como **COBIT**, **COSO** e **ITIL**.

Actualmente la certificación de ISACA para ser CISA *Certified Information Systems Auditor* es una de las más reconocidas y avaladas por los estándares internacionales ya que el proceso de selección consta de un examen inicial bastante extenso y la necesidad de mantenerse actualizado acumulando horas (puntos) para no perder la certificación.

Tipos de Auditoria Informática

Dentro de la auditoria informática destacan los siguientes tipos (entre otros):

- **Auditoria de la gestión** : La contratación de bienes y servicios, documentación de los programas, etc.
- **Auditoria legal del Reglamento de Protección de Datos** : Cumplimiento legal de las medidas de seguridad exigidas por el Reglamento de desarrollo de la Ley Orgánica de Protección de Datos.
- **Auditoria de los datos** : Clasificación de los datos, estudio de las aplicaciones y análisis de los flujogramas.
- **Auditoria de las bases de datos** : Controles de acceso, de actualización, de integridad y calidad de los datos.
- **Auditoria de la seguridad** : Referidos a datos e información verificando disponibilidad, integridad, confidencialidad, autenticación y no repudio.
- **Auditoria de la seguridad física** : Referido a la ubicación de la organización, evitando ubicaciones de riesgo, y en algunos casos no revelando la situación física de esta. También está referida a las protecciones externas (arcos de seguridad, CCTV, vigilantes, etc) y protecciones del entorno.
- **Auditoria de la seguridad lógica** : Comprende los métodos de autenticación de los sistemas de información.
- **Auditoria de las comunicaciones** : Se refiere a la auditoria de los procesos de autenticación en los sistemas de comunicación.
- **Auditoria de la seguridad en producción** : Frente a errores, accidentes y fraudes.

Principales pruebas y herramientas para efectuar una auditoria informática

En la realización de una auditoria informática el auditor puede realizar las siguientes pruebas:

- **Pruebas sustantivas** : Verifican el grado de confiabilidad del SI del organismo. Se suelen obtener mediante observación, cálculos, muestreos, entrevistas, técnicas de examen analítico, revisiones y conciliaciones. Verifican asimismo la exactitud, integridad y validez de la información.
- **Pruebas de cumplimiento** : Verifican el grado de cumplimiento de lo revelado mediante el análisis de la muestra. Proporciona evidencias de que los controles claves existen y que son aplicables efectiva y uniformemente.

Las principales herramientas de las que dispone un auditor informáticos son:

- **Observación**
- **Realización de cuetionarios**
- **Entrevistas a auditados y no auditados**
- **Muestreo estadístico (Trazas y/o huellas)**

- **Flujogramas**
- **Listas de chequeo (checklist)**
- **Mapas conceptuales**
- **Inventario**

Fases Auditoria Informática

- Fase I: Conocimientos del Sistema
 - Aspectos Legales y Políticas Internas
 - Características del Sistema Operativo
 - Características de la aplicación de computadora
- Fase II: Análisis de transacciones y recursos
 - Definición de transacciones
 - Análisis de las transacciones
 - Análisis de los recursos
 - Relación entre transacciones y recursos
- Fase III: Análisis de riesgos y amenazas
 - Identificación de riesgos
 - Identificación de amenazas
 - Relación entre recursos/amenazas/riesgos
- Fase IV: Análisis de controles
 - Codificación de controles
 - Relación entre controles
 - Análisis de cobertura de los controles requeridos
- Fase V: Evaluación de controles
 - Objetivos de la evaluación
 - Plan de pruebas de los controles
 - Pruebas de controles
 - Análisis de resultados de las pruebas
- Fase VI: El informe de auditoria
 - Informe detallado de recomendaciones
 - Evaluación de las respuestas
 - Informe resumen para la alta gerencia
- Fase VII: Seguimiento de las Recomendaciones
 - Informes de seguimiento
 - Evaluación de los controles implantados

Normas, técnicas y procedimientos de auditoria en informática

El desarrollo de una auditoria se basa en la aplicación de normas, técnicas y procedimientos de auditoria.

Es fundamental mencionar que para el auditor en informática conocer los productos de software que han sido creados para apoyar su función aparte de los componentes de la propia computadora resulta esencial, esto por razones económicas y para facilitar el manejo de la información.

El auditor desempeña sus labores mediante la aplicación de una serie de conocimientos especializados que vienen a formar el cuerpo técnico de su actividad. El auditor adquiere responsabilidades, no solamente con la persona que directamente contratan sus servicios, sino con un número de personas desconocidas para él que van a utilizar el resultado de su trabajo como base para tomar decisiones.

La auditoria no es una actividad meramente mecánica, que implique la aplicación de ciertos procedimientos cuyos resultados, una vez llevados a cabo son de carácter

indudable. La auditoria requiere el ejercicio de un juicio profesional, sólido, maduro, para juzgar los procedimientos que deben seguirse y estimar los resultados obtenidos.

Normas

Las normas de auditoria son los requisitos mínimos de calidad relativos a la personalidad del auditor, al trabajo que desempeña y a la información que rinde como resultado de este trabajo.

Las normas de auditoria se clasifican en:

- **Normas personales** : son cualidades que el auditor debe tener para ejercer sin dolo una auditoria, basados en sus conocimientos profesionales así como en un entrenamiento técnico, que le permita ser imparcial a la hora de dar sus sugerencias.
- **Normas de ejecución del trabajo** : son la planificación de los métodos y procedimientos, tanto como papeles de trabajo a aplicar dentro de la auditoria.
- **Normas de información** : son el resultado que el auditor debe entregar a los interesados para que se den cuenta de su trabajo, también es conocido como informe o dictamen.

Técnicas

Se define a las técnicas de auditoria como “los métodos prácticos de investigación y prueba que utiliza el auditor para obtener la evidencia necesaria que fundamente sus opiniones y conclusiones, su empleo se basa en su criterio o juicio, según las circunstancias”.

Al aplicar su conocimiento y experiencia el auditor, podrá conocer los datos de la empresa u organización a ser auditada, que pudieran necesitar una mayor atención.

Las técnicas procedimientos están estrechamente relacionados, si las técnicas no son elegidas adecuadamente, la auditoria no alcanzará las normas aceptadas de ejecución, por lo cual las técnicas así como los procedimiento de auditoria tienen una gran importancia para el auditor.

Según el IMCP en su libro *Normas y procedimientos de auditoria* las técnicas se clasifican generalmente con base en la acción que se va a efectuar, estas acciones pueden ser oculares, verbales, por escrito, por revisión del contenido de documentos y por examen físico.

Siguiendo esta clasificación las técnicas de auditoria se agrupan específicamente de la siguiente manera:

- Estudio General
- Análisis
- Inspección
- Confirmación
- Investigación
- Declaración
- Certificación
- Observación
- Cálculo

Procedimientos

Al conjunto de técnicas de investigación aplicables a un grupo de hechos o circunstancias que nos sirven para fundamentar la opinión del auditor dentro de una auditoria, se les dan el nombre de procedimientos de auditoria informática.

La combinación de dos o más procedimientos, derivan en programas de auditoria, y al conjunto de programas de auditoria se le denomina plan de auditoria, el cual servirá al auditor para llevar una estrategia y organización de la propia auditoria.

El auditor no puede obtener el conocimiento que necesita para sustentar su opinión en una sola prueba, es necesario examinar los hechos, mediante varias técnicas de aplicación simultánea.

En General los procedimientos de auditoria permiten:

- Obtener conocimientos del control interno.
- Analizar las características del control interno.
- Verificar los resultados de control interno.
- Fundamentar conclusiones de la auditoria.

Por esta razón el auditor deberá aplicar su experiencia y decidir cual técnica o procedimiento de auditoria serán los más indicados para obtener su opinión.

Análisis de datos

Dentro de este trabajo, desarrollaremos diversos tipos de técnicas y procedimientos de auditoria, de los cuales destacan el análisis de datos, ya que para las organizaciones el conjunto de datos o información son de tal importancia que es necesario verificarlos y comprobarlos, así también tiene la misma importancia para el auditor ya que debe de utilizar diversas técnicas para el análisis de los datos, los cuales se describen a continuación:

- **Comparación de programas** : esta técnica se emplea para efectuar una comparación de código (fuente, objeto o comandos de proceso) entre la versión de un programa en ejecución y la versión de un programa piloto que ha sido modificado en forma indebida, para encontrar diferencias.
- **Mapeo y rastreo de programas** : esta técnica emplea un software especializado que permite analizar los programas en ejecución, indicando el número de veces que cada línea de código es procesada y las de las variables de memoria que estuvieron presentes.
- **Análisis de código de programas** : se emplea para analizar los programas de una aplicación. El análisis puede efectuarse en forma manual (en cuyo caso sólo se podría analizar el código ejecutable).
- **Datos de prueba** : se emplea para verificar que los procedimientos de control incluidos en los programas de una aplicación funcionen correctamente. Los datos de prueba consisten en la preparación de una serie de transacciones que contienen tanto datos correctos como datos erróneos predeterminados.
- **Datos de prueba integrados** : técnica muy similar a la anterior, con la diferencia de que en ésta se debe crear una entidad falsa dentro de los sistemas de información.
- **Análisis de bitácoras** : existen varios tipos de bitácoras que pueden ser analizadas por el auditor, ya sea en forma manual o por medio de programas especializados, tales como bitácoras de fallas del equipo, bitácoras de accesos no autorizados, bitácoras de uso de recursos, bitácoras de procesos ejecutados.
- **Simulación paralela** : técnica muy utilizada que consiste en desarrollar programas o módulos que simulen a los programas de un sistema en producción. El objetivo es procesar los dos programas o módulos de forma paralela e identificar diferencias entre los resultados de ambos.

Monitoreo

Dentro de las organizaciones todos los procesos necesitan ser evaluados a través del tiempo para verificar su calidad en cuanto a las necesidades de control, integridad y

confidencialidad, este es precisamente el ámbito de esta técnica, a continuación se muestran los procesos de monitoreo:

- M1 Monitoreo del proceso.
- M2 Evaluar lo adecuado del control Interno.
- M3 Obtención de aseguramiento independiente.
- M4 Proveer auditoria independiente.

Análisis de bitácoras

Hoy en día los sistema de cómputo se encuentran expuestos a distintas amenazas, las vulnerabilidades de los sistemas aumentan, al mismo tiempo que se hacen más complejos, el número de ataques también aumenta, por lo anterior, las organizaciones deben reconocer la importancia y utilidad de la información contenida en las bitácoras de los sistemas de computo así como mostrar algunas herramientas que ayuden a automatizar el proceso de análisis de las mismas.

El crecimiento de Internet enfatiza esta problemática, los sistemas de cómputo generan una gran cantidad de información, conocidas como bitácoras o archivos logs, que pueden ser de gran ayuda ante un incidente de seguridad, así como para el auditor.

Una bitácora puede registrar mucha información acerca de eventos relacionados con el sistema que la genera, los cuales pueden ser:

- Fecha y hora
- Direcciones IP origen y destino
- Dirección IP que genera la bitácora
- Usuarios
- Errores

La importancia de las bitácoras es la de recuperar información ante incidentes de seguridad, detección de comportamiento inusual, información para resolver problemas, evidencia legal, es de gran ayuda en las tareas de cómputo forense.

Las Herramientas de análisis de bitácoras más conocidas son las siguientes:

- Para UNIX: Logcheck, Swatch
- Para Windows: LogAgent

Las bitácoras contienen información crítica, es por ello que deben ser analizadas, ya que están teniendo mucha relevancia, como evidencia en aspectos legales.

El uso de herramientas automatizadas es de mucha utilidad para el análisis de bitácoras, es importante registrar todas las bitácoras necesarias de todos los sistemas de cómputo para mantener un control de las mismas.

Técnicas de auditoria asistida por computadora

La utilización de equipos de computación en las organizaciones, ha tenido una repercusión importante en el trabajo del auditor, no sólo en lo que se refiere a los sistemas de información, sino también al uso de las computadoras en la auditoria.

Al llevar a cabo auditorias donde existen sistemas computarizados, el auditor se enfrenta a muchos problemas de muy diversa condición, uno de ellos, es la revisión de los procedimientos administrativos de control interno establecidos en la empresa que es auditada.

La utilización de paquetes de programas generalizados de auditoria ayuda en gran medida a la realización de pruebas de auditoria, a la elaboración de evidencias plasmadas en los papeles de trabajo.

Según las técnicas de auditoria Asistidas por Computadora (CAAT) son la utilización de determinados paquetes de programas que actúan sobre los datos, llevando a cabo con más frecuencia los trabajos siguientes:

- Selección e impresión de muestras de auditorias sobre bases estadísticas o no estadísticas, a lo que agregamos, sobre la base de los conocimientos adquiridos por los auditores.
- Verificación matemática de sumas, multiplicaciones y otros cálculos en los archivos del sistema auditado.
- Realización de funciones de revisión analítica, al establecer comparaciones, calcular razones, identificar fluctuaciones y llevar a cabo cálculos de regresión múltiple.
- Manipulación de la información al calcular subtotales, sumar y clasificar la información, volver a ordenar en serie la información, etc.
- Examen de registros de acuerdo con los criterios especificados.
- Búsqueda de alguna información en particular, la cual cumpla ciertos criterios, que se encuentra dentro de las bases de datos del sistema que se audita.

Evaluación del control interno

En un ambiente de evolución permanente, determinado por las actuales tendencias mundiales, las cuales se centran en el plano económico soportadas por la evolución tecnológica, surge la necesidad de que la función de auditoria pretenda el mejoramiento de su gestión.

La práctica de nuevas técnicas para evaluar el control interno a través de las cuales, la función de auditoria informática pretende mejorar la efectividad de su función y con ello ofrecer servicios más eficientes y con un valor agregado.

La evolución de la teoría del control interno se definió en base a los principios de los controles como mecanismos o prácticas para prevenir, identificar actividades no autorizadas, más tarde se incluyó el concepto de lograr que las cosas se hagan; la corriente actual define al control como cualquier esfuerzo que se realice para aumentar las posibilidades de que se logren los objetivos de la organización.

En este proceso evolutivo se considera actualmente, y en muchas organizaciones que el director de finanzas o al director de auditoria como los responsables principales del correcto diseño y adecuado funcionamiento de los controles internos.

Benchmarking

Las empresas u organizaciones deben buscar formas o fórmulas que las dirijan hacia una mayor calidad, para poder ser competitivos, una de estas herramientas o fórmulas es el Benchmarking.

Benchmarking es el proceso continuo de medir productos, servicios y prácticas contra los competidores o aquellas compañías reconocidas como líderes en la industria.

Esta definición presenta aspectos importantes tales como el concepto de continuidad, ya que benchmarking no sólo es un proceso que se hace una vez y se olvida, sino que es un proceso continuo y constante.

Según la definición anterior podemos deducir que se puede aplicar benchmarking a todas las facetas de las organizaciones, y finalmente la definición implica que el benchmarking se debe dirigir hacia aquellas organizaciones y funciones de negocios dentro de las organizaciones que son reconocidas como las mejores.

Otra definición puede ser: "benchmarking es un proceso sistemático y continuo para comparar nuestra propia eficiencia en términos de productividad, calidad y prácticas con aquellas compañías y organizaciones que representan la excelencia.

Dentro del benchmarking existen los siguientes tipos:

- Benchmarking interno
- Benchmarking competitivo
- Benchmarking genérico

Auditoría Informática como objeto de estudio

La auditoría en informática es la revisión y la evaluación de los controles, sistemas, procedimientos de informática; de los equipos de cómputo, su utilización, eficiencia y seguridad, de la organización que participan en el procesamiento de la información, a fin de que por medio del señalamiento de cursos alternativos se logre una utilización más eficiente y segura de la información que servirá para una adecuada toma de decisiones. Los factores que pueden influir en una organización a través del control y la auditoría en informática, son:

- Necesidad de controlar el uso evolucionado de las computadoras.
- Controlar el uso de la computadora, que cada día se vuelve más importante y costosa.
- Altos costos que producen los errores en una organización.
- Abuso en las computadoras.
- Posibilidad de pérdida de capacidades de procesamiento de datos.
- Posibilidad de decisiones incorrectas.
- Valor del hardware, software y personal.
- Necesidad de mantener la privacidad individual.
- Posibilidad de pérdida de información o de mal uso de la misma.
- Necesidad de mantener la privacidad de la organización.

La información es un factor importante y cada día cobra más valor para una empresa u organización para la continuidad de las operaciones, ya que la imagen de su ambiente depende de la situación actual, su desarrollo y competitividad dependen del ambiente pasado y futuro, ya que tomar una decisión incorrecta mediante datos erróneos proporcionados por los sistemas trae como consecuencia efectos significativos que afectan directamente a la organización.

Objetivo fundamental de la auditoria Informática

La operatividad en una empresa es el punto más importante, es encargada de vigilar el funcionamiento de mínimos consistentes de la organización y las máquinas a nivel global como parcial. La auditoria se debe realizar en el momento en que la maquinaria informática está en funcionamiento, con el fin de identificar falencias que obstruyan la operatividad de las mismas, con el objeto de corregir o buscar alternativas de solución a tiempo, sin tener que parar el trabajo.

La operatividad de los sistemas ha de constituir entonces la principal preocupación del auditor informático. Para conseguirla hay que acudir a la realización de Controles Técnicos Generales de Operatividad y Controles Técnicos Específicos de Operatividad, previos a cualquier actividad de aquel.

Los Controles Técnicos Generales son importantes en las instalaciones de empresas grandes, ya que se realizan para verificar la compatibilidad de funcionamiento simultáneo del SO y el software de base con todos los subsistemas existentes, como también la compatibilidad del hardware y del software instalado. En una empresa existen diferentes entornos de trabajo que conlleva a la contratación de productos de software básico, así como software especial para algunos departamentos, con el riesgo de abonar más de una vez el mismo producto o desaprovechar el software instalado, así mismo puede existir software desarrollado por personal de sistemas de la misma empresa que hagan mal uso y que no se aprovechen todos los recursos de este, sobre todo cuando los diversos equipos

están ubicados en Centros de Proceso de datos geográficamente alejados. Lo negativo de esta situación es que puede producir la inoperatividad del conjunto. Cada Centro de Proceso de Datos tal vez sea operativo trabajando independientemente, pero no será posible la interconexión e intercomunicación de todos los centros de proceso de datos si no existen productos comunes y compatibles.

Los controles técnicos específicos, menos evidentes, son también necesarios para lograr la operatividad de los sistemas. Es decir por más pequeña que sea la aplicación que se deba ejecutar, esta debe funcionar al máximo, evitando así la inoperatividad, bien sea en hardware como en software.

Una vez conseguida la operatividad de los sistemas, el segundo objetivo de la auditoria es la verificación de la observación de las normas teóricamente existentes en el departamento de informática y su coherencia con las del resto de la empresa. Para ello, habrán de revisarse sucesivamente y en este orden:

1. Las normas generales de la instalación informática. Se realiza una revisión inicial sencilla, verificando la aplicación de las normas pero también registrando las áreas que no cumplan o que no las apliquen, sin olvidar que esta normativa no está en contradicción con alguna norma no informática de la empresa.
2. Los procedimientos generales informáticos. Se verificará su existencia, al menos en los sectores más importantes. Por ejemplo, la recepción definitiva de las máquinas deberá estar firmada por la persona responsable de este cargo. Tampoco el alta de una nueva aplicación podría producirse si no existieran los Procedimientos de Backup y recuperación correspondientes.
3. Los procedimientos específicos informáticos. Igualmente, se revisará su existencia en las áreas fundamentales. Así, explotación no debería explotar una aplicación sin haber exigido a desarrollo la pertinente documentación. Del mismo modo, deberá comprobarse que los procedimientos específicos no se opongan a los procedimientos generales. En todos los casos anteriores, a su vez, deberá verificarse que no existe contradicción alguna con la normativa y los procedimientos generales de la propia empresa, a los que la informática debe estar sometida.

Características de la auditoria informática

La información de la empresa y para la empresa, siempre importante, se ha convertido en un activo real de la misma, como sus Stocks o materias primas si las hay. Por ende, han de realizarse inversiones informáticas, materia de la que se ocupa la auditoria de Inversión informática.

Del mismo modo, los sistemas informáticos han de protegerse de modo global y particular: a ello se debe la existencia de la auditoria de seguridad informática en general, o a la auditoria de seguridad de alguna de sus áreas, como pudieran ser desarrollo o técnicas de sistemas.

Cuando se producen cambios estructurales en la informática, se reorganiza de alguna forma su función: se está en el campo de la auditoria de organización informática.

Estos tres tipos de auditorias engloban a las actividades auditadoras que se realizan en una auditoria parcial. De otra manera: cuando se realiza una auditoria del área de desarrollo de proyectos de la informática de una empresa, es porque en ese desarrollo existen, además de ineficiencias, debilidades de organización, o de inversiones, o de seguridad, o alguna mezcla de ellas.

Teniendo en cuenta lo anterior y partiendo de las diferentes actividades de sistemas que cada empresa tiene dentro de su organización dentro de las áreas generales, se establecen las siguientes divisiones de auditoria informática de Explotación, de sistemas, de comunicaciones y de desarrollo de proyectos. Estas son las áreas específicas de la

auditoria informática más importantes. Cada área específica puede ser auditada desde los siguientes criterios generales, que pueden modificarse según sea el tipo de empresa a auditar:

- Desde su propio funcionamiento interno.
- Desde el apoyo que recibe de la dirección y, en sentido ascendente, del grado de cumplimiento de las directrices de ésta.
- Desde la perspectiva de los usuarios, destinatarios reales de la informática.
- Desde el punto de vista de la seguridad que ofrece la informática en general o la rama auditada.

Clasificación de la auditoria informática

Auditoria informática de explotación

Eplotación informática se encarga de obtener resultados informáticos, como son: listados impresos, ficheros soportados magnéticamente, órdenes automatizadas para lanzar o modificar procesos industriales, entre otras. Los datos es la materia prima que hay que transformar por medio del proceso informático (gobernado por programas), bajo el criterio de integridad y control de calidad y así lleguen finalmente al usuario. Para auditar explotación hay que auditar las sesiones que la componen y sus interrelaciones.

Auditoria informática de sistemas

Encargada de analizar todo lo concerniente a técnica de sistemas en todas sus facetas, teniendo como resultado en la actualidad que todo lo que forme el entorno general de sistemas, como son las comunicaciones, líneas y redes de las instalaciones informáticas, se auditen por separado. Dentro de la auditoria informática de sistemas se evalúa lo siguiente:

- *Sistemas operativos* : debe verificarse en primer lugar que los sistemas estén actualizados con las últimas versiones del fabricante, indagando las causas de las omisiones si las hubiera. El análisis de las versiones de los SO permite descubrir las posibles incompatibilidades entre otros productos de software básico adquiridos por la instalación y determinadas versiones de aquellas.
- *Software básico* : es fundamental para el auditor conocer los productos de software básico que han sido adquiridos aparte de la propia computadora. Esto, por razones económicas y por razones de comprobación de que la computadora podría funcionar sin el producto adquirido por el cliente. En cuanto al software desarrollado por el personal informático de la empresa, el auditor debe verificar que éste no agrede ni condiciona al Sistema. Igualmente, debe considerar el esfuerzo realizado en términos de costes, por si hubiera alternativas más económicas.
- *Tunning* : es el conjunto de técnicas de observación y de medidas encaminadas a la evaluación del comportamiento de los Subsistemas y del sistema en su conjunto. Las acciones de tunning deben diferenciarse de los controles habituales que realiza el personal de técnica de sistemas. El tunning posee una naturaleza más revisora, estableciéndose previamente planes y programas de actuación según los síntomas observados.
- *Optimización de los sistemas y subsistemas* : la técnica de sistemas debe realizar acciones permanentes de optimización como consecuencia de la realización de tunnings pre-programados o específicos. El auditor verificará que las acciones de optimización fueron efectivas y no comprometieron la operatividad de los sistemas ni el plan crítico de producción diaria de explotación.
- *Administración de base de datos* : el diseño de las BD, sean relacionales o jerárquicas, se ha convertido en una actividad muy compleja y sofisticada, por lo general desarrollada en el ámbito de técnica de sistemas, y de acuerdo con las áreas de desarrollo y usuarios de la empresa. Al conocer el diseño y arquitectura de éstas por parte de Sistemas, se les encomienda también su administración. Al auditor de

BD analizará los sistemas de salvaguarda existentes, revisará finalmente la integridad y consistencia de los datos, así como la ausencia de redundancias entre ellos.

- *Investigación y desarrollo* : como empresas que utilizan y necesitan de informáticas desarrolladas, saben que sus propios efectivos están desarrollando aplicaciones y utilidades que, concebidas inicialmente para su uso interno, pueden ser susceptibles de adquisición por otras empresas, haciendo competencia las compañías del mismo campo. La auditoria informática deberá cuidar de que la actividad de investigación y desarrollo no interfiera ni dificulte las tareas fundamentales internas. La propia existencia de aplicativos para la obtención de estadísticas desarrollados por los técnicos de sistemas de la empresa auditada, y su calidad, proporcionan al auditor experto una visión bastante exacta de la eficiencia y estado de desarrollo de los sistemas.

Auditoria informática de comunicaciones y redes

Para el informático y para el auditor informático, el entramado que constituyen las redes nodales, líneas, concentradores, multiplexores, redes locales, etc. no son sino el soporte físico-lógico del tiempo real. El auditor tropieza con la dificultad técnica del entorno, pues ha de analizar situaciones y hechos alejados entre sí, y está condicionado a la participación del monopolio telefónico que presta el soporte. Como en otros casos, la auditoría de este sector requiere un equipo de especialistas, expertos simultáneamente en comunicaciones y en redes locales (no hay que olvidarse que en entornos geográficos reducidos, algunas empresas optan por el uso interno de redes locales, diseñadas y cableadas con recursos propios).

El auditor de comunicaciones deberá inquirir sobre los índices de utilización de las líneas contratadas con información abundante sobre tiempos de desuso. Deberá proveerse de la topología de la red de comunicaciones, actualizada, ya que la desactualización de esta documentación significaría una grave debilidad. La inexistencia de datos sobre las líneas existen, cómo son y donde están instaladas, supondría que se bordea la inoperatividad informática. Sin embargo, las debilidades más frecuentes o importantes se encuentran en las disfunciones organizativas. La contratación e instalación de líneas va asociada a la instalación de los puestos de trabajo correspondientes (Pantallas, Servidores de redes locales, computadoras con tarjetas de comunicaciones, impresoras, etc). Todas estas actividades deben estar muy coordinadas y de ser posible, dependientes de una sola organización.

Auditoria informática de desarrollo de proyectos o aplicaciones

El desarrollo es una evolución del llamado análisis y programación de sistemas y aplicaciones, que a su vez, engloba muchas áreas que tiene la empresa. Una aplicación recorre las siguientes fases:

- Pre-requisitos del usuario (único o plural) y del entorno
- Análisis funcional
- Diseño
- Análisis orgánico (pre-programación y programación)
- Pruebas
- Entrega a explotación y alta para el proceso

Estas fases deben estar sometidas a un exigente control interno, caso contrario, además del disparo de los costes, podrá producirse la insatisfacción del usuario. Finalmente, la auditoria deberá comprobar la seguridad de los programas en el sentido de garantizar que los ejecutados por la máquina sean exactamente los previstos y no otros.

Auditoria de la seguridad informática

La computadora es un instrumento que estructura gran cantidad de información, la cual puede ser confidencial para individuos, empresas o instituciones, y puede ser mal utilizada o divulgada a personas que hagan mal uso de esta. También pueden ocurrir robos, fraudes o sabotajes, virus, etc, que provoquen la destrucción total o parcial de la actividad computacional. Esta información puede ser de suma importancia, y el no tenerla en el momento preciso puede provocar retrasos sumamente costosos.

Al auditar los sistemas se debe tener cuidado que no se tengan copias "piratas" o bien que, al conectarse en red con otras computadoras, no exista la posibilidad de transmisión del virus. El uso inadecuado de la computadora comienza desde la utilización de tiempo de máquina para usos ajenos de la organización, la copia de programas para fines de comercialización sin reportar los derechos de autor hasta el acceso por vía telefónica a BD a fin de modificar la información con propósitos fraudulentos. La seguridad en la informática abarca los conceptos de seguridad física y seguridad lógica.

Metodología de auditoria informática

Como auditor se debe recolectar toda la información general, que permita así mismo definir un juicio global objetivo siempre amparadas en pruebas o hechos demostrables. Dar como resultado un informe claro, conciso y a la vez preciso depende del análisis y experiencia del auditor, frente a diferentes entornos a evaluar, dependiendo de las debilidades y fortalezas encontradas en dicha empresa auditada. La recolección de información, el análisis, la aplicación de diferentes normas de acuerdo al tipo de auditoria, los hallazgos encontrados y pruebas que avalen estos resultados son indispensables en la realización de una auditoria. Para llegar al resultado hay que seguir una serie de pasos que permiten tener claridad y orden de la auditoria a aplicar.

El método de trabajo del auditor pasa por las siguientes etapas:

1. Alcance y objetivos de la auditoria informática.
2. Estudio inicial del entorno auditabile.
3. Determinación de los recursos necesarios para realizar la auditoria.
4. Elaboración del plan y de los programas de trabajo.
5. Actividades propiamente dichas de la auditoria.
6. Confección y redacción del informe final.
7. Redacción de la carta de introducción o carta de presentación del informe final.

Alcance y objetivos de la auditoria informática

El alcance de la auditoria expresa los límites de la misma. Debe existir un acuerdo muy preciso entre auditores y clientes sobre las funciones, las materias y las organizaciones a auditar. A los efectos de acotar el trabajo, resulta muy beneficioso para ambas partes expresar las excepciones de alcance de la auditoria, es decir cuales materias, funciones u organizaciones no van a ser auditadas. Tanto los alcances como las excepciones deben figurar al comienzo del informe final.

Estudio inicial del entorno auditabile

Esta etapa es una de las más importantes en el desarrollo de la auditoria, ya que el auditor debe conocer todos los procesos desarrollados, relacionado con el área tomada como caso de estudio. Para realizar dicho estudio ha de examinarse las funciones y actividades generales de la informática. Para su realización el auditor debe conocer lo siguiente:

Organización: para el auditor, el conocimiento de quién ordena, quién diseña y quién ejecuta es fundamental. Para realizarlo el auditor deberá fijarse en:

- *Organigrama* : el organigrama expresa la estructura oficial de la organización a auditar. Permite identificar las jerarquías, dependencias y direcciones entre las áreas existentes.
 - Departamentos: se entiende como departamento a los órganos que siguen inmediatamente a la Dirección. El equipo auditor describirá brevemente las funciones de cada uno de ellos.
 - Relaciones Jerárquicas y funcionales entre órganos de la Organización: el auditor verificará si se cumplen las relaciones funcionales y jerárquicas previstas por el organigrama, o por el contrario detectará, por ejemplo, si algún empleado tiene dos jefes. Las de jerarquía implican la correspondiente subordinación. Las funcionales por el contrario, indican relaciones no estrictamente subordinables.
 - Flujos de información: además de las corrientes verticales intra-departamentales, la estructura organizativa cualquiera que sea, produce corrientes de información horizontales y oblicuas extra-departamentales.
 - Número de puestos de trabajo: el equipo auditor comprobará que los nombres de los puestos de trabajo de la organización corresponden a las funciones reales distintas.
 - Número de personas por puesto de trabajo: es un parámetro que los auditores informáticos deben tener en cuenta ya que la inadecuación del personal determina que el número de personas que realizan las mismas funciones rara vez coincida con la estructura oficial de la organización.
- *Entorno operacional* : el auditor informático debe tener una referencia del entorno en el que va a desenvolverse y se obtiene determinando lo siguiente:
 - Situación geográfica de los sistemas: se determinará la ubicación geográfica de los distintos centros de proceso de datos en la empresa, continuando con la verificación de la existencia de responsables en cada uno de ellos, así como el uso de los mismos estándares de trabajo.
 - Arquitectura y configuración de hardware y software: cuando existen varios equipos, es fundamental la configuración elegida para cada uno de ellos, ya que los mismos deben constituir un sistema compatible e intercomunicado. La configuración de los sistemas está muy ligada a las políticas de seguridad lógica de las compañías, para esto es importante que los auditores, en su estudio inicial, tengan en su poder la distribución e interconexión de los equipos.
 - Inventario de hardware y software: el auditor recabará información escrita, en donde figuren todos los elementos físicos y lógicos de la instalación. En cuanto al hardware figurarán las CPU's, unidades de control local y remotas, periféricos de todo tipo, etc. El inventario de software debe contener todos los productos lógicos del sistema, desde el software básico hasta los programas de utilidad adquiridos o desarrollados internamente. Suele ser habitual clasificarlos en facturables y no facturables.
 - Comunicación y redes de comunicación: al realizar el estudio inicial los auditores dispondrán del número, situación y características principales de las líneas, así como de los accesos a la red pública de comunicaciones, igualmente, poseerán información de las redes locales de la Empresa y todo lo que tenga que ver con la red de comunicaciones.

Determinación de los recursos necesarios para realizar la auditoria

Mediante los resultados del estudio inicial realizado se procede a determinar los recursos humanos y materiales que han de emplearse en la auditoria.

Recursos humanos : la cantidad de recursos depende del volumen auditável. Las características y perfiles del personal seleccionado dependen de la materia auditável. Es

igualmente señalable que la auditoria en general suele ser ejercida por profesionales universitarios y por otras personas de probada experiencia multidisciplinaria.

Recursos materiales : los recursos materiales del auditor son de dos tipos:

- Recursos software como son, cantidad y complejidad de BD y ficheros, que son programas propios de la auditoria, son muy potentes y flexibles.
- Recursos materiales hardware: los recursos hardware que el auditor necesita son proporcionados por el cliente. Los procesos de control deben efectuarse necesariamente en las computadoras del auditado. Por lo cual habrá de convenir, tiempo de máquina, espacio de disco, impresoras ocupadas, scanner, etc.

Elaboración del plan y de los programas de trabajo

Una vez asignados los recursos, el responsable de la auditoria y sus colaboradores establecen un plan de trabajo y así, se procede a la programación del mismo. El plan se elabora teniendo en cuenta, entre otros criterios, los siguientes:

- Si la revisión debe realizarse por áreas generales o áreas específicas.
- Si la auditoria es global, de toda la informática, o parcial. El volumen determina no solamente el número de auditores necesarios, sino las especialidades necesarias del personal.
- En el Plan no se consideran calendarios, porque se manejan recursos genéricos y no específicos.
- En el Plan se establecen los recursos y esfuerzos globales que van a ser necesarios.
- En el Plan se establecen las prioridades de materias auditables, de acuerdo siempre con las prioridades del cliente.
- El Plan establece disponibilidad futura de los recursos durante la revisión.
- El Plan estructura las tareas a realizar por cada integrante del grupo.
- En el Plan se expresan todas las ayudas que el auditor ha de recibir del auditado.

Una vez elaborado el plan, se procede a la programación de actividades, esta ha de ser lo suficientemente flexible como para permitir modificaciones a lo largo del proyecto.

Actividades propiamente dichas de la auditoria informática

La auditoria informática general se realiza por áreas generales o por áreas específicas. Si se examina por grandes temas, resulta evidente la mayor calidad y el empleo de más tiempo total y mayores recursos. Cuando la auditoria se realiza por áreas específicas, se abarcan de una vez todas las peculiaridades que afectan a la misma, de forma que el resultado se obtiene más rápidamente y con menor calidad. Existen técnicas que hacen que el auditor las aplique de acuerdo a su juicio y al tipo de auditoria a ejecutar son:

- *Técnicas de Trabajo :*
 - Análisis de la información obtenida del auditado.
 - Análisis de la información propia.
 - Cruzamiento de las informaciones anteriores.
 - Entrevistas.
 - Simulación.
 - Muestreos.
 - Inspección.
 - Confirmación.
 - Investigación.
 - Certificación.
 - Observación.
- *Herramientas :*
 - Cuestionario general inicial.
 - Cuestionario Checklist.
 - Estándares.

- Monitores.
- Simuladores (Generadores de datos).
- Paquetes de auditoria (Generadores de programas)
- Matrices de riesgo.

Confección y redacción del informe final

La función de la auditoria se materializa exclusivamente por escrito. Por lo tanto, la elaboración final es el exponente de su calidad. Resulta evidente la necesidad de redactar borradores e informes parciales previos al informe final, los que son elementos de contraste entre opinión entre auditor y auditado y que pueden descubrir fallos de apreciación en el auditor.

Redacción de la carta de introducción o carta de presentación del informe final

La carta de introducción tiene especial importancia porque en ella ha de resumirse la auditoria realizada. Se destina exclusivamente al responsable máximo de la empresa, o a la persona concreta que encargó o contrató la auditoria.

Así como pueden existir tantas copias del informe final como solicite el cliente, la auditoria no hará copias de la citada carta de introducción. La carta de introducción poseerá los siguientes atributos:

- Tendrá como máximo 4 folios.
- Incluirá fecha, naturaleza, objetivos y alcance.
- Cuantificará la importancia de las áreas analizadas.
- Proporcionará una conclusión general, concretando las áreas de gran debilidad.
- Presentará las debilidades en orden de importancia y gravedad.

En la carta de introducción no se escribirán nunca recomendaciones.

Estructura del informe final : el informe comienza con la fecha de comienzo de la auditoria y la fecha de redacción del mismo. Se incluyen los nombres del equipo auditor y los nombres de todas las personas entrevistadas, con indicación de la jefatura, responsabilidad y puesto de trabajo que ostente. Siguiendo los siguientes pasos:

- Definición de objetivos y alcance de la auditoria.
- Enumeración de temas considerados.
- Cuerpos expositivo.

Para cada tema, se seguirá el siguiente orden:

1. Situación actual. Cuando se trate de una revisión periódica, en la que se analiza no solamente una situación sino además su evolución en el tiempo, se expondrá la situación prevista y la situación real.
2. Tendencias. Se tratarán de hallar parámetros que permitan establecer tendencias futuras.
3. Puntos débiles y amenazas.
4. Recomendaciones y planes de acción. Constituyen junto con la exposición de puntos débiles, el verdadero objetivo de la auditoria informática.
5. Redacción posterior de la carta de introducción o presentación.

Modelo conceptual de la exposición del informe final:

- El informe debe incluir solamente hechos importantes. La inclusión de hechos poco relevantes o accesorios desvía la atención del lector.
- El informe debe consolidar los hechos que se describen en el mismo.

- El término de “hechos consolidados” adquiere un especial significado de verificación objetiva y de estar documentalmente probados y soportados. La consolidación de los hechos debe satisfacer, al menos los siguientes criterios:

1. El hecho debe poder ser sometido a cambios.
2. Las ventajas del cambio deben superar los inconvenientes derivados de mantener la situación.
3. No deben existir alternativas viables que superen al cambio propuesto.
4. La recomendación del auditor sobre el hecho, debe mantener o mejorar las normas y estándares existentes en la instalación.

La aparición de un hecho en un informe de auditoría implica necesariamente la existencia de una debilidad que ha de ser corregida.

Flujo del hecho o debilidad:

Hecho encontrado.

- A de ser relevante para el auditor y para el cliente.
- A de ser exacto, y además convincente.
- No deben existir hechos repetidos.

Consecuencias del hecho : las consecuencias deben redactarse de modo que sean directamente deducibles del hecho.

Repercusión del hecho : se redactará las influencias directas que el hecho pueda tener sobre otros aspectos informáticos u otros ámbitos de la empresa.

Conclusión del hecho : no deben redactarse conclusiones más que en los casos en que la exposición haya sido muy extensa o compleja.

Recomendación del auditor informático

- Deberá entenderse por sí sola, por simple lectura.
- Deberá estar suficientemente soportada en el propio texto.
- Deberá ser concreta y exacta en el tiempo, para que pueda ser verificada su implementación.
- La recomendación se redactará de forma que vaya dirigida expresamente a la persona o personas que puedan implementarla.

Herramientas y técnicas para la auditoría informática

Cuestionarios

Las auditorías informáticas se materializan recabando información y documentación de todo tipo. Los informes finales de los auditores dependen de sus capacidades para analizar las situaciones de debilidad o fortaleza de los diferentes entornos. El trabajo de campo del auditor consiste en lograr toda la información necesaria para la emisión de un juicio global objetivo, siempre amparado en hechos demostrables, llamados también evidencias.

Para esto, suele ser habitual comenzar solicitando la cumplimentación de cuestionarios pre-impresos que se envían a las personas concretas que el auditor cree adecuadas, sin que sea obligatorio que dichas personas sean las responsables oficiales de las diversas áreas a auditar. Estos cuestionarios no pueden ni deben ser repetidos para instalaciones distintas, sino diferentes y muy específicos para cada situación, y muy cuidados en su fondo y su forma.

Sobre esta base, se estudia y analiza la documentación recibida, de modo que tal análisis determine a su vez la información que deberá elaborar el propio autor. El cruzamiento de ambos tipos de información es una de las bases fundamentales de la auditoría.

Cabe aclarar, que esta primera fase puede omitirse cuando los auditores hayan adquirido por otros medios la información que aquellos pre-impresos hubiesen proporcionado.

Entrevistas

El auditor comienza a continuación las relaciones personales con el auditado. Lo hace de tres formas:

1. Mediante la petición de documentación concreta sobre alguna materia de su responsabilidad.
2. Mediante “entrevistas” en las que no se sigue un plan determinado ni un método estricto de sometimiento a un cuestionario.
3. Por medio de entrevistas en las que el auditor sigue un método preestablecido de antemano y busca unas finalidades concretas.

La entrevista es una de las actividades personales más importantes del auditor; en ellas, éste recoge más información, y mejor matizada, que la proporcionada por medios propios puramente técnicos o por las respuestas escritas a cuestionarios.

Aparte de algunas cuestiones menos importantes, la entrevista entre auditor y auditado se basa fundamentalmente en el concepto de interrogatorio; es lo que hace un auditor, interroga y se interroga a sí mismo. El auditor informático experto entrevista al auditado siguiendo un cuidadoso sistema previamente establecido, consistente en que bajo la forma de una conversación correcta y lo menos tensa posible, el auditado conteste sencillamente y con pulcritud a una serie de preguntas variadas, también sencillas. Sin embargo, esta sencillez es solo aparente. Tras ella debe existir una preparación muy elaborada y sistematizada, y que es diferente para cada caso particular.

Checklist

El auditor profesional y experto es aquél que reelabora muchas veces sus cuestionarios en función de los escenarios auditados. Tiene claro lo que necesita saber, y por qué. Sus cuestionarios son vitales para el trabajo de análisis, cruzamiento y síntesis posterior, lo cual no quiere decir que haya que someter al auditado a unas preguntas estereotipadas que no conducen a nada. Muy por el contrario, el auditor conversará y hará preguntas “normales”, que en realidad servirán para la cumplimentación sistemática de sus cuestionarios, de sus checklists.

Hay opiniones que descalifican el uso de las checklists, ya que consideran que leerle una pila de preguntas recitadas de memoria o leídas en voz alta descalifica al auditor informático. Pero esto no es usar checklists, es una evidente falta de profesionalismo. El profesionalismo pasa por un procesamiento interno de información a fin de obtener respuestas coherentes que permitan una correcta descripción de puntos débiles y fuertes. El profesionalismo pasa por poseer preguntas muy estudiadas que han de formularse flexiblemente.

El conjunto de estas preguntas recibe el nombre de checklist. Salvo excepciones, las checklists deben ser contestadas oralmente, ya que superan en riqueza y generalización a cualquier otra forma.

Según la claridad de las preguntas y el talante del auditor, el auditado responderá desde posiciones muy distintas y con disposición muy variable. El auditado, habitualmente informático de profesión, percibe con cierta facilidad el perfil técnico y los conocimientos

del auditor, precisamente a través de las preguntas que éste le formula. Esta percepción configura el principio de autoridad y prestigio que el auditor debe poseer.

Por ello, aun siendo importante tener elaboradas listas de preguntas muy sistematizadas, coherentes y clasificadas por materias, todavía lo es más el modo y el orden de su formulación. Las empresas externas de auditoría informática guardan sus checklists, pero de poco sirven si el auditor no las utiliza adecuada y oportunamente. No debe olvidarse que la función auditora se ejerce sobre bases de autoridad, prestigio y ética.

El auditor deberá aplicar la checklist de modo que el auditado responda clara y concisamente. Se deberá interrumpir lo menos posible a éste, y solamente en los casos en que las respuestas se aparten sustancialmente de la pregunta. En algunas ocasiones, se hará necesario invitar a aquél a que exponga con mayor amplitud un tema concreto, y en cualquier caso, se deberá evitar absolutamente la presión sobre el mismo.

Algunas de las preguntas de las checklists utilizadas para cada sector, deben ser repetidas. En efecto, bajo apariencia distinta, el auditor formulará preguntas equivalentes a las mismas o a distintas personas, en las mismas fechas, o en fechas diferentes. De este modo, se podrán descubrir con mayor facilidad los puntos contradictorios; el auditor deberá analizar los matices de las respuestas y reelaborar preguntas complementarias cuando hayan existido contradicciones, hasta conseguir la homogeneidad. El entrevistado no debe percibir un excesivo formalismo en las preguntas. El auditor, por su parte, tomará las notas imprescindibles en presencia del auditado, y nunca escribirá cruces ni marcará cuestionarios en su presencia.

Los cuestionarios o checklists responden fundamentalmente a dos tipos de "filosofía" de calificación o evaluación:

Checklist de rango

Contiene preguntas que el auditor debe puntuar dentro de un rango preestablecido (por ejemplo, de 1 a 5, siendo 1 la respuesta más negativa y el 5 el valor más positivo).

Ejemplo: Se supone que se está realizando una auditoría sobre la seguridad física de una instalación y, dentro de ella, se analiza el control de los accesos de personas y cosas al Centro de Cálculo. Podrían formularse las preguntas que figuran a continuación, en donde las respuestas tienen los siguientes significados:

1. Muy deficiente.
2. Deficiente.
3. Mejorable.
4. Aceptable.
5. Correcto.

Se figuran posibles respuestas de los auditados. Las preguntas deben sucederse sin que parezcan encorsetadas ni clasificadas previamente. Basta con que el auditor lleve un pequeño guión. La cumplimentación de la checklist no debe realizarse en presencia del auditado. Ejemplo:

- ¿Existe personal específico de vigilancia externa al edificio?
 - Respuesta: No, solamente un guarda por la noche que atiende además otra instalación adyacente.
 - Puntuación: 1
- ¿Para la vigilancia interna del edificio, ¿Hay al menos un vigilante por turno en los aledaños del centro de cálculo?
 - Respuesta: Si, pero sube a las otras 4 plantas cuando se le necesita.
 - Puntuación: 2

- ¿Hay salida de emergencia además de la habilitada para la entrada y salida de máquinas?
 - Respuesta: Si, pero existen cajas apiladas en dicha puerta. Algunas veces la quitan.
 - Puntuación: 3
- El personal de comunicaciones, ¿Puede entrar directamente en la Sala de Computadoras?
 - Respuesta: No, solo tiene tarjeta el Jefe de Comunicaciones. No se la da a su gente más que por causa muy justificada, y avisando casi siempre al jefe de explotación.
 - Puntuación: 4

El resultado sería el promedio de las puntuaciones: $(1 + 2 + 3 + 4) / 4 = 2,25$

Checklist binaria

Es la constituida por preguntas con respuesta única y excluyente: Si o No. Aritméticamente, equivalen a 1 (uno) o 0 (cero), respectivamente. Ejemplo: Se supone que se está realizando una revisión de los métodos de pruebas de programas en el ámbito de desarrollo de proyectos.

- ¿Existe Normativa de que el usuario final compruebe los resultados finales de los programas?
 - Puntuación: 1
- ¿Conoce el personal de desarrollo la existencia de la anterior normativa?
 - Puntuación: 1
- ¿Se aplica dicha norma en todos los casos?
 - Puntuación: 0
- ¿Existe una norma por la cual las pruebas han de realizarse con juegos de ensayo o copias de BD reales?
 - Puntuación: 0

Obsérvese como en este caso están contestadas las siguientes preguntas:

- ¿Se conoce la norma anterior?
 - Puntuación: 0
- ¿Se aplica en todos los casos?
 - Puntuación: 0

Las checklist de rango son adecuadas si el equipo auditor no es muy grande y mantiene criterios uniformes y equivalentes en las valoraciones. Permiten una mayor precisión en la evaluación que en la checklist binaria. Sin embargo, la bondad del método depende excesivamente de la formación y competencia del equipo auditor.

Las checklists binarias siguen una elaboración inicial mucho más ardua y compleja. Deben ser de gran precisión, como corresponde a la suma precisión de la respuesta. Una vez construidas, tienen la ventaja de exigir menos uniformidad del equipo auditor y el inconveniente genérico del <Si o No> frente a la mayor riqueza del intervalo.

No existen checklists estándar para todas y cada una de las instalaciones informáticas a auditar. Cada una de ellas posee peculiaridades que hacen necesarios los retoques de adaptación correspondientes en las preguntas a realizar.

Trazas y/o huellas

Con frecuencia, el auditor informático debe verificar que los programas, tanto de los sistemas como de usuario, realizan exactamente las funciones previstas, y no otras. Para ello se apoya en productos Software muy potentes y modulares que, entre otras funciones, rastrean los caminos que siguen los datos a través del programa.

Muy especialmente, estas “Trazas” se utilizan para comprobar la ejecución de las validaciones de datos previstas. Las mencionadas trazas no deben modificar en absoluto el sistema. Si la herramienta auditora produce incrementos apreciables de carga, se convendrá de antemano las fechas y horas más adecuadas para su empleo.

Por lo que se refiere al análisis del sistema, los auditores informáticos emplean productos que comprueban los valores asignados por técnica de sistemas a cada uno de los parámetros variables de las Librerías más importantes del mismo. Estos parámetros variables deben estar dentro de un intervalo marcado por el fabricante. A modo de ejemplo, algunas instalaciones descompensan el número de iniciadores de trabajos de determinados entornos o toman criterios especialmente restrictivos o permisivos en la asignación de unidades de servicio para según cuales tipos carga. Estas actuaciones, en principio útiles, pueden resultar contraproducentes si se traspasan los límites.

Observación

La observación es una de las técnicas más utilizadas en la recolección de información para aplicación de una auditoria, ya que a través de diferentes técnicas y métodos de observación permite recolectar directamente la información necesaria sobre el comportamiento del sistema, del área de sistemas, de las funciones, actividades y operaciones del equipo procesador o de cualquier otro hecho, acción o fenómeno del ámbito de sistemas. Existen diferentes tipos de observación, entre las cuales están:

- Observación directa.
- Observación indirecta.
- Observación oculta.
- Observación participativa.
- Observación no participativa.
- Introspección.
- Estrospección.
- Observación histórica.
- Observación controlada.
- Observación natural.

Inventarios

Esta forma de recopilación de información consiste en hacer un recuento físico de lo que se está auditando, consiste propiamente en comparar las cantidades reales existentes con las que debería haber para comprobar que sean iguales o, en caso contrario, para resaltar las posibles diferencias e investigar sus causas.

Los principales tipos de inventarios aplicables en el ambiente de sistemas computacionales, son:

- Inventario de software.
- Inventario de hardware.
- Inventario de documentos.
 - Inventario de documentos administrativos.
 - Manuales de la organización.
 - Manuales de procedimientos administrativos.
 - Manuales de perfil de puestos.
 - Otros manuales administrativos.
 - Inventario de documentos técnicos para el sistema.
 - Manuales e instructivos técnico del hardware, periféricos y componentes del sistema.
 - Manuales e instructivos de mantenimiento físico del sistema (hardware), entre otros.

Estándares de Auditoría

Para la realización y ejecución de una auditoría se hace necesario aplicar normas o estándares bajo los cuales las empresas deben regirse, de allí la importancia de identificar los estándares internacionales que en este caso, son:

Diretrices gerenciales de COBIT, desarrollado por la Information Systems Audit and Control Association (ISACA) Asociación de auditoria y control de los sistemas de información: las directrices generenciales son un marco internacional de referencias que abordan las mejores prácticas de auditoria y control de sistemas de información. Permiten que la gerencia incluya, comprenda y administre los riesgos relacionados con la tecnología de información y establezca el enlace entre los procesos de administración, aspectos técnicos, la necesidad de controles y los riesgos asociados. Uno de los objetivos de ISACA es promover estándares aplicables internacionalmente para cumplir con su visión. La estructura para los estándares de auditoria de SI brinda múltiples niveles de asesoría, como:

- Los auditores de SI respecto al nivel mínimo de desempeño aceptable requerido para cumplir con las responsabilidades profesionales indicadas en el código de ética Profesional de ISACA.
 - La dirección y otras partes interesadas en las expectativas de la profesión con respecto al trabajo de sus profesionales.
 - Los poseedores de la designación de auditor certificado de sistemas de información (Certified Information Systems Auditor, CISA) respecto a los requisitos que deben cumplir. El incumplimiento de estos estándares puede resultar en una investigación de la conducta del poseedor del certificado CISA por parte de la junta de directores de ISACA o del comité apropiado de ISACA y, en última instancia, en sanciones disciplinarias, así:
1. The Management of the Control of data Infromation Technology, desarrollado por el Instituto Canadiense de Contadores Certificados (CICA): Este modelo está basado en el concepto de roles y establece responsabilidades relacionadas con seguridad y los controles correspondientes. Dichos roles están clasificados con base en siete grupos: administración general, gerentes de sistemas, dueños, agentes, usuarios de sistemas de información, así como proveedores de servicios, desarrollo y operaciones de servicios y soporte de sistemas. Además, hace distinción entre los conceptos de autoridad, responsabilidad y responsabilidad respecto a control y riesgo previo al establecimiento del control, en términos de objetivos, estándares y técnicas mínimas a considerar.
 2. Administración de la inversión de tecnología de Inversión: un marco para la evaluación y mejora del proceso de madurez, desarrollado por la oficina de contabilidad general de los Estados Unidos (GAO): Este modelo identifica los procesos críticos, asegurando el éxito de las inversiones de tecnología de información y comunicación electrónicas. Además los organiza en cinco niveles de madurez, similar al modelo CMM.
 3. Estándares de administración de calidad y aseguramiento de calidad ISO 9000, desarrollados por la Organización Internacional de Estándares (ISO): La colección ISO 9000 es un conjunto de estándares y directrices que apoyan a las organizaciones a implementar sistemas de calidad efectivos, para el tipo de trabajo que ellos realizan.
 4. SysTrust - Principios y criterios de confiabilidad de sistemas, desarrollados por la Asociación de Contadores Públicos (AICPA) y el CICA: Este servicio pretende incrementar la confianza de la alta gerencia, clientes y socios, con respecto a la confiabilidad en los sistemas por una empresa o actividad en particular. Este modelo incluye elementos como: infraestructura, software de cualquier naturaleza, personal especializado y usuarios, procesos manuales y automatizados, y datos. El modelo persigue determinar si un sistema de información es confiable, (si un sistema

funciona sin errores significativos, o fallas durante un periodo de tiempo determinado bajo un ambiente dado).

5. Modelo de evolución de capacidades de software (CMM), desarrollado por el Instituto de Ingeniería de software (SEI): Este modelo hace posible evaluar las capacidades o habilidades para ejecutar, de una organización, con respecto al desarrollo y mantenimiento de sistemas de información. Consiste en 18 sectores clave, agrupados alrededor de cinco niveles de madurez. Se puede considerar que CMM es la base de los principios de evaluación recomendados por COBIT, así como para algunos de los procesos de administración de COBIT.
6. Administración de sistemas de información: una herramienta de evaluación práctica, desarrollado por la directiva de recursos de tecnología de información (ITRB): Este es una herramienta de evaluación que permite a entidades gubernamentales, comprender la implementación estratégica de tecnología de información y comunicación electrónica que puede apoyar su misión e incrementar sus productos y servicios.
7. Guía para el cuerpo de conocimientos de administración de proyectos, desarrollado por el comité de estándares del instituto de administración de proyectos: esta guía está enfocada en las mejores prácticas sobre administración de proyectos. Se refiere a aspectos sobre los diferentes elementos necesarios para una administración exitosa de proyectos de cualquier naturaleza. En forma precisa, este documento identifica y describe las prácticas generalmente aceptadas de administración de proyectos que pueden ser implementadas en las organizaciones.
8. Ingeniería de seguridad de sistemas - Modelo de madurez de capacidades (SSE - CMM), desarrollado por la agencia de seguridad nacional (NSA) con el apoyo de la Universidad de Carnegie Mellon: Este modelo describe las características esenciales de una arquitectura de seguridad organizacional para tecnología de información y comunicación electrónica, de acuerdo con las prácticas generalmente aceptadas observadas en las organizaciones.
9. Administración de seguridad de información: aprendiendo de organizaciones líderes, desarrollado por la oficina de contabilidad general de los Estados Unidos (GAO): este modelo considera ocho organizaciones privadas reconocidas como líderes respecto a seguridad en cómputo. Este trabajo hace posible la identificación de 16 prácticas necesarias para asegurar una adecuada administración de la seguridad de cómputo, las cuales deben ser suficientes para incrementar significativamente el nivel de administración de seguridad en tecnología de información y comunicación electrónica.

El Modelo COBIT para auditoria y control de sistemas de información

La evaluación de los requerimientos del negocio, los recursos y procesos IT, son puntos bastante importantes para el buen funcionamiento de una compañía y para el aseguramiento de su supervivencia en el mercado. COBIT es precisamente un modelo para auditar la gestión y control de los sistemas de información y tecnología, orientado a todos los sectores de una organización, es decir, administradores IT, usuarios y por supuesto, los auditores involucrados en el proceso.

Las siglas COBIT significan Objetivos de Control para Tecnología de Información (Control Objectives for Information Systems and related Technology). El modelo es el resultado de una investigación con expertos de varios países, desarrollado por ISACA (Information Systems Audit and Control Association).

COBIT, lanzado en 1996, es una herramienta de gobierno de TI que ha cambiado la forma en que trabajan los profesionales de tecnología. Vinculando tecnología informática y prácticas de control, el modelo COBIT consolida y armoniza estándares de fuentes globales prominentes en un recurso crítico para la gerencia, los profesionales de control y los auditores.

La estructura del modelo COBIT propone un marco de acción donde se evalúan los criterios de información, como por ejemplo la seguridad y calidad, se auditán los recursos que comprenden la tecnología de información, como por ejemplo el recurso humano, instalaciones, sistemas, entre otros, y finalmente se realiza una evaluación sobre los procesos involucrados en la organización.

“La adecuada implementación de COBIT en una organización, provee una herramienta automatizada, para evaluar de manera ágil y consistente el cumplimiento de los objetivos de control y controles detallados, que aseguran que los procesos y recursos de información y tecnología contribuyen al logro de los objetivos del negocio en un mercado cada vez más exigente, complejo y diversificado. Cualquier tipo de empresa puede adoptar una metodología COBIT, como parte de un proceso de reingeniería en aras de reducir los índices de incertidumbre sobre vulnerabilidades y riesgos de los recursos IT y consecuentemente, sobre la posibilidad de evaluar el logro de los objetivos del negocio apalancado en procesos tecnológicos”, señaló un informe de ETEK.

COBIT se aplica a los sistemas de información de toda la empresa, incluyendo los computadores personales y las redes. Está basado en la filosofía de que los recursos TI necesitan ser administrados por un conjunto de procesos naturalmente agrupados para proveer la información pertinente y confiable que requiere una organización para lograr sus objetivos.

Criterios de información de COBIT

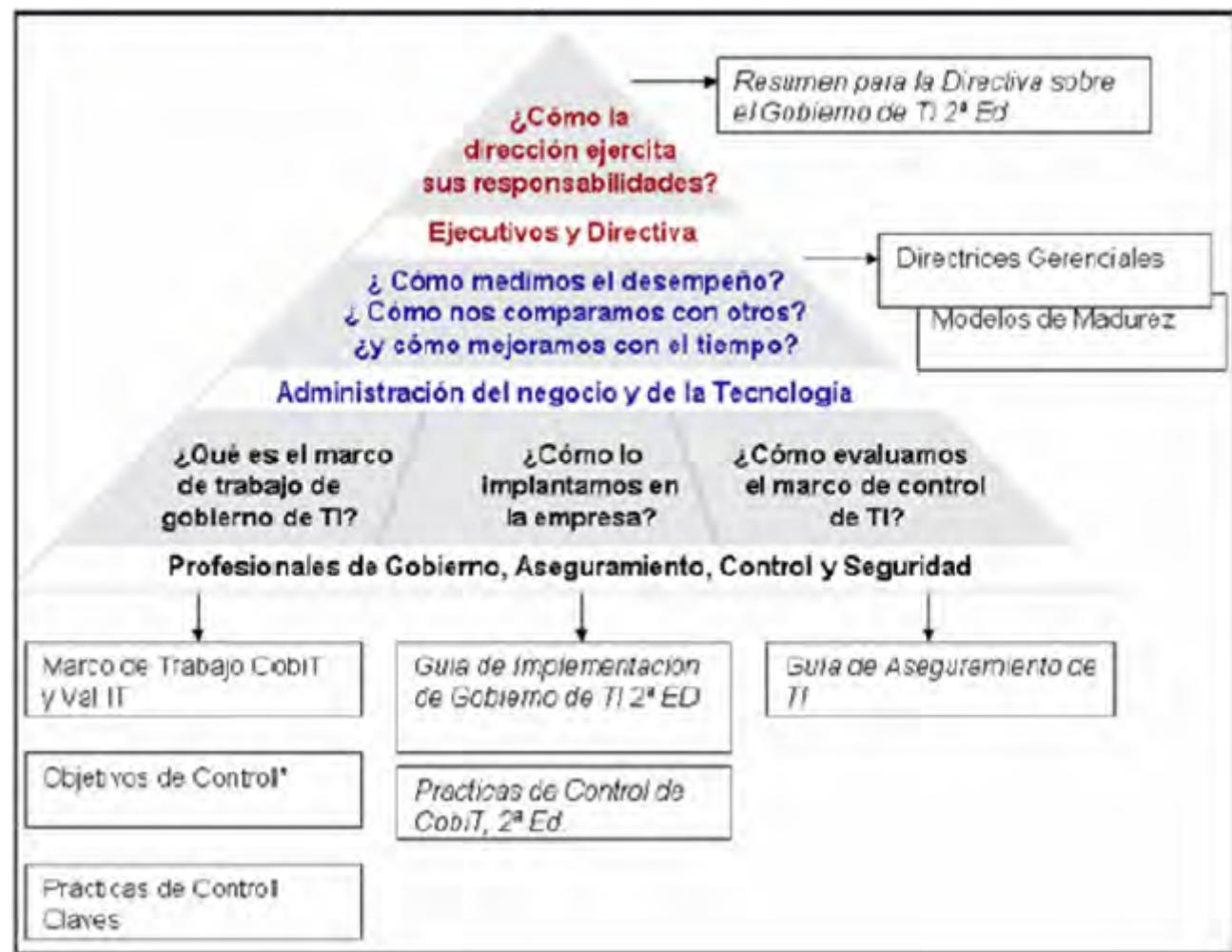
Para satisfacer los objetivos del negocio, la información necesita adaptarse a ciertos criterios de control, los cuales son referidos en COBIT como requerimientos de información del negocio. Con base en los requerimientos más amplios de calidad, fiduciarios y de seguridad, se definieron los siguientes siete criterios de información:

- **La efectividad** tienen que ver con que la información sea relevante y pertinente a los procesos del negocio, y se propone de una manera oportuna, correcta, consistente y utilizable.
- **La eficiencia** consiste en que la información sea generada con el óptimo (más productivo y económico) uso de los recursos.
- **La confidencialidad** se refiere a la protección de información sensitiva contra revelación no autorizada.
- **La integridad** está relacionada con la precisión y completitud de la información, así como con su validez de acuerdo a los valores y expectativas del negocio.
- **La disponibilidad** se refiere a que la información esté disponible cuando sea requerida por los procesos del negocio en cualquier momento. También concierne a la protección de los recursos y las capacidades necesarias asociadas.
- **El cumplimiento** tienen que ver con acatar aquellas leyes, reglamentos y acuerdos contractuales a los cuales está sujeto el proceso de negocios, es decir, criterios de negocios impuestos externamente, así como políticas internas.
- **La confiabilidad** se refiere a proporcionar la información apropiada para que la gerencia administre la entidad y ejerza sus responsabilidades fiduciarias y de gobierno.

Los productos COBIT se han organizado en tres niveles, diseñados para dar soporte a lo siguiente:

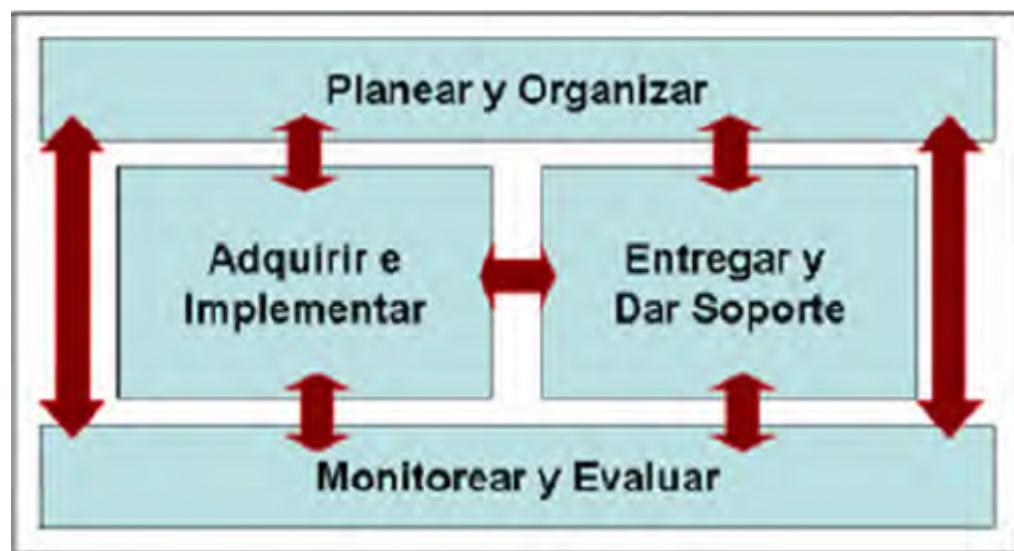
- Administración y consejos ejecutivos.
- Administración del negocio y de TI.
- Profesionales en Gobierno, aseguramiento, control y seguridad.

Figura de Diagrama de Contenido del COBIT:



El diagrama de contenido de COBIT mostrado presenta las audiencias principales, sus preguntas sobre gobierno TI y los productos que generalmente les aplican para proporcionar las respuestas. También hay productos derivados para propósitos específicos, para dominios tales como seguridad o empresas específicas.

El conjunto de lineamientos y estándares internacionales conocidos como COBIT, define un marco de referencia que clasifica los procesos de las unidades de tecnología de información de las organizaciones en cuatro “dominios” principales, a saber:



- **Planear y organizar (PO)** : Proporciona dirección para la entrega de soluciones (AI) y la entrega de servicios (DS).
- **Adquirir e implementar (AI)** : Proporciona las soluciones y las pasa para convertirlas en servicios.

- **Entregar y dar soporte (DS)** : Recibe las soluciones y las hace utilizables por los usuarios finales.
- **Monitorear y evaluar (ME)** : Monitorear todos los procesos para asegurar que se sigue la dirección provista.

Planificación y organización PO

- PO1 Definir un plan estratégico de TI
- PO2 Definir la arquitectura de la información
- PO3 Determinar la dirección tecnológica
- PO4 Definir los procesos, organización y relaciones de TI
- PO5 Administrar la inversión en TI
- PO6 Comunicar las aspiraciones y la dirección de la gerencia
- PO7 Administrar recursos humanos de TI
- PO8 Administrar la calidad
- PO9 Evaluar y administrar los riesgos de TI
- PO10 Administrar proyectos

Adquisición e implantación AI

- AI1 Identificar soluciones automatizadas
- AI2 Adquirir y mantener software aplicativo
- AI3 Adquirir y mantener infraestructura tecnológica
- AI4 Facilitar la operación y el uso
- AI5 Adquirir recursos de TI
- AI6 Administrar cambios
- AI7 Instalar y acreditar soluciones y cambios

Sopor te y Servicios DS

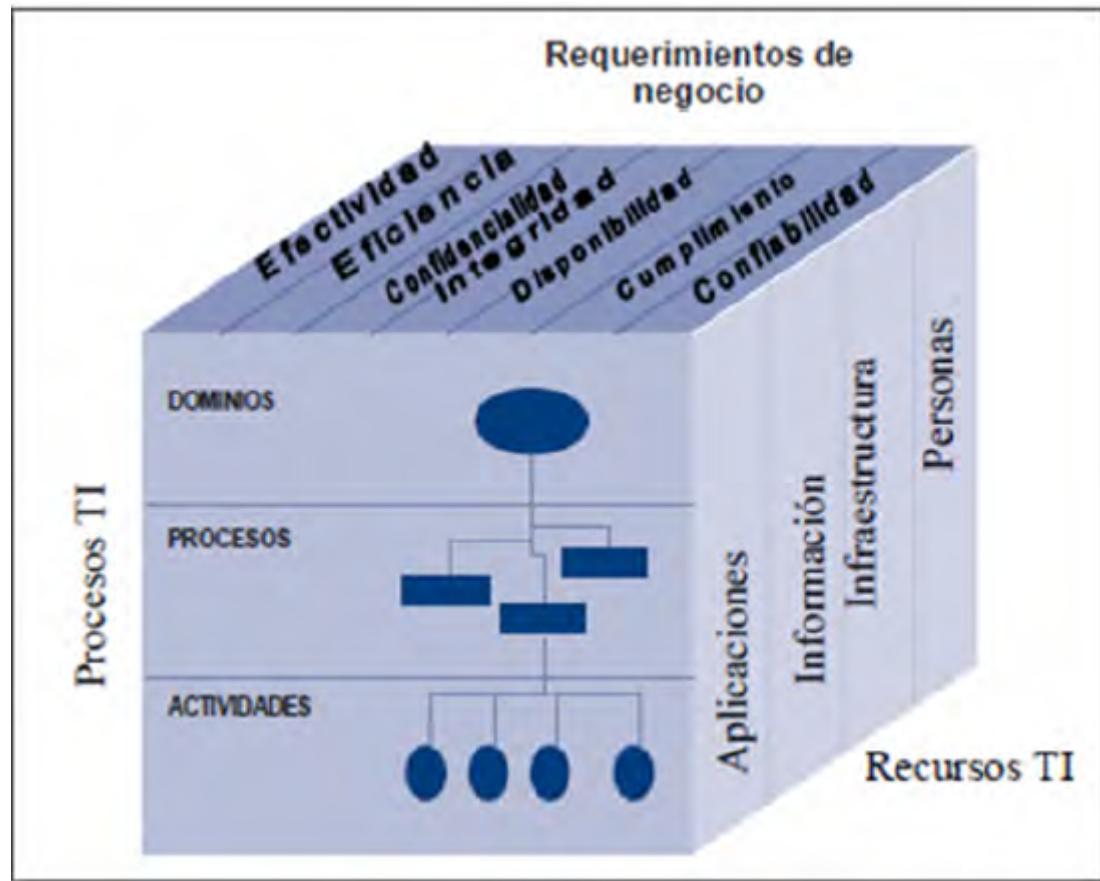
- DS1 Definir y administrar los niveles de servicio
- DS2 Administrar los servicios de terceros
- DS3 Administrar el desempeño y la capacidad
- DS4 Garantizar la continuidad del servicio
- DS5 Garantizar la seguridad de los sistemas
- DS6 Identificar y asignar costos
- DS7 Educar y entrenar a los usuarios
- DS8 Administrar la mesa de servicio y los incidentes
- DS9 Administrar la configuración
- DS10 Administrar los problemas
- DS11 Administrar los datos
- DS12 Administrar el ambiente físico
- DS13 Administrar las operaciones

Monitoreo y evaluación ME

- ME1 Monitorear y evaluar el desempeño de TI
- ME2 Monitorear y evaluar el control interno
- ME3 Garantizar el cumplimiento regulatorio
- ME4 Proporcionar gobierno de TI

Estos dominios agrupan objetivos de control de alto nivel que cubren tanto los aspectos de información, como de la tecnología que la respalda. Estos dominios y objetivos de control facilitan que la generación y procesamiento de la información cumplan con las características de efectividad, eficiencia, confidencialidad, integridad, disponibilidad, cumplimiento y confiabilidad.

El cubo de COBIT:



Asimismo, se debe tomar en cuenta los recursos que proporciona la tecnología de información, tales como: datos, aplicaciones, plataformas tecnológicas, instalaciones y recurso humano.

Dominios de COBIT

Entendiéndose como dominio, la agrupación natural de procesos, normalmente corresponden a un dominio o una responsabilidad organizacional, los procesos a su vez son conjuntos o series de actividades unidas con delimitación o cortes de control y las actividades son acciones requeridas para lograr un resultado medible.

COBIT proporciona una lista completa de procesos que puede ser utilizada para verificar que se completan las actividades y responsabilidades; sin embargo, no es necesario que apliquen todas, y, aun más, se pueden combinar como se necesite por cada empresa.

Dominio: Planificación y organización (PO)

Este dominio cubre las estrategias y tácticas, y se refiere a la identificación de la forma en que la tecnología de información puede contribuir de la mejor manera al logro de los objetivos de negocio. Además, la consecución de la visión estratégica necesita ser planeada, comunicada y administrada desde diferentes perspectivas. Finalmente, deberán establecerse una organización y una infraestructura tecnológica apropiadas.

Procesos:

PO1 Definición de un plan estratégico

Objetivo: Lograr un balance óptimo entre las oportunidades de tecnología de información y los requerimientos de TI de negocio, para asegurar sus logros futuros.

Su realización se concreta a través de un proceso de planeación estratégica emprendido en intervalos regulares dando lugar a planes a largo plazo, los que deberán ser traducidos

periódicamente en planes operacionales estableciendo metas claras y concretas a corto plazo, teniendo en cuenta:

- La definición de objetivos de negocio y necesidades de TI, la alta gerencia será la responsable de desarrollar e implementar planes a largo y corto plazo que satisfagan la misión y las metas generales de la organización.
- El inventario de soluciones tecnológicas e infraestructura actual, deberá evaluar los sistemas existentes en términos de: nivel de automatización de negocio, funcionalidad, estabilidad, complejidad, costo y fortalezas y debilidades, con el propósito de determinar el nivel de soporte que reciben los requerimientos del negocio de los sistemas existentes.
- Los cambios organizacionales, se deberá asegurar que se establezca un proceso para modificar oportunamente y con precisión el plan a largo plazo de tecnología de información con el fin de adaptar los cambios al plan a largo plazo de la organización y los cambios en las condiciones de la TI.
- Estudios de factibilidad oportunos, para que se puedan obtener resultados efectivos.

PO2 Definición de la arquitectura de información

Objetivo: satisfacer los requerimientos de negocio, organizando de la mejor manera posible los sistemas de información, a través de la creación y mantenimiento de un modelo de información de negocio, asegurándose que se definan los sistemas apropiados para optimizar la utilización de esta información, tomando en consideración:

- La documentación deberá conservar consistencia con las necesidades permitiendo a los responsables llevar a cabo sus tareas eficiente y oportunamente.
- El diccionario de datos, el cual incorpora las reglas de sintaxis de datos de la organización deberá ser continuamente actualizado.
- La propiedad de la información y la clasificación de severidad con el que se establecerá un marco de referencia de clasificación general relativo a la ubicación de datos en clases de información.

PO3 Determinación de la dirección tecnológica

Objetivo: aprovechar al máximo la tecnología disponible o tecnología emergente, satisfaciendo los requerimientos de negocio, a través de la creación y mantenimiento de un plan de infraestructura tecnológica, tomando en consideración:

- La capacidad de adecuación y evolución de la infraestructura actual, que deberá concordar con los planes a largo y corto plazo de tecnología de información y debiendo abarcar aspectos tales como arquitectura de sistemas, dirección tecnológica y estrategias de migración.
- El monitoreo de desarrollos tecnológicos que serán tomados en consideración durante el desarrollo y mantenimiento del plan de infraestructura tecnológica.
- Las contingencias (por ejemplo, redundancia, resistencia, capacidad de adecuación y evolución de la infraestructura), con lo que se evaluará sistemáticamente el plan de infraestructura tecnológica.
- Planes de adquisición, los cuales deberán reflejar las necesidades identificadas en el plan de infraestructura tecnológica.

PO4 Definición de la organización y de las relaciones de TI

Objetivo: prestación de servicios de TI.

Esto se realiza por medio de una organización conveniente en número y habilidades, con tareas y responsabilidades definidas y comunicadas, teniendo en cuenta:

- El comité de dirección el cual se encargará de vigilar la función de servicios de información y sus actividades.

- Propiedad, custodia, la gerencia deberá crear una estructura para designar formalmente a los propietarios y custodios de los datos. Sus funciones y responsabilidades deberán estar claramente definidas.
- Supervisión, para asegurar que las funciones y responsabilidades sean llevadas a cabo apropiadamente.
- Segregación de funciones, con la que se evitará la posibilidad de que un solo individuo resuelva un proceso crítico.
- Los roles y responsabilidades, la gerencia deberá asegurarse de que todo el personal deberá conocer y contar con la autoridad suficiente para llevar a cabo las funciones y responsabilidades que le hayan sido asignadas.
- La descripción de puestos, deberá delinear claramente tanto la responsabilidad como la autoridad, incluyendo las definiciones de las habilidades y la experiencia necesarias para el puesto, y ser adecuadas para su utilización en evaluaciones de desempeño.
- Los niveles de asignación de personal, deberán hacerse evaluaciones de requerimientos regularmente para asegurar una asignación de personal adecuada en el presente y en el futuro.
- El personal clave, la gerencia deberá definir e identificar al personal clave de tecnología de información.

PO5 Manejo de la inversión

Objetivo: tiene como finalidad la satisfacción de los requerimientos de negocio, asegurando el financiamiento y el control de desembolsos de recursos financieros. Su realización se concreta a través de presupuestos periódicos sobre inversiones y operaciones establecidas y aprobados por el negocio, teniendo en cuenta:

- Las alternativas de financiamiento, se deberán investigar diferentes alternativas de financiamiento.
- El control del gasto real, se deberá tomar como base el sistema de contabilidad de la organización, mismo que deberá registrar, procesar y reportar rutinariamente los costos asociados con las actividades de la función de servicios de información.
- La justificación de costos y beneficios, deberá establecerse un control gerencial que garantice que la prestación de servicios por parte de la función de servicios de información se justifique en cuanto a costos. Los beneficios derivados de las actividades de TI deberán ser analizados en forma similar.

PO6 Comunicación de la dirección y aspiraciones de la gerencia

Objetivo: Asegura el conocimiento y comprensión de los usuarios sobre las aspiraciones de alto nivel (gerencia), se concreta a través de políticas establecidas y transmitidas a la comunidad de usuarios, necesitándose para esto estándares para traducir las opciones estratégicas en reglas de usuario prácticas y utilizables. Toma en cuenta:

- Los código de ética/conducta, el cumplimiento de las reglas de ética, conducta, seguridad y estándares de control interno deberá ser establecido y promovido por la Alta Gerencia.
- Las directrices tecnológicas.
- El cumplimiento, la Gerencia deberá también asegurar y monitorear la duración de la implementación de sus políticas.
- El compromiso con calidad, la gerencia de la función de servicios de información deberá definir, documentar y mantener una filosofía de calidad, debiendo ser comprendidos, implementados y mantenidos por todos los niveles de la función de servicios de información.
- Las políticas de seguridad y control interno, la alta gerencia deberá asegurar que esta política de seguridad y de control interno especifique el propósito y los objetivos, la estructura gerencial, el alcance dentro de la organización, la definición y asignación de responsabilidades para su implementación a todos los niveles y la

definición de multas y de acciones disciplinarias asociadas con la falta de cumplimiento de estas políticas.

PO7 Administración de recursos humanos

Objetivo: Maximizar las contribuciones del personal a los procesos de TI, satisfaciendo así los requerimientos de negocio, a través de técnicas sólidas para administración de personal, tomando en consideración:

- El reclutamiento y promoción, deberá tener como base criterios objetivos, considerando factores como la educación, la experiencia y la responsabilidad.
- Los requerimientos de calificaciones, el personal deberá estar calificado, tomando como base una educación, entrenamiento y/o experiencia apropiados, según se requiera.
- La capacitación, los programas de educación y entrenamiento estarán dirigidos a incrementar los niveles de habilidad técnica y administrativa del personal.
- La evaluación objetiva y medible del desempeño, se deberá asegurar que dichas evaluaciones sean llevada a cabo regularmente según los estándares establecidos y las responsabilidades específicas del puesto. Los empleados deberán recibir asesoría sobre su desempeño o su conducta cuando esto sea apropiado.

PO8 Asegurar el cumplimiento con los requerimientos externos

Objetivo: Cumplir con obligaciones legales, regulatorias y contractuales.

Para ello se realiza una identificación y análisis de los requerimientos externos en cuanto a su impacto en TI, llevando a cabo las medidas apropiadas para cumplir con ellos y se toma en consideración:

- Definición y mantenimiento de procedimientos para la revisión de requerimientos externos, para la coordinación de estas actividades y para el cumplimiento continuo de los mismos.
- Leyes, regulaciones y contratos.
- Revisiones regulares en cuanto a cambios.
- Búsqueda de asistencia legal y modificaciones.
- Seguridad y ergonomía con respecto al ambiente de trabajo de los usuarios y el personal de la función de servicios de información.
- Privacidad.
- Propiedad intelectual.
- Flujo de datos externos y criptografía.

PO9 Evaluación de riesgos

Objetivo: Asegurar el logro de los objetivos de TI y responder a las amenazas hacia la provisión de servicios de TI.

Para ello se logra la participación de la propia organización en la identificación de riesgos de TI y en el análisis de impacto, tomando medidas económicas para mitigar los riesgos y se toma en consideración:

- Identificación, definición y actualización regular de los diferentes tipos de riesgos de TI (por ej: tecnológicos, de seguridad, etc.) de manera que se pueda determinar la manera en la que los riesgos deben ser manejados a un nivel aceptable.
- Definición de alcances, límites de los riesgos y la metodología para las evaluaciones de los riesgos.
- Actualización de evaluación de riesgos.
- Metodología de evaluación de riesgos.
- Medición de riesgos cualitativos y/o cuantitativos.

- Definición de un plan de acción contra los riesgos para asegurar que existan controles y medidas de seguridad económicas que mitiguen los riesgos en forma continua.
- Aceptación de riesgos dependiendo de la identificación y la medición del riesgo, de la política organizacional, de la incertidumbre incorporada al enfoque de evaluación de riesgos y de que tan económico resulte implementar protecciones y controles.

PO10 Administración de proyectos

Objetivo: Establecer prioridades y entregar servicios oportunamente y de acuerdo al presupuesto de inversión.

Para ello se realiza una identificación y priorización de los proyectos en línea con el plan operacional por parte de la misma organización. Además, la organización deberá adoptar y aplicar sólidas técnicas de administración de proyectos para cada proyecto emprendido y se toma en consideración:

- Definición de un marco de referencia general para la administración de proyectos que defina el alcance y los límites del mismo, así como la metodología de administración de proyectos a ser adoptada y aplicada para cada proyecto emprendido. La metodología deberá cubrir, como mínimo, la asignación de responsabilidades, la determinación de tareas, la realización de presupuestos de tiempo y recursos, los avances, los puntos de revisión y las aprobaciones.
- El involucramiento de los usuarios en el desarrollo, implementación o modificación de los proyectos.
- Asignación de responsabilidades y autoridades a los miembros del personal asignados al proyecto.
- Aprobación de fases de proyecto por parte de los usuarios antes de pasar a la siguiente fase.
- Presupuestos de costos y horas hombre.
- Planes y metodologías de aseguramiento de calidad que sean revisados y acordados por las partes interesadas.
- Plan de administración de riesgos para eliminar o minimizar los riesgos.
- Planes de prueba, entrenamiento, revisión post-implementación.

PO11 Administración de calidad

Objetivo: satisfacer los requerimientos del cliente.

Para ello se realiza una planeación, implementación y mantenimiento de estándares y sistemas de administración de calidad por parte de la organización y se toma en consideración:

- Definición y mantenimiento regular del plan de calidad, el cual deberá promover la filosofía de mejora continua y contestar a las preguntas básicas de qué, quién y cómo.
- Responsabilidades de aseguramiento de calidad que determine los tipos de actividades de aseguramiento de calidad tales como revisiones, auditorias, inspecciones, etc. que deben realizarse para alcanzar los objetivos del plan general de calidad.
- Metodologías del ciclo de vida de desarrollo de sistemas que rija el proceso de desarrollo, adquisición, implementación y mantenimiento de sistemas de información.
- Documentación de pruebas de sistemas y programas.
- Revisiones y reportes de aseguramiento de calidad.

Dominio: Adquisición e implementación (AI)

Para llevar a cabo la estrategia de TI, las soluciones de TI deben ser identificadas, desarrolladas o adquiridas, así como implementadas e integradas dentro del proceso del

negocio. Además, este dominio cubre los cambios y el mantenimiento realizados a sistemas existentes.

Procesos:

AI1 Identificación de soluciones automatizadas

Objetivo: Asegurar el mejor enfoque para cumplir con los requerimientos del usuario.

Para ello se realiza un análisis claro de las oportunidades alternativas comparadas contra los requerimientos de los usuarios y toma en consideración:

- Definición de requerimientos de información para poder aprobar un proyecto de desarrollo.
- Estudios de factibilidad con la finalidad de satisfacer los requerimientos del negocio establecidos para el desarrollo de un proyecto.
- Arquitectura de información para tener en consideración el modelo de datos al definir soluciones y analizar la factibilidad de las mismas.
- Seguridad con relación de costo-beneficio favorable para controlar que los costos no excedan los beneficios.
- Pistas de auditoria para ello deben existir mecanismos adecuados. Dichos mecanismos deben proporcionar la capacidad de proteger datos sensativos (ej. Identificación de usuarios contra divulgación o mal uso).
- Contratación de terceros con el objeto de adquirir productos con buena calidad y excelente estado.
- Aceptación de instalaciones y tecnología a través del contrato con el proveedor donde se acuerda un plan de aceptación para las instalaciones y tecnología específica a ser proporcionada.

AI2 Adquisición y mantenimiento del software aplicativo

Objetivo: Proporcionar funciones automatizadas que soporten efectivamente al negocio.

Para ello se definen declaraciones específicas sobre requerimientos funcionales y operacionales y una implementación estructurada con entregables claros y se toma en consideración:

- Requerimientos de usuarios, para realizar un correcto análisis y obtener un software claro y fácil de usar.
- Requerimientos de archivo, entrada, proceso y salida.
- Interface usuario-máquina asegurando que el software sea fácil de utilizar y que sea capaz de auto documentarse.
- Personalización de paquetes.
- Realizar pruebas funcionales (unitarias, de aplicación, de integración y de carga y estrés), de acuerdo con el plan de prueba del proyecto y con los estándares establecidos antes de ser aprobado por los usuarios.
- Controles de aplicación y requerimientos funcionales.
- Documentación (materiales de consulta y soporte para usuarios) con el objeto de que los usuarios puedan aprender a utilizar el sistema o puedan sacarse todas aquellas inquietudes que se les puedan presentar.

AI3 Adquisición y mantenimiento de la infraestructura tecnológica

Objetivo: proporcionar las plataformas apropiadas para soportar aplicaciones de negocios.

Para ello se realizará una evaluación del desempeño del hardware y software, la provisión de mantenimiento preventivo de hardware y la instalación, seguridad y control del software del sistema y toma en consideración:

- Evaluación de tecnología para identificar el impacto del nuevo hardware o software sobre el rendimiento del sistema general.
- Mantenimiento preventivo del hardware con el objeto de reducir la frecuencia y el impacto de fallas de rendimiento.
- Seguridad del software de sistema, instalación y mantenimiento para no arriesgar la seguridad de los datos y programas ya almacenados en el mismo.

AI4 Desarrollo y mantenimiento de procedimientos

Objetivo: Asegurar el uso apropiado de las aplicaciones y de las soluciones tecnológicas establecidas.

Para ello se realiza un enfoque estructurado del desarrollo de manuales de procedimientos de operaciones para usuarios, requerimientos de servicio y material de entrenamiento y toma en consideración:

- Manuales de procedimientos de usuarios y controles, de manera que los mismos permanezcan en permanente actualización para el mejor desempeño y control de los usuarios.
- Manuales de operaciones y controles, de manera que estén en permanente actualización.
- Materiales de entrenamiento enfocados al suso del sistema en la práctica diaria.

AI5 Instalación y aceptación de los sistemas

Objetivo: Verificar y confirmar que la solución sea adecuada para el propósito deseado.

Para ello se realiza una migración de instalación, conversión y plan de aceptaciones adecuadamente formalizadas y toma en consideración:

- Capacitación del personal de acuerdo al plan de entrenamiento definido y los materiales relacionados.
- Conversión / carga de datos, de manera que los elementos necesarios del sistema anterior sean convertidos al sistema nuevo.
- Pruebas específicas (cambios, desempeño, aceptación final, operacional) con el objeto de obtener un producto satisfactorio.
- Acreditación de manera que la Gerencia de operaciones y usuaria acepten los resultados de las pruebas y el nivel de seguridad para los sistemas, junto con el riesgo residual existente.
- Revisiones post implementación con el objeto de reportar si el sistema proporcionó los beneficios esperados de la manera más económica.

AI6 Administración de los cambios

Objetivo: minimizar la probabilidad de interrupciones, alteraciones no autorizadas y errores.

Esto se hace posible a través de un sistema de administración que permita el análisis, implementación y seguimiento de todos los cambios requeridos y llevados a cabo a la infraestructura de TI actual y toma en consideración:

- Identificación de cambios tanto internos como por parte de proveedores.
- Procedimientos de categorización, priorización y emergencia de solicitudes de cambios.
- Evaluación del impacto que provocaran los cambios.

- Autorización de cambios.
- Manejo deliberación de manera que la liberación de software esté regida por procedimientos formales asegurando aprobación, empaque, pruebas de regresión, entrega, etc.
- Distribución de software, estableciendo medidas de control específicas para asegurar la distribución de software correcto al lugar correcto, con integridad y de manera oportuna.

Dominio: Entregar y dar soporte (DS)

En este dominio se hace referencia a la entrega de los servicios requeridos, que abarca desde las operaciones tradicionales hasta el entrenamiento, pasando por seguridad y aspectos de continuidad. Con el fin de proveer servicios, deberán establecerse los procesos de soporte necesarios. Este dominio incluye el procesamiento de los datos por sistemas de aplicación, frecuentemente clasificados como controles de aplicación.

Procesos:

DS1 Definición de niveles de servicio

Objetivo: Establecer una comprensión común del nivel de servicio requerido.

Para ello se establecen convenios de niveles de servicio que formalicen los criterios de desempeño contra los cuales se medirá la cantidad y la calidad del servicio y se toma en consideración:

- Convenios formales que determinen la disponibilidad, confiabilidad, desempeño, capacidad de crecimiento, niveles de soporte proporcionados al usuario, plan de contingencia / recuperación, nivel mínimo aceptable de funcionalidad del sistema satisfactoriamente liberado, restricciones (límites en la cantidad de trabajo), cargos por servicio, instalaciones de impresión central (disponibilidad), distribución de impresión central y procedimiento de cambio.
- Definición de las responsabilidades de los usuarios y de la función de servicios de información.
- Procedimientos de desempeño que aseguren que la manera y las responsabilidades sobre las relaciones que rigen el desempeño entre todas las partes involucradas sean establecidas, coordinadas, mantenidas y comunicadas a todos los departamentos afectados.
- Definición de dependencias asignando un Gerente de nivel de Servicio que sea responsable de monitorear y reportar los alcances de los criterios de desempeño del servicio especificado y todos los problemas encontrados durante el procesamiento.
- Provisiones para elementos sujetos a cargos en los acuerdos de niveles de servicio para hacer posibles comparaciones y decisiones de niveles de servicios contra su costo.
- Garantías de integridad.
- Convenios de confidencialidad.
- Implementación de un programa de mejoramiento del servicio.

DS2 Administración de servicios prestados por terceros

Objetivo: Asegurar que las tareas y responsabilidades de las terceras partes estén claramente definidas, que cumplan y continúen satisfaciendo los requerimientos.

Para ello se establecen medidas de control dirigidas a la revisión y monitoreo de contratos y procedimientos existentes, en cuanto a su efectividad y suficiencia, con respecto a las políticas de la organización y toma en consideración:

- Acuerdos de servicios con terceras partes a través de contratos entre la organización y el proveedor de las administración de instalaciones esté basado en niveles de

procesamiento requeridos, seguridad, monitoreo y requerimientos de contingencia, así como en otras estipulaciones según sea apropiado.

- Acuerdos de confidencialidad. Además, se deberá calificar a los terceros y el contrato deberá definirse y acordarse para cada relación de servicio con un proveedor.
- Requerimientos legales regulatorios de manera de asegurar que estos concuerde con los acuerdos de seguridad identificados, declarados y acordados.
- Monitoreo de la entrega de servicio con el fin de asegurar el cumplimiento de los acuerdos del contrato.

DS3 Administración de desempeño y capacidad

Objetivo: Asegurar que la capacidad adecuada está disponible y que se esté haciendo el mejor uso de ella para alcanzar el desempeño deseado.

Para ello se realizan controles de manejo de capacidad y desempeño que recopilen datos y reporten acerca del manejo de cargas de trabajo, tamaño de aplicaciones, manejo y demanda de recursos y toma en consideración:

- Requerimientos de disponibilidad y desempeño de los servicios de sistemas de información.
- Monitoreo y reporte de los recursos de tecnología de información.
- Utilizar herramientas de modelado apropiadas para producir un modelo del sistema actual para apoyar el pronóstico de los requerimientos de capacidad, confiabilidad de configuración, desempeño y disponibilidad.
- Administración de capacidad estableciendo un proceso de planeación para la revisión del desempeño y capacidad de hardware con el fin de asegurar que siempre exista una capacidad justificable económicamente para procesar cargas de trabajo con cantidad y calidad de desempeño.
- Prevenir que se pierda la disponibilidad de recursos mediante la implementación de mecanismos de tolerancia de fallas, de asignación de recursos y de prioridad de tareas.

DS4 Asegurar el servicio continuo

Objetivo: mantener el servicio disponible de acuerdo con los requerimientos y continuar su provisión en caso de interrupciones.

Para ello se tiene un plan de continuidad probado y funcional, que esté alineado con el plan de continuidad del negocio y relacionado con los requerimientos de negocio y toma en consideración:

- Planificación de severidad.
- Plan documentado.
- Procedimientos alternativos.
- Respaldo y recuperación.
- Pruebas y entrenamiento sistemático y singulares.

DS5 Garantizar la seguridad de sistemas

Objetivo: salvaguardar la información contra uso no autorizados, divulgación, modificación, daño o pérdida.

Para ello se realizan controles de acceso lógico que aseguren que el acceso a sistemas, datos y programas está restringido a usuarios autorizados y toma en consideración:

- Autorización, autenticación y el acceso lógico junto con el uso de los recursos de TI deberá restringirse a través de la instrumentación de mecanismos de autenticación de usuarios identificados y recursos asociados con las reglas de acceso.

- Perfiles e identificación de usuarios estableciendo procedimientos para asegurar acciones oportunas relacionadas con la requisición, establecimiento, emisión, suspensión y suspensión de cuentas de usuario.
- Administración de llaves criptográficas definiendo e implementando procedimientos y protocolos a ser utilizados en la generación, distribución, certificación, almacenamiento, entrada, utilización y archivo de llaves criptográficas con el fin de asegurar la protección de las mismas.
- Manejo, reporte y seguimiento de incidentes implementando capacidad para la atención de los mismos.
- Prevención y detección de virus tales como Caballos de Troya, estableciendo adecuadas medidas de control preventivas, detectivas y correctivas.
- Utilización de Firewalls si existe una conexión con Internet u otras redes públicas en la organización.

DS6 Educación y entrenamiento de usuarios

Objetivo: Asegurar que los usuarios estén haciendo un uso efectivo de la tecnología y estén conscientes de los riesgos y responsabilidades involucrados.

Para ello se realiza un plan completo de entrenamiento y desarrollo y se toma en consideración:

- Currículo de entrenamiento estableciendo y manteniendo procedimientos para identificar y documentar las necesidades de entrenamiento de todo el personal que haga uso de los servicios de información.
- Campañas de concienciación, definiendo los grupos objetivos, identificar y asignar entrenadores y organizar oportunamente las sesiones de entrenamiento.
- Técnicas de concienciación proporcionando un programa de educación y entrenamiento que incluya conducta ética de la función de servicios de información.

DS7 Identificación y asignación de costos

Objetivo: asegurar un conocimiento concreto de los costos atribuibles a los servicios de TI.

Para ello se realiza un sistema de contabilidad de costos que asegure que éstos sean registrados, calculados y asignados a los niveles de detalle requeridos y toma en consideración:

- Los elementos sujetos a cargos deben ser recursos identificables, medibles y predecibles para los usuarios.
- Procedimientos y políticas de cargo que fomenten el uso apropiado de los recursos de cómputo y aseguren el trato justo de los departamentos usuarios y sus necesidades.
- Tarifas definiendo e implementando procedimientos de costeo de prestar servicios, para ser analizados, monitoreados, evaluados asegurando al mismo tiempo la economía.

DS8 Apoyo y asistencia a los clientes de TI

Objetivo: asegurar que cualquier problema experimentado por los usuarios sea atendido apropiadamente.

Para ello se realiza un buró de ayuda que proporcione soporte y asesoría de primera línea y toma en consideración:

- Consultas de usuarios y respuesta a problemas estableciendo un soporte de una función de buró de ayuda.
- Monitoreo de consultas y despacho estableciendo procedimientos que aseguren que las preguntas de los clientes que pueden ser resueltas sean reasignadas al nivel adecuado para atenderlas.

- Análisis y reporte de tendencias adecuado de las preguntas de los clientes y su solución, de los tiempos de respuesta y la identificación de tendencias.

DS9 Administración de la configuración

Objetivo: dar cuenta de todos los componentes de TI, prevenir alteraciones no autorizadas, verificar la existencia física y proporcionar una base para el sano manejo de cambios.

Para ello se realizan controles que identifiquen y registren todos los activos de TI así como su localización física y un programa regular de verificación que confirme su existencia y toma en consideración:

- Registro de activos estableciendo procedimientos para asegurar que sean registrados únicamente elementos de configuración autorizados e identificables en el inventario, al momento de adquisición.
- Administración de cambios en la configuración asegurando que los registros de configuración reflejen el status real de todos los elementos de la configuración.
- Chequeo de software no autorizado revisando periódicamente las computadoras personales de la organización.
- Controles de almacenamiento de software definiendo un área de almacenamiento de archivos para todos los elementos de software válidos en las fases del ciclo de vida de desarrollo de sistemas.

DS10 Administración de problemas

Objetivo: Asegurar que los problemas e incidentes sean resueltos y que sus causas sean investigadas para prevenir que vuelvan a suceder.

Para ello se necesita un sistema de manejo de problemas que registre y dé seguimiento a todos los incidentes, además de un conjunto de procedimientos de escalamiento de problemas para resolver de la manera más eficiente los problemas identificados. Este sistema de administración de problemas deberá también realizar un seguimiento de las causas a partir de un incidente dado.

DS11 Administración de datos

Objetivo: Asegurar que los datos permanezcan completos, precisos y válidos durante su entrada, actualización, salida y almacenamiento.

Lo cual se logra a través de una combinación efectiva de controles generales y de aplicación sobre las operaciones de TI. Para tal fin, la gerencia deberá diseñar formatos de entrada de datos para los usuarios de manera que se minimicen los errores y las omisiones durante la creación de los datos.

Este proceso deberá controlar los documentos fuentes (de donde se extraen los datos), de manera que estén completos, sean precisos y se registren apropiadamente. Se deberán crear también procedimientos que validen los datos de entrada y corrijan o detecten los datos erróneos, como así también procedimientos de validación para transacciones erróneas, de manera que éstas no sean procesadas. Cabe destacar la importancia de crear procedimientos para el almacenamiento, respaldo y recuperación de datos, teniendo un registro físico (discos, discuetes, CD y cintas magnéticas) de todas las transacciones y datos manejados por la organización, albergados tanto dentro como fuera de la empresa.

La gerencia deberá asegurar también la integridad, autenticidad y confidencialidad de los datos almacenados, definiendo e implementando procedimientos para tal fin.

DS12 Administración de las instalaciones

Objetivo: proporcionar un ambiente físico conveniente que proteja al equipo y al personal de TI contra peligros naturales (fuego, polvo, calor excesivos) o fallas humanas lo cual se hace posible con la instalación de controles físicos y ambientales adecuados que sean revisados regularmente para su funcionamiento apropiado definiendo procedimientos que provean control de acceso del personal a las instalaciones y contemplen su seguridad física.

DS13 Administración de la operación

Objetivo: asegurar que las funciones importantes de soporte de TI estén siendo llevadas a cabo regularmente y de una manera ordenada.

Esto se logra a través de una calendarización de actividades de soporte que sea registrada y completada en cuanto al logro de todas las actividades. Para ello, la gerencia deberá establecer y documentar procedimientos para las operaciones de tecnología de información (incluyendo operaciones de red), los cuales deberán ser revisados periódicamente para garantizar su eficiencia y cumplimiento.

Dominio: monitoreo y evaluación (ME)

Todos los procesos de una organización necesitan ser evaluados regularmente a través del tiempo para verificar su calidad y suficiencia en cuanto a los requerimientos de control, integridad y confidencialidad. Este es, precisamente, el ámbito de este dominio.

Procesos:

M1 Monitoreo del proceso

Objetivo: Asegurar el logro de los objetivos establecidos para los procesos de TI. Lo cual se logra definiendo por parte de la gerencia reportes e indicadores de desempeño gerenciales y la implementación de sistemas de soporte así como la atención regular a los reportes emitidos.

Para ello la gerencia podrá definir indicadores claves de desempeño y/o factores críticos de éxito y compararlos con los niveles objetivos propuestos para evaluar el desempeño de los procesos de la organización. La gerencia deberá también medir el grado de satisfacción de los clientes con respecto a los servicios de información proporcionados para identificar deficiencias en los niveles de servicio y establecer objetivos de mejoramiento, confeccionando informes que indiquen el avance de la organización hacia los objetivos propuestos.

M2 Monitorear y evaluar el control interno

Objetivo: Asegurar el logro de los objetivos de control interno establecidos para los procesos de TI.

Para ello la gerencia es la encargada de monitorear la efectividad de los controles internos a través de actividades administrativas y de supervisión, comparaciones, reconciliaciones y otras acciones rutinarias, evaluar su efectividad y emitir reportes sobre ellos en forma regular. Estas actividades de monitoreo continuo por parte de la Gerencia deberán revisar la existencia de puntos vulnerables y problemas de seguridad.

M3 Garantizar el cumplimiento con requerimientos

Objetivo: Incrementar los niveles de confianza entre la organización, clientes y proveedores externos. Este proceso se lleva a cabo a intervalos regulares de tiempo.

Para ello la gerencia deberá obtener una certificación o acreditación independientes de seguridad y control interno antes de implementar nuevos servicios de tecnología de

información que resulten críticos, como así también para trabajar con nuevos proveedores de servicios de tecnología de información. Luego la gerencia deberá adoptar como trabajo rutinario tanto hacer evaluaciones periódicas sobre la efectividad de los servicios de tecnología de información y de los proveedores de estos servicios como así también asegurarse el cumplimiento de los compromisos contractuales de los servicios de tecnología de información y de los proveedores de estos servicios.

M4 Proporcionar gobierno de TI

Objetivo: Incrementar los niveles de confianza y beneficiarse de recomendaciones basadas en mejores prácticas de su implementación, lo que se logra con el uso de auditorias independientes desarrolladas a intervalos regulares de tiempo. Para ello la gerencia deberá establecer los estatutos para la función de auditoria, destacando en este documento la responsabilidad, autoridad y obligaciones de la auditoria. El auditor deberá ser independiente del auditado, esto significa que los auditores no deberán estar relacionados con la sección o departamento que esté siendo auditado y en lo posible deberá ser independiente de la propia empresa.

Esta auditoria deberá respetar la ética y los estándares profesionales, seleccionando para ello auditores que sean técnicamente competentes, es decir que cuenten con habilidades y conocimientos que aseguren tareas efectivas y eficientes de auditoria.

La función de auditoria deberá proporcionar un reporte que muestre los objetivos de la auditoria, período de cobertura, naturaleza y trabajo de auditoria realizado, así como también la organización, conclusión y recomendaciones relacionadas con el trabajo de auditoria llevado a cabo.

Los 34 procesos propuestos se concretan en 32 objetivos de control detallados anteriormente.

Un control se define como “las normas, estándares, procedimientos, usos y costumbres y las estructuras organizativas, diseñadas para proporcionar garantía razonable de que los objetivos empresariales se alcanzarán y que los eventos no deseados se preverán o se detectarán, y corregirán”.

Un objetivo de control se define como “la declaración del resultado deseado o propuesto que se ha de alcanzar mediante la aplicación de procedimientos de control en cualquier actividad de TI”.

En resumen, la estructura conceptual se puede enfocar desde tres puntos de vista:

- Los recursos de las TI.
- Los criterios empresariales que deben satisfacer la información.
- Los procesos de TI.

Para cada uno de estos 34 procesos, tiene un enlace a las metas de negocio y TI que soporta. Información de cómo se pueden medir las metas, también se proporcionan cuáles son sus actividades clave y entregables principales, y quién es el responsable de ellas.

Glosario

- **Amenaza** : Según [ISO/IEC 13335-1:2004], causa potencial de un incidente no deseado, el cual puede causar el daño a un sistema o la organización.
- **Análisis de riesgos** : Según [ISO/IEC Guía 73:2002], uso sistemático de la información para identificar fuentes y estimar el riesgo.
- **Análisis de riesgos cualitativo** : Análisis de riesgos en el que se usa una escala de puntuaciones para situar la gravedad del impacto.

- **Análisis de riesgos cuantitativo** : Análisis de riesgos en función de las pérdidas financieras que causaría el impacto.
- **Auditoría** : Proceso planificado y sistemático en el cual un auditor obtiene evidencias objetivas que le permitan emitir un juicio informado sobre el estado y efectividad del SGSI de una organización.
- **Auditor** : Persona encargada de verificar, de manera independiente, la calidad e integridad del trabajo que se ha realizado en un área particular.
- **Autenticación** : Proceso que tiene por objetivo asegurar la identificación de una persona o sistema.
- **Backup** : Acción de copiar archivos o datos de forma que estén disponibles en caso de que un fallo produzca la pérdida de los originales. Esta sencilla acción evita numerosos, y a veces irremediables problemas si se realiza de forma habitual y periódica.
- **Centro de cómputo** : Es un área de trabajo cuya función es la de concentrar, almacenar y procesar los datos y funciones operativas de una empresa de manera sistematizada.
- **Checklist** : Lista de apoyo para el auditor con los puntos a auditar, que ayuda a mantener claros los objetivos de la auditoría, sirve de evidencia del plan de auditoría, asegura su continuidad y profundidad y reduce los prejuicios del auditor y su carga de trabajo. Este tipo de listas también se pueden utilizar durante la implantación del SGSI para facilitar su desarrollo.
- **Cliente** : Cliente o 'programa cliente' es aquel programa que permite conectarse a un determinado sistema, servicio o red.
- **COBIT** : (Control Objectives for Information and related Technology) Objetivos de Control para la información y tecnología relacionadas. Publicados y mantenidos por ISACA. Su misión es investigar, desarrollar, publicar y promover un conjunto de objetivos de control de tecnología de información, aceptados para ser empleados por gerentes de empresas y auditores.
- **Control** : Las políticas, los procedimientos, las prácticas y las estructuras organizativas concebidas para mantener los riesgos de seguridad de la información por debajo del nivel de riesgo asumido.
- **Datos** : Término general para la información procesada por un ordenador.
- **Desastre** : Cualquier evento accidental, natural o malintencionado que interrumpe las operaciones o servicios habituales de una organización durante el tiempo suficiente como para verse la misma afectada de manera significativa.
- **Dominio** : Agrupación de objetivos de control de etapas lógicas en el ciclo de vida de inversión de TI.
- **Evaluación de riesgos** : Según [ISO/IEC Guía 73:2002]: proceso de comparar el riesgo estimado contra un criterio de riesgo dado con el objeto de determinar la importancia del riesgo.
- **Gestión de riesgos** : Proceso de identificación, control y minimización o eliminación, a un coste aceptable, de los riesgos que afecten a la información de la organización. Incluye la valoración de riesgos y el tratamiento de riesgos. Según [ISO/IEC Guía 73:2002]: actividades coordinadas para dirigir y controlar una organización con respecto al riesgo.
- **Hardware** : Conjunto de dispositivos de los que consiste un sistema. Comprende componentes tales como el teclado, el Mouse, las unidades de disco y el monitor.
- **Impacto** : El coste para la empresa de un incidente de la escala que sea, que puede o no ser medido en términos estrictamente financieros ej., pérdida de reputación, implicaciones legales, etc.
- **Información** : En sentido general, es todo lo que reduce la incertidumbre y sirve para realizar acciones y tomar decisiones.
- **Integridad** : Mantenimiento de la exactitud y completitud de la información y sus métodos de proceso. Según [ISO/IEC 13335-1:2004]: propiedad/característica de salvaguardar la exactitud y completitud de los activos.
- **Infraestructura** : La tecnología, los recursos humanos y las instalaciones que permiten el procesamiento de las aplicaciones.

- **Internet** : Interconexión de redes informáticas que permite a las computadoras conectadas comunicarse directamente.
- **Inventario de activos** : Lista de todos aquellos recursos (físicos, de información, software, documentos, servicios, personas, reputación de la organización, etc.) dentro del alcance del SGSI, que tengan valor para la organización y necesiten por tanto ser protegidos de potenciales riesgos.
- **ISACA** : (Information Systems Audit and Control Association) Asociación de Auditoría y Control de los Sistemas de Información. Publica COBIT y emite diversas acreditaciones en el ámbito de la seguridad de la información.
- **ISO** : (International Organization for Standardization) Organización Internacional para la Normalización. Organización de carácter voluntario fundada en 1946 que es responsable de la creación de estándares internacionales en muchas áreas, incluyendo la informática y las comunicaciones.
- **Mantenimiento Correctivo** : Medida de tipo reactivo orientada a eliminar la causa de una no-conformidad, con el fin de prevenir su repetición.
- **Mantenimiento Preventivo** : Medida de tipo pro-activo orientada a prevenir potenciales no-conformidades.
- **Norma** : Principio que se impone o se adopta para dirigir la conducta o la correcta realización de una acción o el correcto desarrollo de una actividad.
- **Objetivo** : Declaración del resultado o fin que se desea lograr mediante la implementación de procedimientos de control en una actividad de TI determinada.
- **Organización** : Conjunto de personas e instalaciones con una disposición de responsabilidades, autoridades y relaciones. Una organización puede ser pública o privada.
- **Políticas de seguridad** : Segundo [ISO/IEC 27002:2005]: intención y dirección general expresada formalmente por la Dirección.
- **Procedimiento** : Forma especificada para llevar a cabo una actividad o un proceso.
- **Proceso** : Por lo general, un conjunto de procedimientos influenciados por las políticas y estándares de la organización, que toman las entradas provenientes de un número de fuentes, incluyendo otros procesos, manipula las entradas, y genera salidas, incluyendo a otros procesos, para los clientes de los procesos. Los procesos tienen razones claras de negocio para existir, propietarios responsables, rol claro y responsabilidades alrededor de la ejecución del proceso, así como los medios para medir el desempeño.
- **Red** : Servicio de comunicación de datos entre ordenadores. Conocido también por su denominación inglesa: 'network'. Se dice que una red está débilmente conectada cuando la red no mantiene conexiones permanentes entre los ordenadores que la forman. Esta estructura es propia de redes no profesionales con el fin de abaratar su mantenimiento.
- **Riesgo** : Segundo [ISO Guía 73:2002]: combinación de la probabilidad de un evento y sus consecuencias.
- **Riesgo residual** : Segundo [ISO/IEC Guía 73:2002] El riesgo que permanece tras el tratamiento del riesgo.
- **Seguridad de la información** : Segundo [ISO/IEC 27002:2005]: Preservación de la confidencialidad, integridad y disponibilidad de la información; además, otras propiedades como autenticidad, responsabilidad, no repudio y fiabilidad pueden ser también consideradas.
- **Servidor** : Ordenador que ejecuta uno o más programas simultáneamente con el fin de distribuir información a los ordenadores que se conecten con él para dicho fin. Vocablo más conocido bajo su denominación inglesa 'server'.
- **Software** : Componentes inmateriales del ordenador: programas, SO, etc.
- **TI** : Tecnologías de Información.
- **Tratamiento de riesgos** : Segundo [ISO/IEC Guía 73:2002]: Proceso de selección e implementación de medidas para modificar el riesgo.
- **Usuario** : Una persona o una entidad externa o interna que recibe los servicios empresariales de TI.

- **Valoración de riesgos** : Según [ISO/IEC Guía 73:2002]: Proceso completo de análisis y evaluación de riesgos.
- **Vulnerabilidad** : Según [ISO/IEC 13335-1:2004]: debilidad de un activo o conjunto de activos que puede ser explotado por una amenaza.

Gestión de la atención a clientes y usuarios: centros de contacto, CRM. Arquitectura multicanal. Sistemas de respuesta de voz interactiva (IVR). Voice XML.

Gestión de la atención a clientes y usuarios: centros de contacto, CRM.

Introducción

La **customer relationship management** , más conocida por sus siglas **CRM** , puede tener varios significados:

- **Administración basada en la relación con los clientes** , un modelo de gestión de toda la organización, basada en la satisfacción del cliente (u orientación al mercado según otros autores). El concepto más cercano es *marketing relacional* (según se usa en España) y tiene mucha relación con otros conceptos como: *clienting*, *marketing 1×1*, *marketing directo de base de datos* , etc.
- **Software para la administración de la relación con los cliente** . Sistemas informáticos de apoyo a la gestión de las relaciones con los clientes, a la venta y al marketing, y que se integran en los llamados *Sistemas de Gestión Empresarial (SGE)* , y que incluyen *CRM*, *ERP*, *PLM*, *SCM*, y *SRM* . El software de CRM puede comprender varias funcionalidades para gestionar las ventas y los clientes de la empresa: automatización y promoción de ventas, tecnologías “data warehouse” (Almacén de datos) para agregar la información transaccional y proporcionar capa de reporting, dashboards e indicadores claves de negocio, funcionalidades para seguimiento de campañas de marketing y gestión de oportunidades de negocio, capacidades predictivas y de proyección de ventas.

Customer relationship management (CRM) es un enfoque para gestionar la interacción de una empresa con sus clientes actuales y potenciales. Utiliza el análisis de datos de la historia de los clientes con la empresa y para mejorar las relaciones comerciales con dichos clientes, centrándose específicamente en la retención de los mismos y, en última instancia, impulsando el crecimiento de las ventas.

Un aspecto importante del enfoque de CRM son los sistemas informáticos de CRM que recopilan datos de una variedad de canales de comunicación diferentes, incluidos el sitio web, el teléfono, el correo electrónico, el chat en vivo, los materiales de marketing y, más recientemente, las redes sociales de la compañía. A través del enfoque de CRM y los sistemas utilizados para facilitarlo, las empresas aprenden más sobre sus audiencias objetivo y cómo atender mejor sus necesidades. Sin embargo, la adopción del enfoque de CRM también puede ocasionalmente generar favoritismo entre una audiencia de consumidores, lo que resulta en insatisfacción entre los clientes y en derrotar el propósito de CRM.

Funciones

Cuando el software CRM está separado para gestionar el negocio, la gestión del ciclo de vida de las ventas y clientes es difícil o imposible. Y la gestión de ciclo de vida es muy importante ya que muchas empresas hoy en día interactúan con el cliente mucho tiempo después de que se realizó la venta, colaborando con ellos en la ingeniería bajo pedido (ETO), configurar a pedido (CTO) o procesos de gestión de servicios. Cada vez más el CRM debe ser extensible para apoyar a la planificación de recursos empresariales funcionalidades como la ingeniería, fabricación, compras, finanzas y gestión de servicios. Debido a que el CRM de empresa - o el CRM estratégico - es una parte integral del ERP, aporta información completa del cliente sobre el proyecto, las facturas, inventario, etc.

CRM como modelo de gestión

De acuerdo con Peppers y Rogers, "una empresa que se vuelve a sus clientes es una empresa que utiliza la información para obtener una ventaja competitiva y alcanzar el crecimiento y la rentabilidad. En su forma más generalizada, CRM puede ser considerado un conjunto de prácticas diseñadas, simplemente, para poner a una empresa en un contacto mucho más cercano con sus clientes. De este modo, aprender más acerca de cada uno, con el objetivo más amplio de que cada uno sea más valioso incrementando el valor de la empresa".

CRM social

CRM es una forma de pensar y actuar de una empresa hacia los clientes/consumidores. A partir de la formación de grandes corporaciones, el contacto 1 a 1 se va perdiendo y se despersonaliza cualquier transacción, dejando de lado la relación de los clientes con la marca.

El CRM, y especialmente el CRM Social nacen de la necesidad de recuperar los vínculos personales con los clientes, especialmente en la era de las *Redes Sociales*, en donde cada opinión se multiplica de forma viral y afecta significativamente la imagen de la marca. Por eso el Social CRM difiere del tradicional agregando la posibilidad de intercambio y conversación con los clientes.

Mediante la conexión constante y el registro de la información de la actividad, la empresa lleva un seguimiento de cada uno de sus contactos. Se les provee de información y soporte, se les avisa de nuevas activaciones y propuestas, y se les recompensa por producir contenido positivo. Esto conduce a una constante realimentación, pues los clientes tienen la posibilidad de opinar y compartir mediante redes sociales como *Facebook* y *Twitter*, que también permiten identificar prospectos y conocer sus gustos y preferencias. Así la producción de contenidos se vuelve cada vez más personalizada y relevante, profundizando la relación.

Un CRM abarca a los sistemas que mantienen datos específicos con el fin de mantener la relación de los clientes con la empresa en todo momento.

Módulo de ventas

Automatización de la parte o eslabón final: entre el cliente y el punto de venta. Un módulo de ventas es incluido en la mayoría de los CRM para poder tomar estas acciones dentro del almacén de datos. Por medio de esto se asigna oportunidades potenciales y tareas de manera automática según reglas predefinidas analizadas por medio de la información recaudada por los puntos de ventas automatizados.

Módulo de mercado

CRM que sea flexible, fácil de usar y que esté diseñada para la empresa. Transforma cada punto de contacto en una oportunidad de marketing y aprovecha el potencial oculto dentro de la base de datos de los clientes. Con las capacidades de marketing familiares y afines pueden comercializar productos de manera más eficaz, mejorar la productividad y obtener conocimientos accionables con los esfuerzos de marketing. Señala esfuerzos de marketing. Amplía la captura de pantalla. Usa consultas en idioma natural para segmentar de manera instantánea clientes o clientes potenciales. Crear listas altamente dirigidas y asociarlas con campañas y compañías. Configurar vistas personales o públicas para reutilización. Compartir fácilmente listas dirigidas con colegas y proveedores. Exportar listas en varios formatos para comunicaciones por correo electrónico masivo o correo directo. Planear actividades, tareas, presupuestos y detalles para cada actividad de marketing, y realizar su seguimiento. Coordinar de mejor manera las ventas al hacer un seguimiento de las oportunidades potenciales en un sistema centralizado. Asignar o clasificar oportunidades potenciales de manera automática según los flujos de trabajo predefinidos. El cliente o consumidor es el corazón de toda organización.

Características de programación

Los sistemas CRM tienen distintos módulos y categorías de programación, Plugins que funcionan de manera sincrónica realizando acciones durante la pre y post creación y actualización de registros y workflow que realiza tareas de manera asincrónica.

Software CRM

Se muestran algunos programas CRM:

- CiviCRM
- Cligraphcrm screenshots
- Hypos CRM
- IDempiere
- Microsoft Dynamics CRM
- Odoo
- OpenERP
- OpenZ
- Salesforce.com
- SAP CRM
- SugarCRM

Sistemas de respuesta de voz interactiva (IVR)

Introducción

La respuesta de voz interactiva o **IVR** (*Interactive Voice Response*) consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz y el reconocimiento de respuestas simples, como "sí", "no" u otras. Es un sistema automatizado de respuesta interactiva, orientado a entregar o capturar información a través del teléfono, permitiendo el acceso a servicios de información u otras operaciones. Los sistemas de IVR implementados en la red tienen capacidad para administrar grandes volúmenes de llamadas y también se usan para llamadas salientes, ya que estos sistemas son más inteligentes que muchos sistemas de marcación predictiva.

IVR se puede utilizar para compras con dispositivos móviles, pagos y servicios bancarios, pedidos minoristas, servicios públicos, información sobre viajes y condiciones

meteorológicas. Un concepto erróneo común hace referencia a un asistente automático como un sistema de IVR. Los términos son distintos y significan cosas diferentes para los profesionales tradicionales del ámbito de las telecomunicaciones: el propósito de un sistema de IVR es tomar la entrada, procesarla y devolver un resultado, mientras que la tarea de un asistente automático consiste en redirigir las llamadas. En ocasiones, también se utiliza el término **unidad de respuesta de voz** o VRU (*Voice Response Unit*).

Historia

A pesar del crecimiento de la tecnología IVR durante la década de 1970, la tecnología se consideraba compleja y costosa para la automatización de tareas en los centros de llamadas. Los primeros sistemas de respuesta de voz se basaban en la tecnología DSP y estaban limitados a vocabularios reducidos. A principios de 1980, Perception Technology de Leon Ferber se convirtió en el primer competidor del mercado principal, después de que la tecnología de discos duros (del acceso aleatorio de lectura/escritura a datos de voz digitalizados) alcanzara un nivel de precio rentable. En aquel momento, un sistema podía guardar palabras digitalizadas en un disco, reproducir el mensaje hablado apropiado y procesar la respuesta DTMF humana.

Se utilizan dos variantes principales del reconocimiento de voz en IVR: una basada en gramáticas predefinidas (utilizadas en diálogos "dirigidos") y otra basada en modelos de lenguaje preparados estadísticamente (utilizada en diálogos de "lenguaje natural"). Los diálogos dirigidos presentan a la persona que llama preguntas u opciones específicas. Los diálogos de lenguaje natural utilizan preguntas abiertas (por ejemplo, "¿En qué puedo ayudarlo?"), son más coloquiales y pueden interpretar respuestas de formato libre.

A menudo, un distribuidor automático de llamadas o ACD (Automatic Call Distributor) es el primer punto de contacto al llamar a muchas grandes empresas. Un ACD utiliza dispositivos de almacenamiento digital para reproducir saludos o anuncios, pero comúnmente redirige a la persona que llama sin solicitar ninguna entrada. Un sistema de IVR puede reproducir anuncios y solicitar una entrada a la persona que llama. Esta información se puede usar para obtener el perfil de la persona que llama y redirigir la llamada a un agente con un conjunto de aptitudes específico. (Un conjunto de aptitudes es una función que se aplica a un grupo de agentes de un centro de llamadas con una habilidad en concreto).

La respuesta de voz interactiva se puede utilizar para establecer la interfaz inicial de las operaciones de un centro de llamadas mediante la identificación de las necesidades de la persona que llama. Se puede obtener información de la persona que llama, como por ejemplo, un número de cuenta. Asimismo, se pueden brindar respuestas a preguntas simples, como saldos de cuentas o información previamente grabada, sin necesidad de que intervenga el operador. Con frecuencia, los números de cuenta obtenidos del sistema de IVR se comparan con los datos de identificación de la llamada por cuestiones de seguridad y, si la identificación de la llamada no coincide con el registro de la cuenta, se requieren respuestas de IVR adicionales.

Servicios

El IVR se implementa habitualmente en empresas o entidades que reciben gran cantidad de llamadas, a fin de reducir la necesidad de personal y los costes que el servicio ofrecido representan para dicha entidad. Entre otras, podemos mencionar a las bancas telefónicas.

Las empresas suelen usar la tecnología IVR para dirigir una llamada entrante hacia un departamento u otro, sin la necesidad de intervención humana, así reduciendo el tiempo de espera de sus clientes.

En los centros de atención telefónica al cliente, se utiliza la tecnología IVR para dirigir las llamadas hacia los agentes con mayor conocimiento de una materia específica, reduciendo así el tiempo de la llamada y evitando la necesidad de hacer transferencias entre agentes.

Se está implementando también en empresas de taxis, en las que la identificación del número que llama permite conocer dónde se encuentra el pasajero y generar el viaje rápidamente sin la intervención de un telefonista físico.

Puede combinarse con SMS para prestar cualquier clase de servicio: televotación, encuestas, sorteos, acceso a bases de datos, servicios informativos, etc.

Uso

Los sistemas de IVR se utilizan para atender gran cantidad de llamadas, reducir los costos y mejorar la experiencia del cliente. El uso de IVR y la automatización de voz permiten resolver las consultas de quienes llaman sin necesidad de colocar las llamadas en cola ni incurrir en el costo de un agente real. Si las personas que llaman no encuentran la información que necesitan o requieren asistencia adicional, las llamadas se suelen transferir a un agente. Esto da lugar a un sistema eficiente, que permite a los agentes tener más tiempo para abordar interacciones complejas. Cuando un sistema de IVR responde llamadas de varios números de teléfono, el uso del Servicio de identificación de número marcado (DNIS) garantiza la ejecución de la aplicación y el idioma correcto. Un único sistema de IVR grande puede atender llamadas para miles de aplicaciones, cada una con sus propios números de teléfono y guiones.

Sector bancario

Las instituciones bancarias dependen de los sistema de IVR para mantener las relaciones con los clientes y ampliar el horario comercial para ofrecer servicios las 24 horas de los 7 días de la semana. El servicio de Banca telefónica permite a los clientes consultar saldos e históricos de transacciones, así como realizar pagos y transferencias. A medida que han surgido los canales en línea, la satisfacción del cliente bancario ha disminuido.

Sector médico

Las empresas farmacéuticas y las organizaciones de investigación por contrato utilizan los sistemas de IVR para realizar ensayos clínicos y administrar el gran volumen de datos que se genera. La persona que llama responde a las preguntas en el idioma de su preferencia y las respuestas se ingresan en una base de datos y, posiblemente, se registran al mismo tiempo para confirmar la autenticidad. Entre las aplicaciones se incluyen la asignación aleatoria de pacientes y la gestión del suministro de medicamentos. También se emplean para registrar los diarios y cuestionarios del paciente.

Los sistemas de IVR permiten a quienes llaman obtener datos de manera relativamente anónima. En hospitales y clínicas, los sistemas de IVR se han utilizado para que quienes llaman puedan obtener acceso anónimo a los resultados de las pruebas. Este tipo de información podría ser fácilmente gestionada por una persona, pero se emplea el sistema de IVR para preservar la privacidad y evitar la posible incomodidad que puede suponer la información sensible o los resultados de las pruebas. Los usuarios reciben una clave de ingreso para poder acceder a sus resultados.

Encuestas

Algunas de las mayores plataformas de IVR instaladas se utilizan para la "televotación" en concursos televisivos, como Pop Idol y Gran Hermano, que pueden generar enormes picos de llamada. A menudo, el proveedor de red implementará un método de bloqueo de

destinos (call gapping) en la red telefónica pública conmutada (PSTN) para evitar la saturación de la red. Las organizaciones de sondeo también pueden usar sistemas de IVR para formular preguntas más sensibles cuando los investigadores tienen que un participante podría sentirse incómodo al dar sus respuestas a un interlocutor humano (por ejemplo, preguntas sobre consumo de drogas o comportamiento sexual). En algunos casos, se puede usar un sistema de IVR en la misma encuesta junto con un encuestador humano.

Tecnología involucrada

El IVR para brindar mejores servicios involucra otras tecnologías como, por ejemplo:

- **DTMF (Dual Tone Multi Frequency)** : Propia de la telefonía, es la tecnología de tonos utilizada para el marcado.
- **TTS (Text To Speech)** : Iniciada en la informática, le da capacidad de transformar texto a audio que escucha el operador.
- **ASR (Reconocimiento de Voz)** : Iniciada por la informática. Le da la capacidad de reconocer palabras del usuario y aceptarlas como órdenes.

Voice XML

Introducción

Muchos usuarios encuentran mucho más práctico los servicios automatizados por voz, y en la constante evolución de la web no ha podido faltar VoiceXML, que se ha convertido en un estandard W3C capaz de darnos la oportunidad de navegar interactuando con el ordenador utilizando exclusivamente nuestra voz, se deja de lado los periféricos como el ratón y el teclado para dar lugar al micrófono.

Como si se tratara de una conversación se establecen los roles de emisor y receptor, alternándose entre ordenador y usuario.

Historia

AT&T, IBM, Lucent, y Motorola creó el foro de VoiceXML en 1999, antes de septiembre de 1999 el foro lanzó VoiceXML 0.9 y en 2000 publicaron VoiceXML 1.0.

El W3C lo aceptó como "estándard" en marzo de 2004 en su versión 2.0, algo más tarde surgió la 2.1 añadiendo algunas pequeñas mejoras, las cuales se convirtieron en recomendación W3C en 2007.

Actualmente se está trabajando en VoiceXML 3.0, el cual utilizará un nuevo idioma descriptivo del statechart de XML llamado SCXML.

¿Qué es?

VoiceXML, es un lenguaje destinado al manejo y creación de aplicaciones de voz, que son empleadas para navegar, de forma auditiva en vez de utilizar la forma visual, más convencional y extendida hasta el momento.

VoiceXML es un lenguaje de etiquetas basado en XML que permite describir servicios de voz con independencia de la aplicación en la que corran. De esta manera no es necesario conocer detalles específicos de una plataforma para entender el funcionamiento del sistema de diálogo.

Los documentos que origina, son los llamados XML (eXtensible Markup Language), que admiten y poseen las características necesarias para dar lugar a la reproducción de sonidos digitales y sintetizados.

Possee un tipo de arquitectura no delimitada y de alto nivel de compatibilidades con respecto a las distintas salidas o recursos de la informática e internet.

El lenguaje VoiceXML describe la interacción hombre-máquina a partir de los siguientes elementos:

- Salida de texto-a-voz
- Salida de audio grabado
- Reconocimiento de entrada hablada
- Reconocimiento de tonos DTMF
- Grabación de entrada hablada
- Control de flujo de diálogo
- Funciones de telefonía (transferencia de llamada, desconexión, ect).

Componentes

Las aplicaciones de VoiceXML, contienen ciertos componentes, normalmente comunes entre ellos como:

El Servidor de aplicaciones que es el encargado al igual que cualquier función de un servidor, de proporcionar y almacenar datos de las aplicaciones e interfaces, para poder facilitarlas a otras externas.

Por otra parte, el Servidor de VoiceXML de Telefonía que es una plataforma que actúa como cliente frente al servidor de aplicaciones acabado de mencionar. Éste controla los diálogos producidos en VoiceXML, y los entiende para su control del habla y los diferentes recursos que posee (como ADR o TTS).

También posee una red de paquetes TCP/IP basada en la conexión del servidor de aplicaciones y el servidor de telefonía a través de protocolos HTTP.

Y a su vez, contiene una red telefónica comúnmente pública (PSTN), aunque no descarta la posibilidad de ser privada (PBX).

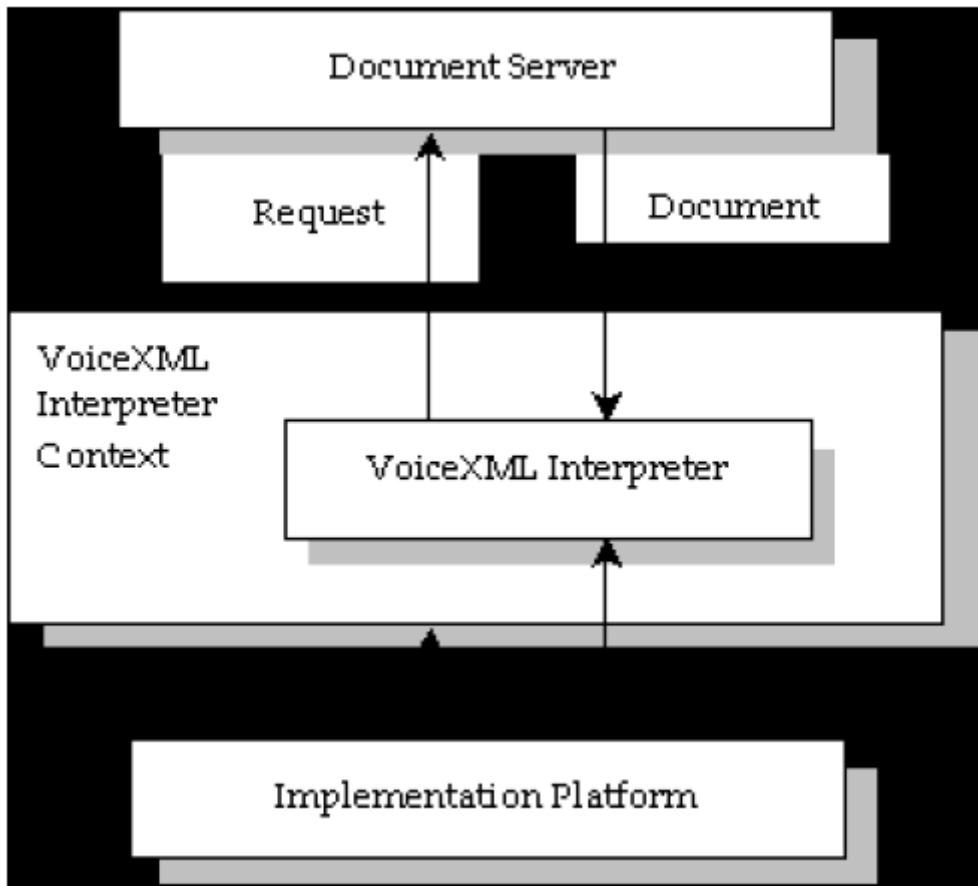
Funcionamiento

El usuario utiliza su voz para empezar a dar órdenes, de modo VoiceXML pone en marcha su ASR (un sistema encargado de reconocer la voz humana) transformando así la voz en una señal digital formada por 0's y 1's.

Una vez se procesa y si es necesario, la máquina puede contestar también mediante voz al usuario, poniendo en marcha el TTS y mediante éste se crean los documentos XML nombrados con anterioridad.

Para la creación de estos documentos, se utiliza ésta tecnología específica denominada TTS, que es referente a tecnologías de síntesis de voz.

Y la síntesis de voz consiste en la reproducción de manera no natural, es decir, artificial, del lenguaje natural y su origen proviene de las señales de voz que son generadas por el ordenador, que da lugar a un proceso inverso al ASR, es decir, transforma la señal digital que crea (respuesta) en voz entendible para el usuario.



Aplicaciones

VoiceXML está en expansión, y seguramente tenga cabida en multitud de entornos, actualmente es más usado en servicios telefónicos, un ejemplo claro lo encontramos cuando hacemos llamadas a nuestro operador telefónico, donde una voz nos va pidiendo datos para poder emparejarnos después con una persona real.

Otra aplicación importante es en los sistemas de información, incluso en el ámbito turístico, dando la opción de comunicarse con la máquina en múltiples idiomas.

Pero además de la comodidad que nos puede proporcionar una navegación mediante VoiceXML nos encontramos con una muy buena opción para dotar a cualquier página web de más usabilidad para gente con problemas de movilidad, incapaces de moverse con la soltura necesaria mediante los periféricos como el ratón y el teclado.

Ejemplo de sintaxis

Como ya sabemos, una de las primeras pruebas a la hora de empezar con un lenguaje es el famoso "Hola mundo".

Así quedaría en **VXML** (es la extensión de los ficheros VoiceXML):

```

<?xml version="1.0" encoding="iso-8859-1"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.w3.org/2001/vxml
                           http://www.w3.org/TR/voicexml20/vxml.xsd"
       version="2.0">
<property name="xml:lang" value="es"/>
<form id="saludo">
  <block>

```

```
<prompt>¡Hola mundo!</prompt>
<disconnect/>
</block>
</form>
```

Conclusiones

VoiceXML, utilizado de manera conjunta con otros estándares, proporciona una base sólida para la definición de sistemas de voz. Las constantes revisiones y ampliaciones del estándar aseguran su continuidad y progresiva incorporación en las herramientas para la construcción de aplicaciones de voz.

El aumento de los sistemas de telefonía a través de Internet y los progresos en los campos del reconocimiento de voz y las herramientas digitales para la lecturas así como la progresiva incorporación de los estándares propuestos por el W3C, señalan la gran importancia que sin duda VoiceXML y el resto de estándares de voz tendrán en un futuro muy próximo.

Seguridad física y lógica de un sistema de información. Herramientas en ciberseguridad. Gestión de incidentes. Informática forense.

Seguridad Lógica de un Sistema de Información

Introducción. Concepto de Seguridad Informática

Podemos entender como seguridad una característica de cualquier sistema (informático o no) que nos indica que ese sistema está libre de todo peligro, daño o riesgo, y que es, en cierta manera, infalible. Como esta característica, particularizando para el caso de SO o redes de ordenadores, es muy difícil de conseguir (según la mayoría de expertos, imposible), se suaviza la definición de seguridad y se pasa a hablar de fiabilidad (probabilidad de que un sistema se comporte tal y como se espera de él) más que de seguridad.

Se entiende que mantener un sistema seguro (o fiable) consiste básicamente en garantizar tres aspectos:

- **Confidencialidad o Privacidad** . Es la necesidad de que la información sólo sea conocida por personas autorizadas no convirtiendo esa información en disponible para otras entidades. En casos de falta de confidencialidad, la información puede provocar daños a sus dueños o volverse obsoleta.
- **Integridad** . Es la característica que hace que el contenido de la información permanezca inalterado a menos que sea modificado por personal autorizado, y esta modificación sea registrada para posteriores controles o auditorias.
- **Disponibilidad u Operatividad** . Es la capacidad de que la información esté siempre disponible para ser procesada por personal autorizado. Esto requiere que dicha información se mantenga correctamente almacenada con el hardware y el software funcionando perfectamente y que se respeten los formatos para su recuperación en forma satisfactoria.

Generalmente tienen que existir los tres aspectos descritos para que haya seguridad.

Los tres elementos principales a proteger en cualquier sistema informático son el software, el hardware y los datos. Habitualmente los datos son el principal elemento, ya

que es el más amenazado y el más difícil de recuperar. Así, por ejemplo en una máquina Unix tenemos un hardware ubicado en un lugar de acceso físico restringido, o al menos controlado, en caso de pérdida de una aplicación (o un programa de sistema, o el propio núcleo de Unix) este software se puede restaurar sin problemas desde su medio original (por ejemplo, el CD-ROM con el SO que se utilizó para su instalación). Sin embargo, en caso de pérdida de una BD o de un proyecto de un usuario, no tenemos un medio 'original' desde el que restaurar: hemos de pasar obligatoriamente por un sistema de copias de seguridad, y a menos que la política de copias sea muy estricta, es difícil devolver los datos al estado en que se encontraban antes de la pérdida.

Se deben conocer las amenazas a que los datos están sometidos y las vulnerabilidades de los sistemas en los que residen, así como los principales tipos de ataques para crear una buena política de seguridad que los proteja.

Seguridad Física y Seguridad Lógica

El estudio de la seguridad puede estudiarse dependiendo de las fuentes de las amenazas a los sistemas, lo que da lugar a hablar de seguridad física y seguridad lógica.

La seguridad física trata de la protección de los sistemas ante amenazas físicas. Consiste en la aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas, ante amenazas a los recursos e informaciones confidenciales. Desastres naturales, sabotajes internos o externos, etc, forman parte de este tipo de seguridad.

La seguridad lógica protege la información dentro de su propio medio mediante el uso de herramientas de seguridad. Se puede definir como conjunto de operaciones y técnicas orientadas a la protección de la información contra la destrucción, la modificación, la divulgación indebida o el retraso en su gestación.

El activo más importante de una organización es la información por lo que es necesario técnicas que vayan más allá de la seguridad física para proteger dicha información. Estas técnicas las brinda la seguridad lógica.

Seguridad Lógica

La Seguridad Lógica consiste en la "*aplicación de barreras y procedimientos que resguarden el acceso a los datos y sólo permitan acceder a ellos a las personas autorizadas para hacerlo*".

Los objetivos que se plantean serán:

1. Restringir el acceso a los programas y archivos.
2. Asegurar que los operadores puedan trabajar sin una supervisión minuciosa y no puedan modificar los programas ni los archivos que no les correspondan.
3. Asegurar que se estén utilizando los datos, archivos y programas correctos por el procedimiento correcto.
4. Que la información transmitida sea recibida por el destinatario al que ha sido enviada y no a otro.
5. Que la información recibida sea la misma que ha sido transmitida.
6. Que existan sistemas alternativos secundarios de transmisión entre diferentes puntos.
7. Que se disponga de pasos alternativos de emergencia para la transmisión de información

La seguridad lógica está estandarizada de acuerdo a unos niveles de seguridad. El estándar más utilizado internacionalmente es el TCSEC (Trusted Computer System Evaluation) Orange Book desarrollado en 1982 de acuerdo a las normas de seguridad de

ordenadores del Departamento de Defensa de los Estados Unidos. Los niveles describen diferentes tipos de seguridad del SO y se enumeran desde el mínimo grado de seguridad al máximo. Estos niveles han sido la base de desarrollo de estándares europeos (ITSEC/ITSEM, Information Technology Security Evaluation Criteria / Methodology) y luego internacionales (ISO/IEC). Cada nivel requiere todos los niveles definidos anteriormente: así el subnivel B2 abarca los subniveles B1, C2, C1 y D.

Nivel D

Reservado para sistemas que no cumplen con ninguna especificación de seguridad. Sin sistemas fiables, no hay protección para el hardware, el SO es inestable y no hay autenticación con respecto a usuarios y sus derechos de acceso a la información. Un SO en este nivel por ejemplo MS-DOS.

Nivel C1: Protección Discrecional

El acceso a distinta información se realiza mediante identificación de usuarios. Cada usuario maneja su información privada y distingue entre usuarios y el administrador del sistema, quien tiene control total de acceso.

Muchas de las tareas cotidianas de la administración sólo pueden ser realizadas por este “super usuario” quien tiene gran responsabilidad en la seguridad. Con la actual descentralización de los sistemas, no es raro que en una organización encontramos dos o tres personas cumpliendo este papel.

Los requerimientos mínimos que debe cumplir la clase C1 son:

- Acceso de control discrecional: distinción entre usuario y recursos. Se podrán definir grupos de usuarios y grupos de objetos sobre los cuales podrán actuar usuarios o grupos de ellos.
- Identificación y Autenticación: un usuario debe identificarse antes de comenzar a ejecutar acciones sobre el sistema. El dato de un usuario no podrá ser accedido por un usuario sin autorización o identificación.

Nivel C2: Protección de Acceso controlado

Cuenta con características adicionales al C1 que crean un ambiente de acceso controlado. Se debe llevar una auditoria de accesos e intentos fallidos de acceso a objetos. Tiene la capacidad de restringir aún más el que los usuarios ejecuten ciertos comandos o tengan accesos a ciertos archivos, permite o deniega datos a usuarios concretos en base no sólo a los permisos sino también a los niveles de autorización.

Requiere que se audite el sistema. Esta auditoria es utilizada para llevar registros de todas las acciones relacionadas con la seguridad, como las actividades efectuadas por el administrador y sus usuarios. La auditoria requiere de autenticación adicional para estar seguro de que la persona que ejecuta el comando es quien dice ser. Su mayor desventaja reside en los recursos adicionales requeridos.

Los usuarios de un sistema C2 tienen la autorización para realizar algunas tareas de administración del sistema sin necesidad de ser administradores. Permite llevar mejor cuenta de las tareas relacionadas con la administración del sistema, ya que es cada usuario quién ejecuta y no el administrador.

Nivel B1: Seguridad Etiquetada

A cada objeto del sistema (usuario, dato, etc) se le asigna una etiqueta con nivel de seguridad jerárquico (alto secreto, secreto, reservado, etc) y con unas categorías (contabilidad, nóminas, ventas, etc). Cada usuario que accede a un objeto debe poseer un permiso expreso para hacerlo y viceversa. Es decir que cada usuario tiene sus objetos

asociados. También se establecen controles para limitar la propagación de derecho de accesos a los distintos objetos.

Nivel B2: Protección Estructurada

Requiere que se etiquete cada objeto de nivel superior por ser padre de un objeto inferior. La protección estructurada es la primera que empieza a referirse al problema de un objeto a un nivel más elevado de seguridad y comunicación con otro objeto a un nivel inferior. Así un disco duro será etiquetado por almacenar archivos que son accedidos por distintos usuarios. El sistema es capaz de alertar a los usuarios si sus condiciones de accesibilidad y seguridad son modificadas; y el administrador es el encargado de fijar los canales de almacenamiento y ancho de banda a utilizar por los demás usuarios.

Nivel B3: Dominios de Seguridad

Refuerza a los dominios con las instalación de hardware: por ejemplo, el hardware de administración de memoria se usa para proteger el dominio de seguridad de acceso no autorizado a la modificación de objetos de diferentes dominios de seguridad. Existe un monitor de referencia que recibe las peticiones de acceso de cada usuario y las permite o las deniega según las políticas de acceso que se hayan definido.

Todas las estructuras de seguridad deben ser lo suficientemente pequeñas como para permitir análisis y comprobaciones ante posibles violaciones. Este nivel requiere que el terminal del usuario se conecte al sistema por medio de una conexión segura. Cada usuario tiene asignado los lugares y objetos a los que puede acceder.

Nivel A: Protección Verificada

Es el nivel más elevado, incluye un proceso de diseño, control y verificación, mediante métodos formales (matemáticos) para asegurar todos los procesos que realiza un usuario sobre el sistema.

Para llegar a este nivel todos los componentes de niveles inferiores deben incluirse. El diseño requiere ser verificado de forma matemática y se deben realizar análisis de canales encubiertos y de distribución fiable. El software y el hardware son protegidos para evitar infiltraciones ante traslados o movimientos de equipamiento.

Amenazas, Riesgos, Vulnerabilidades y Ataques

Amenaza

Bajo la etiqueta de ‘amenazas lógicas’ encontramos todo tipo de programas que de una forma u otra pueden dañar a nuestro sistema, creados de forma intencionada (software malicioso, también conocido como malware) o simplemente un error (bugs o agujeros). Esto es, una amenaza es la posibilidad de la ocurrencia de algún evento que afecte el buen funcionamiento de un sistema, es decir, cualquier elemento que comprometa el sistema.

Las amenazas pueden ser analizadas en tres momentos: antes del ataque, durante y después del mismo, por lo que son necesarios mecanismos que garanticen la seguridad para cada momento. Estos son:

- La prevención (antes): mecanismos que aumentan la seguridad (fiabilidad) de un sistema durante su funcionamiento normal. Por ejemplo, el cifrado de información.
- La detección (durante): mecanismos orientados a revelar violaciones a la seguridad. Generalmente son programas de auditoria.
- La recuperación (después): mecanismos que se aplican cuando la violación del sistema ya se ha detectado, para retornar éste a su funcionamiento normal. Por ejemplo, recuperación desde las copias de seguridad realizadas previamente.

La identificación de las amenazas requiere conocer los tipos de ataques, el tipo de acceso, método de trabajo y los objetivos del atacante.

Riesgo

Proximidad o posibilidad de daño sobre un bien.

Ya se trate de actos naturales, errores u omisiones humanas y actos intencionados, cada riesgo debería ser considerado de las siguientes maneras:

- Minimizando la posibilidad de que ocurra.
- Reduciendo al mínimo el perjuicio producido si no ha podido evitarse que ocurriera.
- Diseñando métodos para la más rápida recuperación de los daños experimentados.
- Corrigiendo las medidas de seguridad en función de la experiencia recogida.

Vulnerabilidad

Característica del sistema o del medio ambiente que facilita que la amenaza tenga lugar. Son las debilidades del sistema que pueden ser empleadas por la amenaza para comprometerlo.

Ataque

Evento, exitoso o no, que atenta contra el buen funcionamiento de un sistema, sea intencionado o accidental.

Las consecuencias de los ataques se podrían clasificar en:

- Corrupción de Datos: la información que no contenía defectos pasa a tenerlos.
- Denegación de Servicio: servicios que deberían estar disponibles no lo están.
- Filtrado: los datos llegan a destinos a los que no deberían llegar.

Los ataques de una forma general se clasifican en:

Ataques Pasivos

El atacante no altera la comunicación sino que únicamente la escucha o monitoriza para obtener información que está siendo transmitida. Sus objetivos son la interceptación de datos y el análisis de tráfico. Se suelen emplear para:

- Obtención del origen y destinatario de la comunicación, a través de la lectura de las cabeceras de los paquetes monitorizados.
- Control del volumen de tráfico intercambiando entre las entidades monitorizadas, obteniendo así información acerca de actividad o inactividad inusuales.
- Control de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los períodos de actividad.

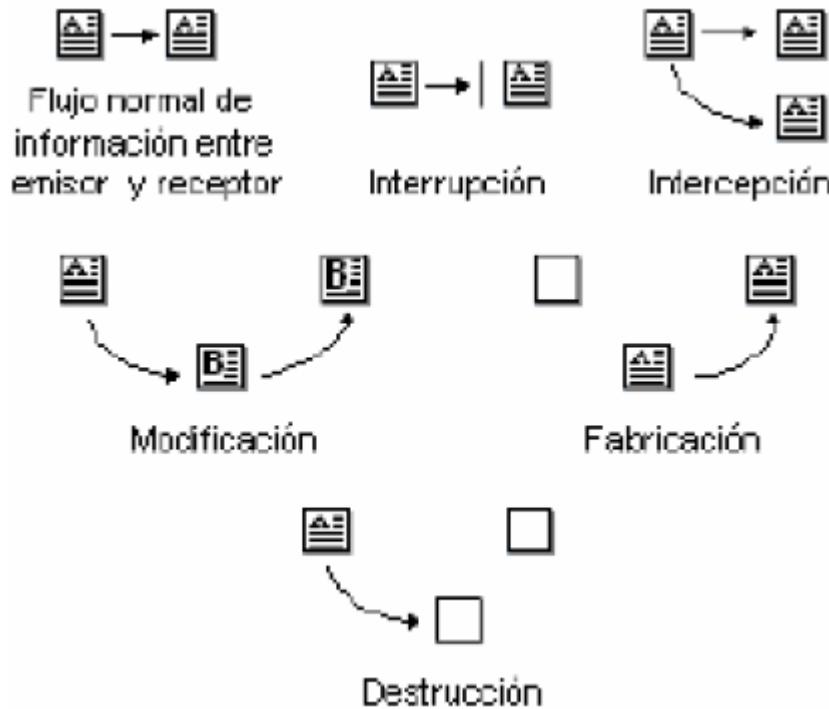
El cifrado de información, por ejemplo, puede evitar el éxito, si bien no el ataque.

Ataques Activos

Estos ataques implican algún tipo de modificación del flujo de datos transmitidos o la creación de un falso flujo de datos. Generalmente son realizados por hackers, piratas informáticos o intrusos remunerados y se los pueden subdividir en varias categorías:

- **Interrupción** : Un ataque se clasifica como interrupción si hace que un objeto del sistema se pierda, quede inutilizable o no disponible.

- **Interceptación** : Se tratará de una interceptación si un elemento no autorizado consigue un acceso a un determinado objeto del sistema.
- **Modificación** : Si además de conseguir el acceso consigue modificar el objeto.
- **Fabricación** : Se dice que un ataque es una fabricación si se trata de una modificación destinada a conseguir un objeto similar al atacado de forma que sea difícil distinguir entre el objeto original y el ‘fabricado’.
- **Destrucción** : Algunos autores consideran un caso especial de la modificación la destrucción, entendiéndola como una modificación que inutiliza al objeto afectado.



Tipos de Ataques

Ingeniería Social

Es la manipulación de las personas para convencerlas de que ejecuten acciones o actos que normalmente no realizan para que revele todo lo necesario para superar las barreras de seguridad. Esta técnica es una de las más usadas y efectivas a la hora de averiguar nombres de usuarios y passwords.

Para evitar situaciones de Ingeniería Social es conveniente tener en cuenta estas recomendaciones:

- Tener servicio técnico propio o de confianza.
- Instruir a los usuarios para que no respondan ninguna pregunta sobre cualquier característica del sistema y deriven la inquietud a los responsables que tenga la competencia para dar esa información.
- Asegurarse que las personas que llaman por teléfono son quien dicen ser.

Ingeniería Social Inversa

En este caso el intruso da a conocer de alguna manera que es capaz de brindar ayuda a los usuarios y estos llaman ante algún imprevisto. El intruso aprovechará la oportunidad para pedir información necesaria para solucionar el problema consiguiendo información útil.

Trashing

Generalmente un usuario anota su login y password en un papelito y luego, cuando lo recuerda, lo arroja a la basura. Este procedimiento por más inocente que parezca es el que

puede aprovechar un atacante para hacerse de una llave para entrar en el sistema. El Trashing puede ser físico (como el caso descrito) o lógico, como analizar buffers de impresora y memoria, bloques de discos, etc.

Ataques de Monitorización

Este tipo de ataque se realiza para observar a la víctima y su sistema, con el objetivo de obtener información y establecer sus vulnerabilidades y posibles formas de acceso futuro.

- **Shoulder Surfing** : Consiste en espiar físicamente a los usuarios para obtener su login y password correspondiente.
- **Decoy** : Son programas diseñados con la misma interfaz que el original. En ellos se imita la solicitud de login y el usuario desprevenido lo hace. Luego el programa guardará esa información y dejará paso a las actividades normales del sistema. La información recopilada será utilizada por el atacante para futuras “visitas”.

Una técnica semejante es aquella, que mediante un programa se guardan todas las teclas presionadas durante una sesión.

- **Scanning** : La idea es recorrer (escanear) tantos puertos de escucha como sea posible, y guardar información de aquellos que sean receptivos o de utilidad para cada necesidad en particular. Muchas utilidades de auditoria también se basan en este paradigma.

Hay varios tipos de scanning entre los que destaca **TCP Connect Scanning** que es la forma básica de escaneo de puertos TCP. Si el puerto está escuchando devolverá una respuesta de éxito; cualquier otro caso significará que el puerto no está abierto o que no se puede establecer conexión con él. Su ventaja es que es rápido y no necesita privilegios especiales. Su desventaja es que es fácilmente detectable por el administrador del sistema.

- **Eavesdropping-Packet Sniffing** : Es la interceptación del tráfico de red.

Esto se realiza con Packet Sniffers, son programas que controlan los paquetes que circulan por una red. Los sniffers pueden ser colocados tanto en estaciones de trabajo conectadas a la red como a un equipo Router o un Gateway de Internet y esto puede ser realizado por un usuario con legítimo acceso o por un intruso que ha ingresado por otras vías.

Cada máquina conectada a la red (mediante una placa con una dirección única) verifica la dirección destino de los paquetes TCP. Si estas direcciones son iguales asume que el paquete enviado es para ella, caso contrario libera el paquete para que otras placas lo analicen. Un sniffer consiste en colocar a la placa de red en un modo llamado promiscuo, el cual desactiva el filtro de verificación de direcciones y por tanto todos los paquetes enviados a la red llegan a esta placa. Actualmente existen sniffers para capturar cualquier tipo de información específica. Por ejemplo, passwords de un recurso compartido o de acceso a una cuenta, que generalmente viajan sin encriptar.

También son útiles para capturar números de tarjetas de crédito y direcciones de correo electrónico entrantes y salientes.

- **Snooping-Downloading** : Además de interceptar el tráfico de red, el atacante ingresa a los documentos, mensajes de correo electrónico y otra información guardada realizando una copia de sus documentos (downloading) a su propio ordenador para luego hacer un análisis exhaustivo de la misma.

Ataques de Autenticación

Este tipo de ataque tiene como objetivo engañar al sistema de la víctima para entrar en este. Generalmente se realiza tomando las sesiones ya establecidas por la víctima u obteniendo su nombre de usuario y password.

- **Spoofing-Looping** : Spoofing puede traducirse como “hacerse pasar por otro”. Una forma común de Spoofing es conseguir el nombre y password para una vez ingresado al sistema, tomar acciones en nombre de él.

El intruso usualmente utiliza un sistema para obtener información e ingresar en otro y luego utiliza este para entrar en otro y así sucesivamente. Este proceso llamado Looping, tiene la finalidad de ocultar la identificación y ubicación del atacante.

- **IP Spoofing** : El atacante genera paquetes de Internet con una dirección de red falsa en el origen, pero que es aceptada por el destinatario del paquete. Su utilización más común es enviar los paquetes con la dirección de un tercero de forma que la víctima “ve” un ataque proveniente de esa tercera red y no la dirección real del intruso.
- **DNS Spoofing** : Se manipulan paquetes UDP pudiéndose comprometer el servidor de nombres del Dominio (Domain Name Server DNS).
- **Web Spoofing** : El atacante crea un sitio web completo (falso) similar al que la víctima desea entrar. Los accesos a este sitio están dirigidos por el atacante permitiéndole controlar todas las acciones de la víctima, desde sus datos hasta las passwords, número de tarjetas de crédito, etc.

El atacante también es libre de modificar cualquier dato que se esté transmitiendo entre el servidor original y la víctima o viceversa.

- **IP Splicing-Hijacking** : Se produce cuando un atacante consigue interceptar una sesión ya establecida. El atacante espera a que la víctima se identifique ante el sistema y tras ello le suplanta.

Para entender el procedimiento supongamos la siguiente situación:

IP Cliente: IP 195.1.1.1
IP Servidor: IP 195.1.1.2
IP Atacante: IP 195.1.1.3

El cliente establece una conexión con su servidor enviando un paquete que contendrá la dirección origen, destino, número de secuencia y un número de autenticación utilizado por el servidor para reconocer el paquete siguiente en la secuencia. Supongamos que este paquete contiene:

IP Origen: 195.1.1.1 Puerto 1025
IP Destino: 195.1.1.2 Puerto 23
SEQ=3DF45ADA
ACK=F454FDF5
Datos: Solicitud

El servidor luego de recibir el primer paquete contesta al cliente con paquete Echo (recibido).

IP Origen: 195.1.1.2 Puerto 1025
IP Destino: 195.1.1.1 Puerto 23
SEQ=F454FDF5 (ACK enviado por el cliente)
ACK=3DF454E4
Datos: Recepción OK (Echo)

El cliente envía un paquete ACK al servidor, sin datos, en donde le comunica lo “perfecto de la comunicación.

IP Origen: 195.1.1.1 Puerto 1025
IP Destino: 195.1.1.2 Puerto 23
SEQ=3DF454E4 (ACK enviado por el servidor)
ACK=F454FDFF
Datos: Confirmación de Recepción (ACK)

El atacante que ha visto, mediante un Sniffer, los paquetes que circularon por la red calcula el número de secuencia siguiente: el actual + tamaño del campo de datos. Para calcular el tamaño de este campo:

1º paquete ACK Cliente=F454FDF5
2º paquete ACK Cliente=F454FDFF
Tamaño del campo de datos = F454FDFF - F454FD5 = 0A

Hecho esto el atacante envía un paquete con el siguiente aspecto:

IP Origen: 195.1.1.1 (IP del Cliente por el atacante)
IP Destino: 195.1.1.2 (IP del servidor)
SEQ=3DF454E4 (Último ACK enviado por el cliente)
ACK=F454FE09 (F454FDFF + 0A)

El servidor al recibir estos datos no detectará el cambio de origen ya que los campos que ha recibido como secuencia y ACK son los que esperaba recibir. El cliente, a su vez, quedará esperando datos como si su conexión estuviera establecida pero sin enviar datos y el atacante podrá seguir enviando datos mediante el procedimiento descrito.

- **Utilización de puertas traseras (Backdoors)** : Las puertas traseras son trozos de código en un programa que permiten saltarse los métodos usuales de autenticación para realizar ciertas tareas. Habitualmente son insertados por los programadores del sistema para agilizar la tarea de probar el código durante la fase de desarrollo. Esta situación se convierte en una falla de seguridad si se mantiene, involuntaria o intencionalmente, una vez terminado el producto ya que cualquiera que conozca el agujero o lo encuentre en su código podrá saltarse los mecanismos de control normales.
- **Utilización de Exploits** : Es frecuente entrar en un sistema explotando agujeros en los algoritmos de encriptación usados, en la administración de claves por parte de la empresa o encontrando un error en los programas utilizados.

Los programas para explotar estos “agujeros” reciben el nombre de Exploits y lo que hacen es aprovechar la debilidad, fallo o error hallado en el sistema (hardware o software) para entrar al mismo. Nuevos agujeros se publican cada día por lo que mantenerse informado de los mismos y de las herramientas para combatirlos es de vital importancia.

- **Obtención de Passwords** : Este método comprende la obtención por “Fuerza Bruta” de aquellas claves que permiten entrar en los sistemas, aplicaciones, cuentas, etc, atacados.

Muchas passwords de acceso son obtenidas fácilmente porque involucran el nombre u otro dato familiar del usuario, y además, nunca (o rara vez) se cambia. En este caso el ataque se simplifica e involucra algún tiempo de prueba y error. Otras veces se realizan ataques sistemáticos con la ayuda de programas especiales y diccionarios que prueban millones de posibles claves hasta encontrar la password correcta.

El programa encargado encripta cada una de ellas, mediante el algoritmo utilizado por el sistema atacado y compara la palabra encriptada contra el archivo de passwords del sistema atacado, previamente obtenido. Si coinciden se ha encontrado la clave de acceso.

Denegación de servicio (DoS - Denial of Service)

Los protocolos existentes actualmente fueron diseñados para ser empleados en una comunidad abierta y con una relación de confianza mutua. La realidad indica que es más fácil desorganizar el funcionamiento de un sistema que acceder al mismo; así los ataques de negación de servicio tienen como objetivo saturar los recursos de las víctimas de forma que se inhabiliten los servicios brindados por la misma.

Algunas razones por las que son útiles para un atacante son:

- Se ha instalado un troyano y se necesita que la víctima reinicie la máquina para que surta efecto.
- Se necesita cubrir inmediatamente sus acciones o un uso abusivo de CPU.
- El intruso cree que actúa bien al dejar fuera de servicio algún sitio web que le disgusta.
- El administrador de sistema quiere comprobar que sus instalaciones no son vulnerables a estos ataques.
- El administrador del sistema tiene un proceso que no puede “matar” en su servidor y debido a este no puede acceder al sistema. Para ello, lanza contra sí mismo un ataque DoS deteniendo los servicios.

Entre los distintos tipos de DoS tenemos:

- **Jamming o Flooding** : Este tipo de ataques desactivan o saturan los recursos del sistema. Por ejemplo, se puede consumir toda la memoria o espacio disponible en disco, así como enviar tanto tráfico a la red que nadie puede utilizarla.

El atacante satura el sistema con mensajes que requieren establecer conexión. Sin embargo, en vez de proveer la dirección IP del emisor, el mensaje contiene falsas direcciones IP usando “Spoofing” y “Looping”. El sistema responde al mensaje pero al no recibir respuesta acumula buffers con información de las conexiones abiertas, no dejando lugar a las conexiones legítimas.

- **Connection Flood** : La mayoría de las empresas que brindan servicios de Internet tienen un límite máximo en el número de conexiones simultáneas. Una vez que se alcanza este límite, no se admitirán conexiones nuevas. Así, por ejemplo, un servidor Web puede tener, capacidad para atender a mil usuarios simultáneos. Si un atacante establece mil conexiones y no realiza ninguna petición sobre ellas, monopolizará la capacidad del servidor. Las conexiones van caducando por inactividad poco a poco, pero el atacante sólo necesita intentar nuevas conexiones para mantener fuera de servicio al servidor.
- **Net Flood** : El atacante envía tantos paquetes de solicitud de conexión que las conexiones auténticas no pueden competir. En casos así el primer paso a realizar es ponerse en contacto con el Proveedor del servicio para que intente determinar la fuente del ataque, y como medida provisional, filtre el ataque en su extremo de la línea. El siguiente paso consiste en localizar las fuentes del ataque e informar a sus administradores, ya que seguramente se estarán usando sus recursos sin su conocimiento y consentimiento.
- **Land Attack** : Este ataque se basa en un error de la pila TCP/IP de las plataformas Windows. Consiste en mandar a algún puerto abierto de un servidor un paquete con la dirección y puerto origen igual que la dirección y puerto destino. El resultado es que después de cierta cantidad de mensajes enviados-recibidos la máquina termina colgándose.

- **OOB (Out Of Band), Supernuke o winnuke** : Es un ataque característico en los equipos Windows que hace que los equipos que escuchan por el puerto NETBios sobre TCP/UDP 137 a 139 queden fuera de servicio o disminuyan su rendimiento al enviarle paquetes UDP manipulados. Generalmente se envían fragmentos de paquetes OOB que la máquina víctima detecta como inválidos pasando a un estado inestable. OOB es el término normal pero realmente consiste en configurar el bit Urgente (URG) en los indicadores del encabezamiento TCP.
- **E-Mail Bombing - Spamming** : Consiste en enviar muchas veces un mensaje idéntico a una misma dirección saturando así el buzón de correo del destinatario.

El spamming, en cambio, se refiere a enviar un e-mail a miles de usuarios, hayan estos solicitados el mensaje o no. Es muy utilizado por las empresas para hacer publicidad de sus productos.

Ataques de Modificación-Daño

- **Tampering o Data Diddling** : Se refiere a la modificación desautorizada de los datos o el software instalado en el sistema víctima incluyendo borrado de archivos. Son particularmente serios cuando el que lo realiza ha obtenido derechos de administrador, con la capacidad de ejecutar cualquier comando. Ejemplos típicos: empleados bancarios (o externos) que crean falsas cuentas para derivar fondos de otras, estudiantes que modifican calificaciones de exámenes, et.

Múltiples sitios web han sido víctimas del cambio en sus páginas por imágenes o manifiestos. Otras veces se reemplazan versiones de software por otros con el mismo nombre pero que incorporan código malicioso (virus, troyanos, etc). La utilización de virus y troyanos está dentro de esta categoría y se le dedicará un apartado especial.

- **Borrado de Huellas** : El borrado de huellas es una de las tareas más importantes que debe realizar el intruso después de ingresar en un sistema, ya que si se detecta su ingreso, el administrador buscará como conseguir “tapar el hueco” de seguridad evitando ataques futuros o incluso rastrear al atacante.

Las huellas son todas las operaciones que realizó el intruso en el sistema y, por lo general, son almacenadas en Logs (archivos que guardan la información de lo que se realiza en el sistema) por el SO.

Los archivos de Logs son una de las principales herramientas con las que cuenta un administrador para conocer los detalles de las tareas realizadas en el sistema y para la detección de intrusos.

- **Ataque mediante ActiveX** : ActiveX es una de las tecnologías de Microsoft que permite reutilizar código, descargar código totalmente funcional de un sitio remoto, etc. ActiveX soluciona los problemas de seguridad mediante certificados y firmas digitales. Una Autoridad Certificadora (AC) expide un certificado que acompaña a los controles activos y a una firma digital del programador. Cuando un usuario descarga una página con un control, se le preguntará si confía en la AC que expedió el certificado y en el control ActiveX. Si el usuario acepta el control, éste puede pasar a ejecutarse sin ningún tipo de restricciones salvo las propias que tenga el usuario en el SO por lo que la seguridad del sistema se deja en manos del usuario, característica que es utilizada para realizar ataques.
- **Vulnerabilidad en los Navegadores** : Un fallo común ha sido el denominado “Buffer Overflow” que consisten en explotar una debilidad relacionada con los buffers que la aplicación usa para almacenar las entradas de usuario. Por ejemplo, cuando el usuario escribe una dirección en formato URL ésta se guarda en un buffer para luego procesarla. Si no se realizan las oportunas operaciones de comprobación, un usuario podría manipular estas direcciones.

Muchos sistemas están expuestos a “agujeros” de seguridad que son explotados para acceder a archivos, obtener privilegios o realizar sabotaje. Estas vulnerabilidades ocurren por varias razones y miles de “puertas invisibles” son descubiertas cada día en SO, aplicaciones de software, protocolos de red, exploradores de internet, correo electrónico y toda clase de servicios informáticos disponibles.

Los SO abiertos como Unix o Linux tienen agujeros más conocidos y controlados que aquellos que son cerrados, por ejemplo Windows. La importancia y ventaja del código abierto radica en miles de usuarios que analizan dicho código y buscan posibles errores y ayudan a obtener soluciones de forma inmediata.

Virus Informáticos

Un virus informático es un pequeño programa invisible para el usuario de funcionamiento específico y subrepticio cuyo código incluye información suficiente y necesaria para que utilizando los mecanismos de ejecución que le ofrecen otros programas puedan reproducirse formando réplicas de sí mismos susceptibles de mutar, resultando de dicho proceso la modificación, alteración y/o destrucción de los programas, información y/o hardware afectado. Un virus informático es un ataque de tipo “Tampering” que puede ser ingresado al sistema por un dispositivo externo o a través de la red (emails u otros protocolos) sin intervención directa del atacante.

Existen distintos tipos de virus, como aquellos que infectan archivos ejecutables, los sectores de arranque y la tabla de partición de los discos, etc. Los que causan mayores problemas suelen ser las macros y virus scripts que están ocultos en simples documentos, plantillas de cálculo, correo electrónico y aplicaciones que utiliza cualquier usuario de PC. La difusión se potencia con la posibilidad de su transmisión de un continente a otro a través de cualquier red o Internet y además son multiplataformas.

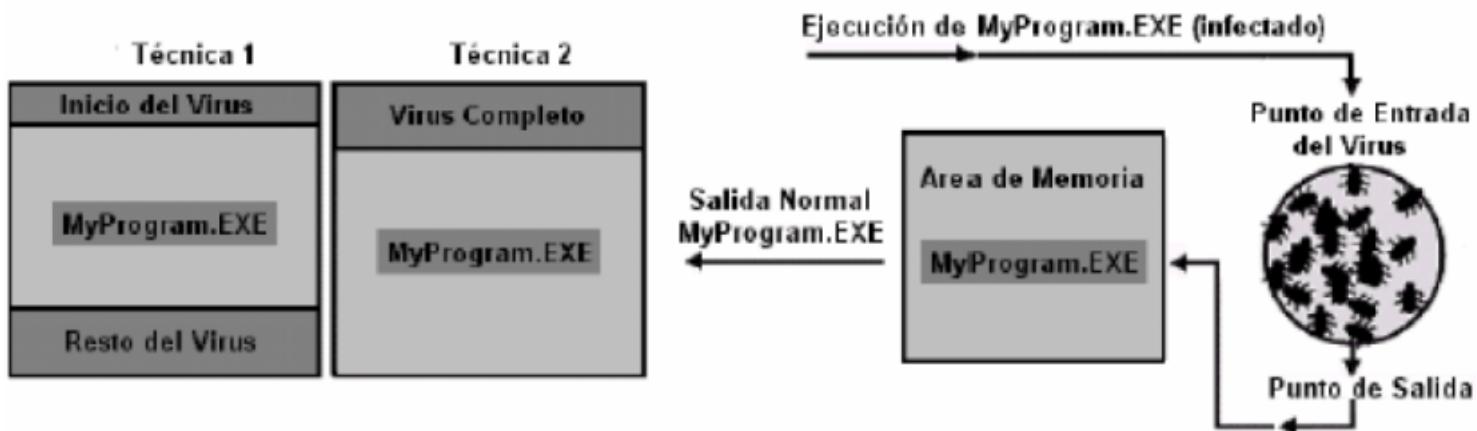
Las técnicas de propagación más comunes son:

- **Disquetes y otros medios:** A la posibilidad de que un disquete contenga un archivo infectado se une el peligro de que integre un virus de sector de arranque (Boot). En este segundo caso, y si el usuario lo deja en la disquetera infectará el ordenador al encenderlo ya que el sistema arrancará desde el disquete.
- **Correo electrónico:** el usuario no necesita hacer nada para recibir mensajes que en muchos casos ni siquiera ha solicitado y que pueden llegar de cualquier lugar del mundo. Los mensajes de correo electrónico pueden incluir archivos, documentos o cualquier objeto Active-X-Java infectado que al ejecutarse contagian la computadora del usuario. En las últimas generaciones de virus se envían emails sin mensajes pero con archivos adjuntos que al abrirlos proceden a su ejecución y posterior infección ya que se envían automáticamente a los contactos de la libreta de direcciones del sistema infectado.
- **IRC o chat:** Las aplicaciones de mensajería instantánea proporcionan un medio de comunicación anónimo, rápido, eficiente, cómodo y barato. Sin embargo, son peligrosas ya que los entornos chat ofrecen generalmente facilidades para transmisión de archivos con un gran riesgo en un entorno de red.
- **Páginas web y transferencias vía FTP:** los archivos que se descargan desde Internet pueden estar infectados y pueden provocar acciones dañinas en el sistema en el que se ejecutan.
- **Grupos de noticias:** Sus mensajes e información pueden estar infectados y por lo tanto contagiar al equipo del usuario que participe en ellos.

Los tipos de virus más habituales son:

- **Archivos Ejecutables (Virus ExeVir)** : El virus se adosa a un archivo ejecutable y desvía el flujo de ejecución a su código, para luego retornar al huésped y ejecutar acciones esperadas por el usuario. Al realizarse esta acción el usuario no se percata

de lo sucedido. Una vez que el virus es ejecutado se aloja en memoria y puede infectar otros archivos ejecutables que son abiertos en esa máquina.



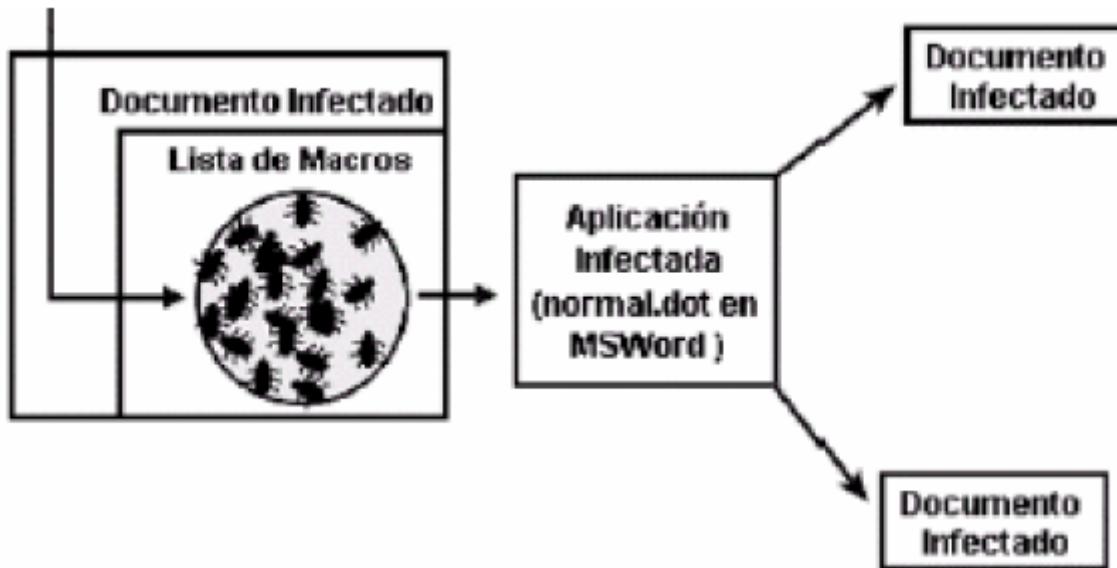
- **Virus en el Sector de Arranque (Virus Anterior a la Carga del SO)** : En los primeros 512 bytes de un disquete formateado están las rutinas necesarias para la carga y reconocimiento de dicho disquete. Entre ellas se encuentra la función invocada si no se encuentra el SO.

Se guarda la zona de arranque original en otro sector del disco. Luego el virus carga la antigua zona de arranque. Al arrancar el disquete ejecuta el virus (que obligatoriamente debe tener 512 bytes o menos) quedando residente en memoria, luego ejecuta la zona de arranque original, salvada anteriormente.



- **Virus Residente** : El objetivo de residir en memoria es controlar los accesos a disco realizados por el usuario y el SO. Cada vez que se produce un acceso, el virus verifica si el disco o archivo objeto al que se accede está infectado y si no lo está procede a almacenar su propio código en el mismo. Este código se almacenará en un archivo, tabla de partición o en el sector de arranque dependiendo del tipo de virus de que se trate.
- **Virus de Macros** : Estos virus infectan archivos de información generados por aplicaciones de oficina que cuentan con lenguajes de programación de macros. Su

funcionamiento consiste en que si una aplicación abre un archivo infectado, la aplicación (o parte de ella) se infecta y cada vez que se genera un nuevo archivo o se modifique uno existente contendrá el virus de macros.



- **Virus de Mail** : Su modo de actuar se basa en la confianza excesiva por parte del usuario, a este le lleva vía mail un mensaje con un archivo comprimido, el usuario lo descomprime y al terminar esta acción, el contenido del archivo se ejecuta y comienza el daño. Este tipo de virus tomó relevancia con la explosión masiva de Internet y virus tipo Melissa y I Love You. Generalmente estos virus se auto-envían a algunas de las direcciones de la libreta.
- **Hoax, los Virus Fantasmas** : No es un virus realmente. El auge del correo electrónico generó la posibilidad de transmitir mensajes de alerta de seguridad. Así comenzaron a circular mensajes de distinta índole de casos inexistentes. Los objetivos de estas alertas pueden causar alarma, pérdida de tiempo, robo de direcciones de correo y saturación de los servidores.
- **Gusanos** : Un gusano es un programa capaz de ejecutarse y propagarse por sí mismo a través de redes, en ocasiones portando virus o aprovechando errores de los sistemas a los que se conecta para dañarlos. Al ser difíciles de programar, su número no es muy elevado, pero el daño que pueden causar es muy grande.

Hemos de pensar que un gusano puede automatizar y ejecutar en unos segundos todos los pasos que seguiría un atacante humano para acceder a nuestro sistema, mientras que una persona, por muchos conocimientos y medios que posea, tardaría como mínimo horas en controlar nuestra red completa (un tiempo más que razonable para detectarlo), un gusano puede hacer eso mismo en pocos minutos: de ahí su enorme peligro y sus devastadores efectos.

- **Caballos de Troya** : Es un programa que aparentemente realiza una función útil a la vez que una operación que el usuario desconoce y que generalmente beneficia al autor del troyano o daña el sistema huésped.

Ejemplos conocidos son el Back Orifice y el Net Bus que son utilizados como poderosas armas para tomar el control del ordenador infectado. Estos programas pueden ser utilizados para la administración total del sistema atacado por parte de un tercero, con los mismos permisos y restricciones que el usuario de la misma.

Modelo de Virus Informático

Un virus está compuesto por su propio entorno dentro del que pueden distinguirse tres módulos principales:

- **Módulo de Reproducción** encargado de manejar las rutinas de parasitación de entidades ejecutables con el fin de que el virus pueda ejecutarse subrepticiamente.
- **Módulo de Ataque** que maneja las rutinas de daño adicional al virus y se activan cuando el sistema cumple cierta condición (por ejemplo una fecha).
- **Módulo de Defensa** con la misión de proteger al virus para evitar su detección o eliminación.

Medidas de Protección y Aseguramiento

Hoy es imposible hablar de un sistema cien por cien seguro porque el coste de la seguridad total es muy alto. Sin embargo, sí se pueden aplicar una serie de medidas de protección para controlar todo un conjunto de vulnerabilidades aunque no se logre la seguridad total. En este sentido las Políticas de Seguridad Informática (PSI) surgen como una herramienta de organización para concienciar a los miembros de una organización sobre la importancia y sensibilidad de la información y servicios críticos.

Entre las medidas de protección más comunes tenemos las que se exponen a continuación.

Controles de acceso

Estos controles pueden implementarse en el SO, sobre los sistemas de aplicación, en BD, en un paquete específico de seguridad o en cualquier otro sistema que se utilice.

Constituyen una importante ayuda para proteger al SO de la red, al sistema de aplicación y demás software de la utilización o modificación no autorizadas para mantener la integridad de la información y para resguardar la información confidencial del acceso no autorizado.

Asimismo es conveniente tener en cuenta otras consideraciones referidas a la seguridad lógica como por ejemplo las relacionadas al procedimiento que se lleva a cabo para determinar si corresponde un permiso de acceso (solicitado por un usuario) a un determinado recurso. Al respecto el NIST (National Institute for Standards and Technology) ha resumido los siguientes estándares de seguridad que se refieren a los requisitos mínimos de seguridad en cualquier sistema:

Identificación y Autenticación

Se denomina identificación el momento en que el usuario se da a conocer en el sistema mientras que autenticación es la verificación que realiza el sistema sobre esta identificación.

Existen cuatro tipos de técnicas que permiten realizar la autenticación de la identidad del usuario, que pueden llevarse a cabo individual o combinadamente:

1. Algo que solamente el individuo conoce: una clave secreta o password, clave criptológica o un número de identificación personal (PIN).
2. Algo que la persona posee: por ejemplo una tarjeta magnética.
3. Algo que el individuo es y que lo identifica únicamente: las huellas digitales o la voz.
4. Algo que el individuo es capaz de hacer: por ejemplo los patrones de escritura.

En los dos primeros casos es frecuente que las claves se olviden o que las tarjetas o dispositivos se pierdan mientras que por otro lado los controles de autenticación

biométricos serían más apropiados y fáciles de administrar resultado ser también los más costosos por lo dificultoso de su implementación eficiente.

Desde el punto de vista de la eficiencia es conveniente que los usuarios sean identificados y autenticados solamente una vez, pudiendo acceder a partir de allí a todas las aplicaciones y datos a los que su perfil les permita tanto en sistema locales como en sistemas a los que debe acceder en forma remota.

Esto se denomina “single log-in” o sincronización de passwords.

Una de las posibles técnicas para implementar esta única identificación de usuarios sería la utilización de un servidor de autenticaciones sobre el cual los usuarios se identifican y que se encarga luego de autenticar al usuario sobre los restantes equipos a los que éste pueda acceder. Este servidor de autenticaciones no debe ser necesariamente un equipo independiente y puede tener sus funciones distribuidas tanto geográfica como lógicamente, de acuerdo con los requerimientos de carga de tareas.

La seguridad informática se basa en gran medida en la efectiva administración de los permisos de acceso a los recursos informáticos, basados en la identificación, autenticación y autorización de accesos. Esta administración abarca:

- Proceso de solicitud, establecimiento, manejo y seguimiento y cierre de las cuentas de usuarios. Es preciso considerar que la solicitud de habilitación de un permiso de acceso para un usuario determinado, debe provenir de su superior y de acuerdo con sus requerimientos específicos de acceso debe crearse el perfil en el sistema de seguridad, en el SO o en la aplicación según corresponda.
- Además, la identificación de usuarios debe definirse según una norma homogénea para toda la organización.
- Revisiones periódicas sobre la administración de las cuentas y los permisos de acceso establecidos. Las mismas deben encararse desde el punto de vista del SO, y aplicación por aplicación, pudiendo ser llevadas a cabo por personal de auditoria o por la gerencia propietaria del sistema; siempre sobre la base de que cada usuario disponga del mínimo permiso que requiera de acuerdo con sus funciones.
- Las revisiones deben orientarse a verificar la adecuación de los permisos de acceso de cada individuo de acuerdo con sus necesidades operativas, la actividad de las cuentas de usuarios o la autorización de cada habilitación de acceso.
- Detección de actividades no autorizadas. Además de realizar auditorias o efectuar el seguimiento de los registros de transacciones (pistas) hay otras medidas que ayudan a detectar actividades no autorizadas. Algunas se basan en evitar la dependencia de personas determinadas, estableciendo la obligatoriedad de tomar vacaciones o efectuar rotaciones periódicas a las funciones asignadas a cada una de ellas.
- Nuevas consideraciones sobre cambios en la asignación de funciones del empleado. Para implantar la rotación de funciones o en caso de reasignar funciones por ausencias temporales de algunos empleados, es necesario considerar la importancia de mantener actualizados los permisos de acceso.
- Procedimientos en caso de desvinculaciones de personal con la organización, amistosas o no.

Para evitar estas situaciones es recomendable anular los permisos de acceso a las personas que se desvincularán de la organización lo antes posible. En caso de despido, el permiso de acceso debería anularse previamente a la notificación de la persona sobre la situación.

Roles

El acceso a la información también puede controlarse a través de la función o rol del usuario que requiere dicho acceso. Algunos ejemplos de roles son: programador, jefe de proyecto, gerente, administrador de sistema, etc. En este caso los derechos de acceso se pueden agrupar de acuerdo con el rol.

Limitaciones a los servicios

Estos controles se refieren a las restricciones que dependen de parámetros propios de la utilización de la aplicación o preestablecidos por el administrador del sistema. Un ejemplo podría ser que en la organización se disponga de licencias para la utilización simultánea de un determinado producto de software para cinco personas en donde exista un control a nivel de sistema que no permita la utilización del producto a un sexto usuario.

Modalidad de Acceso

Se refiere al modo de acceso que se permite al usuario sobre los recursos y a la información. Puede ser:

- Lectura: el usuario puede únicamente leer o visualizar la información pero no puede alterarla. La información puede ser copiada o impresa.
- Escritura: se permite agregar datos, modificar o borrar información.
- Ejecución: este acceso otorga al usuario el privilegio de ejecutar programas.
- Borrado: permite al usuario eliminar recursos del sistema.
- Todas las anteriores.

Ubicación y Horario

El acceso a determinados recursos del sistema puede estar basado en la ubicación física o lógica de los datos o personas. En cuanto a horarios, este tipo de controles permite limitar el acceso de los usuarios a determinadas horas del día o días de la semana. Así se mantiene un control más restringido de los usuarios y zonas de ingreso.

Control de Acceso Interno

Palabras claves (passwords)

Generalmente se utilizan para realizar la autenticación del usuario y sirven para proteger los datos y aplicaciones. Los controles independientes a través de la utilización de palabras claves resulta de muy bajo costo. Sin embargo, cuando el usuario se ve en la necesidad de utilizar varias palabras clave para acceder a diversos sistemas encuentra dificultoso recordarlas y probablemente las escriba o elija palabras fácilmente deducibles con lo que se ve disminuida de esta técnica.

Sincronización de passwords: consiste en permitir que un usuario acceda con la misma password a diferentes sistemas interrelacionados y su actualización automática en todos ellos en caso de ser modificada.

Caducidad y control: este mecanismo controla cuando pueden y deben cambiar sus passwords los usuarios. Se define el periodo mínimo que debe pasar para que los usuarios puedan cambiar sus passwords y un periodo máximo que puede transcurrir para que estas caduquen.

Encriptación

La información encriptada solamente puede ser desencriptada por quienes posean la clave apropiada.

Listas de control de accesos

Se refiere a un registro donde se encuentran los nombres de los usuarios que obtuvieron el permiso de acceso a un determinado recurso del sistema así como la modalidad de acceso permitido. Este tipo de listas varían considerablemente en su capacidad y flexibilidad.

Límites sobre la interfaz de usuario

Estos límites generalmente utilizados en conjunto con las listas de control de acceso restringen a los usuarios a realizar funciones específicas. Básicamente pueden ser de 3 tipos: menús, vistas sobre la BD y límites físicos sobre la interfaz de usuario. Por ejemplo, los cajeros automáticos donde el usuario sólo puede ejecutar ciertas funciones presionando teclas específicas.

Etiqueta de seguridad

Consiste en designaciones otorgadas a los recursos (por ejemplo, un archivo) que pueden utilizarse para varios propósitos como control de accesos, especificación de medidas de protección, etc. Estas etiquetas no son modificables.

Control de Acceso Externo

Dispositivos de control de puertos

Estos dispositivos autorizan el acceso a un puerto determinado y pueden estar físicamente separados o incluidos en otro dispositivo de comunicaciones, como por ejemplo un módem.

Firewalls o puertas de seguridad

Permiten bloquear o filtrar el acceso entre dos redes, usualmente una privada y otra externa (por ejemplo Internet). Los firewalls permiten que los usuarios internos se conecten a la red exterior al tiempo que previenen la intromisión de atacantes o virus a los sistemas de la organización.

Acceso de personal contratado o consultores

Debido a que este tipo de personal en general presta servicios temporales, debe ponerse especial consideración en la política y administración de sus perfiles de acceso.

Accesos públicos

Para los sistemas de información consultados por el público en general, o los utilizados para distribuir o recibir información computerizada (mediante, por ejemplo, la distribución y recepción de formularios en soporte magnético o la consulta y recepción de información a través del correo electrónico) deben tenerse en cuenta medidas especiales de seguridad, ya que se incrementa el riesgo y se dificulta su administración.

Debe considerarse para estos casos de sistemas públicos, que un ataque externo o interno puede acarrear un impacto negativo en la imagen de la organización.

Administración

Una vez establecidos los controles de acceso sobre los sistemas y la aplicación, es necesario realizar una eficiente administración de estas medidas de seguridad lógica, lo que involucra la implantación, seguimiento, pruebas y modificaciones sobre los accesos de los usuarios de los sistemas. La política de seguridad que se desarrolle respecto a la seguridad lógica debe guiar a las decisiones referidas a la determinación de los controles de accesos y especificando las consideraciones necesarias para el establecimiento de perfiles de usuarios.

La definición de los permisos de acceso requiere determinar cual será el nivel de seguridad necesario sobre los datos, por lo que es imprescindible clasificar la información, determinando el riesgo que produciría una eventual exposición de la misma a usuarios no autorizados.

Así los diversos niveles de la información requerirán diferentes medidas y niveles de seguridad.

Para empezar la implantación, es conveniente comenzar definiendo las medidas de seguridad sobre la información más sensible o las aplicaciones más críticas y avanzar de acuerdo a un orden de prioridad descendente, establecido alrededor de las aplicaciones. Una vez clasificados los datos, deberán establecerse las medidas de seguridad para cada uno de los niveles.

Un programa específico para la administración de los usuarios informáticos desarrollado sobre la base de las consideraciones expuestas, puede constituir un compromiso vacío, si no existe una conciencia de la seguridad en la organización por parte de todos los empleados. Esta conciencia de la seguridad puede alcanzarse mediante el ejemplo del personal directivo en el cumplimiento de las políticas y el establecimiento de compromisos firmados por el personal, donde se especifique la responsabilidad de cada uno.

Pero además de este compromiso debe existir una concienciación por parte de la administración hacia el personal en donde se remarque la importancia de la información y las consecuencias de su pérdida o apropiación de la misma por agentes extraños a la organización.

Administración del personal y usuarios

Lleva generalmente cuatro pasos:

- Definición de puestos: debe contemplarse la máxima separación de funciones posibles y el otorgamiento del mínimo permiso de acceso requerido por cada puesto para la ejecución de las tareas asignadas.
- Determinación de la sensibilidad del puesto: es necesario determinar si la función requiere permisos rigurosos que le permitan alterar procesos, perpetrar fraudes o visualizar información confidencial.
- Elección de la persona para cada puesto: requiere considerar los requerimientos de experiencia y conocimientos técnicos necesarios para cada puesto. Asimismo, para los puestos definidos como críticos puede requerirse una verificación de antecedentes personales.
- Formación inicial y continua del empleado: cuando la persona seleccionada entra en la organización además de sus responsabilidades individuales para la ejecución de las tareas que se asignen, debe comunicársele las políticas de organizaciones haciendo hincapié en la política de seguridad. El individuo debe conocer las disposiciones de la organización, su responsabilidad en cuanto a la seguridad informática y lo que se espera de él.

Defensa de los Ataques

La mayoría de los ataques mencionados se basan en fallo de diseño inherentes a Internet (o sus protocolos) y a los SO utilizados, por lo que no son “solucionables” en un plazo breve de tiempo.

La solución inmediata en cada caso es mantenerse informado sobre todos los tipos de ataques existentes y las actualizaciones que permanentemente lanzan las empresas de software, principalmente en SO.

Las siguientes medidas preventivas son:

1. Mantener las máquinas actualizadas y seguras físicamente.
2. Mantener personal especializado en cuestiones de seguridad.

3. Aunque una máquina no contenga información valiosa hay que tener en cuenta que puede resultar útil para un atacante, a la hora de ser empleada en una denegación de servicio (DoS) coordinado o para ocultar su verdadera dirección.
4. No permitir el tráfico “broadcast” desde fuera de nuestra red. De esta forma evitamos ser empleados como “multiplicadores”.
5. Auditorías de seguridad y sistemas de detección.
6. Mantenerse informado constantemente sobre cada una de las vulnerabilidades encontradas y parches instalados. Para esto es recomendable estar suscrito a listas que brinden este tipo de información.
7. La información continua del usuario.

Política de Seguridad Informática

Se define (RFC 1244) como “una declaración de intenciones de alto nivel que cubre la seguridad de los sistemas informáticos y que proporciona las bases para definir y delimitar responsabilidades para las diversas actuaciones técnicas y organizativas que se requerirán”.

La política se refleja en una serie de normas, reglamentos y protocolos a seguir donde se definen las medidas a tomar para proteger la seguridad del sistema. Deben tener las siguientes características:

- Cubrir todos los aspectos relacionados con la seguridad.
- Adecuarse a las necesidades y recursos.
- Definir estrategias y criterios generales y adoptar en distintas funciones y actividades las alternativas ante circunstancias que se puedan dar repetidas veces.

Evaluación de Riesgos

El análisis de riesgos supone, además de calcular la posibilidad de que ocurran cosas negativas, los siguientes puntos:

- Poder obtener una evaluación económica del impacto de estos sucesos.
- Tener en cuenta la probabilidad de que sucedan cada uno de los problemas posibles.
- Conocer qué se quiere proteger, dónde y cómo, asegurando que con los costes en que se incurre se obtengan beneficios efectivos. Para ello se deberá identificar los recursos (hardware, software, información, personal, accesorios, etc) con que se cuenta y las amenazas a las que se está expuesto.

Los riesgos se suelen clasificar por su nivel de importancia y por la severidad de su pérdida:

1. Estimación del riesgo de pérdida del recurso (R_i)
2. Estimación de la importancia del recurso (I_i)

Para la cuantificación del riesgo de perder un recurso, es posible asignar un valor numérico de 0 a 10 tanto a la importancia del recurso (10 es el recurso de mayor importancia) como al riesgo de perderlo (10 es el riesgo más alto). El riesgo del recurso será el producto de su importancia por el riesgo de perderlo:

$$WR_i = R_i * I_i$$

Luego, con la siguiente fórmula es posible calcular el riesgo general de los recursos de la red:

$$W_R = \frac{(WR_1 * I_1 + WR_2 * I_2 + \Lambda + WR_n * I_n)}{I_1 + I_2 + \Lambda + I_n}$$

Otros factores que debe considerar para el análisis de riesgo de un recurso de red son su disponibilidad, su integración y su carácter confidencial, los cuales se pueden incorporar a la fórmula para ser evaluados.

Identificación de una Amenaza

Una vez conocido los riesgos, los recursos que se deben proteger y como su daño o falta pueden influir en la organización es necesario identificar cada una de las amenazas y vulnerabilidades que pueden causar estas bajas en los recursos.

Se suele dividir las amenazas existentes según su ámbito de actuación:

- Desastre del entorno (Seguridad física)
- Amenazas del sistema (Seguridad lógica)
- Amenazas en la red (Comunicaciones)
- Amenazas de personas

Estrategia de Seguridad

Para establecer una estrategia adecuada es conveniente pensar una política de protección en los distintos niveles que esta debe abarcar: Física, Lógica, Humana y la interacción que existe entre estos factores.

En cada caso considerado, el plan de seguridad debe incluir tanto una estrategia Proactiva (proteger o proceder) o de previsión de ataques para minimizar los puntos vulnerables existentes en la directiva de seguridad y desarrollar planes de contingencias, como una estrategia Reactiva (perseguir y procesar) posterior al ataque que ayuda al personal de seguridad a evaluar el daño causado o a implantar un plan de contingencia adecuado.

Auditoria y Control

Se estudiarán aparte en el último punto de este tema, dada su gran importancia.

Plan de Contingencia

Los planes de contingencia se elaboran como respuesta a la acción de los diferentes riesgos y tienen los siguientes objetivos fundamentales:

- Minimizar las interrupciones en la operación normal.
- Limitar la extensión de las interrupciones y de los daños que originen
- Posibilitar una vuelta al servicio rápida y sencilla
- Ofrecer al personal unas normas de actuación frente a emergencias
- Dotar de medios alternativos de proceso en caso de catástrofe

Para garantizar la validez del Plan y que no quede obsoleto, debe ser revisado periódicamente. El Plan de Contingencia recoge los siguientes planes como respuesta a los problemas:

- Plan de Emergencia: normas de actuación durante o inmediatamente después de cada fallo o daño.

- Plan de Recuperación: normas para reiniciar todas las actividades del proceso en el Centro.
- Plan de Respaldo: especifica todos los elementos y procedimientos precisos para mantener la seguridad de la información, como configuración del equipo, comunicaciones, SO y opciones, etc.

Equipos de Respuesta a Incidencias

Es aconsejable formar un equipo de respuestas a incidencias implicado en los trabajos del profesional de seguridad que incluye:

- Desarrollo de instrucciones para controlar incidencias.
- Creación del sector o determinación del responsable.
- Identificación de las herramientas de software para responder a incidentes y eventos.
- Investigación y desarrollo de otras herramientas de seguridad informática.
- Realización de actividades formativas y de motivación.
- Realización de investigaciones acerca de virus.
- Ejecución de estudios relativos a ataques al sistema.

Estos trabajos proporcionarán los conocimientos que la organización puede utilizar y la información que hay que distribuir antes y durante los incidentes.

Una vez que el Administrador de seguridad y el equipo de respuestas a incidentes han analizado la incidencia, el Administrador debe delegar la responsabilidad del control de incidentes en el equipo de respuesta que será el responsable de responder a incidentes como virus, gusanos o cualquier otro código dañino, engaños y ataques de personal interno.

Copias de Respaldo (Backups)

El backup de archivos permite tener disponible e íntegra la información para cuando sucedan los accidentes. Sin un backup es imposible devolver la información al estado anterior al desastre. Es necesario realizar un análisis coste/beneficio para determinar qué información será almacenada, espacios de almacenamiento destinados a tal fin, la forma de realización, las estaciones de trabajo que cubrirá el backup, etc.

Para una correcta realización y seguridad de backups se deberán tener en cuenta estos puntos:

- Se debe contar con un procedimiento de respaldo de los SO y de la información de usuario para poder reinstalar fácilmente en caso de sufrir un accidente.
- Se debe determinar el medio y las herramientas correctas para realizar las copias, basándose en análisis de espacios, tiempos de lectura/escritura, tipos de backup a realizar, etc.
- El almacenamiento de los backups debe realizarse en locales diferentes de donde reside la información primaria. De este modo se evita la pérdida si el desastre alcanza a todo el edificio o local.
- Se debe verificar, periódicamente la integridad de los respaldos que se están almacenando.
- Se debe contar con un procedimiento para garantizar la integridad física de los respaldos en previsión de robos o destrucción.
- Se debe contar con una política para garantizar la privacidad de la información que se respalda en medios de almacenamiento secundarios.
- Se debe contar con un procedimiento para borrar físicamente la información de los medios de almacenamiento antes de desecharlos.

Programas Antivirus

Un antivirus es una gran BD con la huella digital de todos los virus conocidos para identificarlos y con las pautas que más contienen los virus. Los fabricantes de antivirus avanzan tecnológicamente en la misma medida que lo hacen los creadores de virus. Esto sirve para combatirlos aunque no para prevenir la creación e infección de otros nuevos.

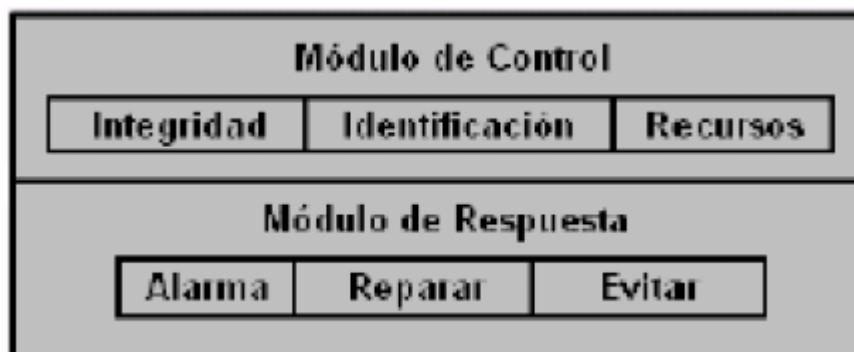
Las funciones presentes en un antivirus son:

- Detección: Se debe poder afirmar la presencia y/o ejecución de un virus en un ordenador. Adicionalmente puede brindar módulos de identificación, erradicación del virus o eliminación de la entidad infectada.
- Identificación de un virus: Existe diversas técnicas para realizar esta acción:
 - Scanning: Técnica que consiste en revisar el código de los archivos (fundamentalmente ejecutables y documentos) en busca de pequeñas porciones de código que puedan pertenecer a un virus almacenados en una BD del antivirus. Su principal ventaja reside en la rápida y exacta que resulta la identificación del virus. Sus desventajas con que brinda una solución a posteriori y es necesario que el virus alcance un grado de dispersión considerable para que llegue a mano de los investigadores y estos lo incorporen a su BD lo que no es una solución definitiva.
 - Heurística: Búsqueda de acciones potencialmente dañinas pertenecientes a un virus informático. No identifica con certeza al virus sino que rastrea rutinas de alteración de información y zonas generalmente no controlada por el usuario (Sector de arranque, FAT, etc). Su principal ventaja reside en que es capaz de detectar virus que no han sido agregados a las BD de los antivirus. Su desventaja radica en que puede sospechar de demasiadas cosas y el usuario debe ser medianamente capaz de identificar falsas alarmas.
- Comprobadores de Integridad. Controlan la actividad del PC señalando si algún proceso intenta modificar sectores críticos de la misma. Su ventaja residen en la prevención aunque a veces pueden ser vulnerados por los virus y ser desactivados por ellos, haciendo que el usuario se crea protegido, no siendo así.

Es importante diferenciar los términos “detectar”: determinación de la presencia de un virus e “identificar”: determinación de qué virus fue el detectado. Lo importante es la detección del virus y luego, si es posible su identificación y erradicación.

MODELO DE UN ANTIVIRUS

Un antivirus puede estar constituido por dos módulos principales y cada uno de ellos contiene otros módulos:



- **Módulo de Control**: Posee la técnica de verificación de Integridad que posibilita el registro de posibles cambios en las zonas y archivos considerados de riesgo.
- **Módulo de Respuesta**: La función de “alarma” se encuentra en todos los antivirus y consiste en detener la ejecución de todos los programas e informar al usuario de la

possible existencia de un virus. La mayoría ofrecen la posibilidad de su erradicación si la identificación ha sido positiva.

Herramientas de Seguridad

Cualquier herramienta de seguridad representa un arma de doble filo: de la misma forma que un administrador las utiliza para detectar y solucionar fallos en sus sistemas o en la red completa, un potencial intruso las puede utilizar para detectar esos mismos fallos y aprovecharlos para atacar los equipos. Herramientas como NESSUS, SAINT o SATAN pasan de ser útiles a ser peligrosas cuando las utilizan crackers que buscan información sobre las vulnerabilidades de un equipo o de una red completa.

La conveniencia de diseñar y distribuir libremente herramientas que puedan facilitar un ataque es un tema peliagudo; incluso expertos reconocidos como Alec Muffet (autor del adivinador de contraseñas Crack) han recibido enormes críticas por diseñar determinadas herramientas de seguridad para Unix. Tras numerosos debates sobre el tema, ha quedado claro que no se puede basar la seguridad de un sistema en el supuesto desconocimiento de sus problemas por parte de los atacantes: esta política, denominada "Security through obscurity", se ha demostrado inservible en múltiples ocasiones. Si los administradores no utilizan herramientas de seguridad que muestren las debilidades de los sistemas (para corregirlas), un atacante no va a dudar en utilizar tales herramientas (para explotar las debilidades encontradas).

Sistemas de Protección a la Integridad y Privacidad de la Información

Herramientas que utilizan criptografía para asegurar que la información sólo sea visible para quien tiene autorización. Su aplicación se realiza principalmente en las comunicaciones entre dos entidades. Dentro de este tipo de herramientas se pueden citar Pretty Good Privacy (PGP), Secure Sockets Layer (SSL) y los Certificados Digitales. También protocolos como Message Digest (MD5) o Secure Hash Algorithm (SHA) tratan de asegurar que no ha habido alteraciones indeseadas en la información que se intenta proteger.

Auditoria de Seguridad Lógica

La auditoria consiste en contar con los mecanismos para determinar qué sucede en el sistema, qué hace cada uno y cuando lo hace. Mediante una auditoria de seguridad informática, se pueden identificar los puntos fuertes y débiles de una organización con respecto al manejo de la seguridad de su información y se pueden definir claramente los pasos a seguir para lograr un perfeccionamiento de la misma.

Ventajas de una auditoria:

- Mejora sustancialmente la eficiencia en la seguridad de la información.
- Minimiza el riesgo de intrusión en sus sistemas, robos, uso indebido o alteración de información, abuso de privilegios o interrupción de los servicios ofrecidos.
- Elimina riesgos innecesarios.
- Posibilita la toma de decisiones sobre la seguridad de sus sistemas basándose en la información más completa.
- Posibilita la definición de responsabilidades bien diferenciadas.
- Brinda protección para toda la organización.

Premisas fundamentales:

- El nivel de seguridad satisfactorio "ahora", es mejor que un nivel de seguridad perfecto "a largo plazo".

- Es imprescindible conocer los propios puntos débiles y evitar riesgos imposibles de cuantificar.
- Realizar y exigir auditorias periódicas mejoran la salud de los sistemas y previenen ataques e incidentes.
- En una organización la seguridad es tan fuerte como el punto más débil de la misma, por lo tanto, interesa concentrarse (en un principio al menos) en estos últimos.
- Un 75% de las agresiones intencionadas son internas a las organizaciones.
- Lo más productivo es concentrarse en amenazas factibles y conocidas.

Las áreas principales a centrarse son las siguientes:

- **Política de seguridad** : Se necesita una política que refleje las expectativas de la organización en materia de seguridad a fin de suministrar administración con dirección y soporte. La política también se puede utilizar como base para el estudio y evaluación en curso.
- **Organización de la seguridad** : Sugiere diseñar una estructura de administración dentro de la organización que establezca la responsabilidad de los grupos en ciertas áreas de la seguridad y un proceso para el manejo de respuestas ante incidentes.
- **Control y clasificación de los recursos de información** : Necesita un inventario de los recursos de información de la organización y con base en éste conocimiento, debe asegurar que se brinde un nivel adecuado de protección.
- **Seguridad del personal** : Establece la necesidad de educar e informar a los empleados sobre lo que se espera de ellos en materia de seguridad y confidencialidad. También determina cómo incide el papel que desempeñan los empleados en materia de seguridad en el funcionamiento general de la compañía. Se debe implantar un plan para reportar los eventuales incidentes.
- **Seguridad física y ambiental** : Responde a la necesidad de proteger las áreas, el equipo y los controles generales.
- **Manejo de las comunicaciones y las operaciones** : Los objetivos de ésta sección son:
 - Asegurar el funcionamiento correcto y seguro de las instalaciones de proceso de datos.
 - Minimizar el riesgo de falla de los sistemas.
 - Proteger la integridad del software y la información.
 - Conservar la integridad y disponibilidad del procesamiento y la comunicación de la información.
 - Garantizar la protección de la información en las redes y de la infraestructura de soporte.
 - Evitar daños a los recursos de información e interrupciones en las actividades de la compañía.
 - Evitar la pérdida, modificación o uso indebido de la información que intercambian las organizaciones.
- **Control de acceso** : Establece la importancia de controlar el acceso a la red y los recursos de aplicación para proteger contra los abusos internos e intrusos externos.
- **Desarrollo y mantenimiento de los sistemas** : Recuerda que en toda labor de tecnología de la información, se debe implantar y mantener la seguridad mediante controles de seguridad en todas las etapas del proceso.
- **Manejo de la continuidad de la empresa** : Aconseja estar preparado para contrarrestar las interrupciones en las actividades de la empresa y para proteger los procesos importantes de la misma en caso de un fallo grave o un suceso fortuito.

Una auditoria de seguridad lógica debe centrarse en los siguientes aspectos:

- Contraseñas de acceso.
- Control de errores.
- Garantías de una transmisión para que sólo sea recibida por el destinatario. Para esto, regularmente se cambia la ruta de acceso de la información a la red.
- Registros de las actividades de los usuarios en la red.

- Encriptación de la información pertinente.
- Si se evita la importación y exportación de datos.
- Que el sistema pida el nombre de usuario y la contraseña para cada sesión:
 - Que en cada sesión de usuario, se revise que no accede a ningún sistema sin autorización y que si un usuario introduce incorrectamente su clave un número establecido de veces, su cuenta queda deshabilitada.
 - Si se obliga a los usuarios a cambiar su contraseña regularmente, y si las contraseñas son mostradas en pantalla cuando se introducen.
 - Si por cada usuario, se da información sobre su última conexión a fin de evitar suplantaciones.
- Si el software o hardware con acceso libre está inhabilitado.
- Generación de estadísticas de las tasas de errores y transmisión.
- Creación de protocolos con detección de errores.
- Mensajes lógicos de transmisión que han de llevar origen, fecha, hora y receptor.
- Software de comunicación, que ha de tener procedimientos correctivos y de control ante mensajes duplicados, fuera de orden, perdidos o retrasados.
- Datos importantes que sólo pueden ser impresos en una impresora especificada y ser vistos desde un terminal debidamente autorizado.
- Análisis del riesgo de aplicaciones en los procesos.
- Análisis de la conveniencia de cifrar los canales de transmisión entre diferentes organizaciones.
- Cifrados de los datos que viajan por internet.
- Si en la LAN hay equipos con módem entonces se debe revisar el control de seguridad asociado para impedir el acceso de equipos fuera de la red.
- Existencia de políticas que prohíben la instalación de programas o equipos personales en la red.
- Inhabilitación de los accesos a servidores remotos.
- Si la propia empresa genera ataques propios para probar la solidez de la red y encontrar posibles fallos en cada una de las siguientes facetas:
 - Servidores = Desde dentro del servidor y de la red interna.
 - Servidores web.
 - Intranet = Desde dentro.
 - Firewall = Desde dentro.
 - Accesos del exterior y/o Internet.

Software libre y software propietario. Características y tipos de licencias. La protección jurídica de los programas de ordenador. Tecnologías de protección de derechos digitales.

Software Libre versus Software Propietario

Definición de software libre

El software libre es aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto, debe venir acompañado del código fuente para hacer efectivas las libertades que lo caracterizan. Dentro de software libre hay, a su vez, matices que es necesario tener en cuenta. Por ejemplo, el software de dominio público significa que no está protegido por el copyright, por lo tanto, podrían generarse versiones no libres del mismo, en cambio el software libre protegido con copyleft impide a los redistribuidores incluir algún tipo de restricción a las libertades propias del software así concebido, es decir, garantiza que las

modificaciones seguirán siendo software libre. También es conveniente no confundir el software libre con el software gratuito, éste no cuesta nada, hecho que no lo convierte en software libre, porque no es una cuestión de precio sino de libertad. Para comprender este concepto, debemos pensar en la acepción de libre como en “libertad de expresión”. El software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar cambiar y mejorar el software. Y se refiere especialmente a cuatro clases de libertad para los usuarios de software:

- ◦ **Libertad 0** . La libertad para ejecutar el programa sea cual sea nuestro propósito.
- ◦ **Libertad 1** . La libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades -el acceso al código fuente es condición indispensable para esto-.
- ◦ **Libertad 2** . La libertad para redistribuir copias y ayudar así a tu vecino.
- **Libertad 3** . La libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad -el acceso al código fuente es condición indispensable para esto-.

Software libre es cualquier programa cuyos usuarios gocen de estas libertades. De modo que deberías ser libre de redistribuir copias con o sin modificaciones, de forma gratuita o cobrando por su distribución, a cualquiera y en cualquier lugar. Gozar de esta libertad significa, entre otras cosas, no tener que pedir permiso ni pagar para ello.

Asimismo, deberías ser libre para introducir modificaciones y utilizarlas de forma privada, ya sea en tu trabajo o en tu tiempo libre, sin siquiera tener que mencionar su existencia. Si se decidiera publicar estos cambios, no se debería estar obligado a notificárselo a ninguna persona ni de ninguna forma en particular.

La libertad para utilizar un programa significa que cualquier individuo u organización podrán ejecutarlo desde cualquier sistema informático, con cualquier fin y sin la obligación de comunicárselo subsiguientemente ni al desarrollador ni a ninguna entidad en concreto.

La libertad para redistribuir copias supone incluir las formas binarias o ejecutables del programa y el código fuente tanto de las versiones modificadas, como de las originales, ya que debemos tener la libertad para redistribuir tales formas si se encuentra el modo de hacerlo, pues las libertades para hacer cambios y para publicar las versiones mejoradas requieren de la accesibilidad de código fuente, por supuesto de manera libre, condición necesaria del software libre.

Cuando hablamos de software libre, debemos evitar utilizar expresiones como “regalar” o “gratis”, ya que se puede caer en el error de interpretarlo como una mera cuestión de precio y no de libertad.

Definición de software propietario

El software no libre también es llamado software propietario, software privativo, software privado o software con propietario. Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o que su código fuente no está disponible o el acceso a éste se encuentra restringido. En el software no libre una persona física o jurídica posee los derechos de autor sobre un software negando o no otorgando, al mismo tiempo, los derechos de usar el programa con cualquier propósito; de estudiar cómo funciona el programa y adaptarlo a las propias necesidades (donde el acceso al código fuente es una condición previa); de distribuir copias; o de mejorar el programa y hacer públicas las mejoras (para esto el acceso al código fuente es un requisito previo). De esta manera, un software sigue siendo no libre aún si el código fuente es hecho público, cuando se

mantiene la reserva de derechos sobre el uso, modificación o distribución. No existe consenso sobre el término a utilizar para referirse al opuesto del software libre. Entre los términos más utilizados e encuentran:

Software semilibre

Es aquel que mantiene las mismas características que el software libre para los usuarios individuales, entidades educativas o sin ánimo de lucro, sin embargo prohíbe esas libertades para su uso comercial o empresarial

Freeware

No tiene una definición clara y precisa, sin embargo suele usarse para clasificar al software que puede redistribuirse libremente pero no modificarse, entre otras cosas, porque no está disponible su código fuente. El freeware no es software libre.

Shareware

Es un software que permite su redistribución, sin embargo no viene acompañado de su código fuente y, por tanto, no puede ser modificado. Además, pasado un periodo de tiempo, normalmente es necesario pagar una licencia para continuar usándolo, luego tampoco es software libre.

Abandonware

El abandonware es “software cuyos derechos de autor ya no son definidos o que ya no está siendo vendido por la compañía que lo hizo”, y por eso, se dice que ha sido “abandonado”.

Warez

Si bien “ware” es un sufijo empleado en la jerga informática para formar términos que aluden a categorías de software, “warez” se refiere a una categoría de software distinta de las anteriores.

“Warez” es un término muy usado en las subculturas cracker para aludir a versiones crackeadas de software comercial, versiones en las cuales la protección de los derechos de autor ha sido quitada. Los hackers reconocen este término, pero no lo usan. Los warez son distribuciones de software sujetas a los derechos de autor, comercializadas en violación a la licencia de derechos de autor del software, en muchos casos la distribución de los warez es ilegal. La justificación que dan los crackers para el uso de los warez incluye la alegada posibilidad de la protección de los derechos de autor y la percibida injusticia de no compartir la información con aquellos que no podrían obtenerlo de otra manera que a través de la compra.

Libertad y costo

Es habitual que los usuarios confundan el software libre con el software gratuito. Es importante distinguir entre las libertades que nos proporciona un software y el coste del mismo. Un programa, por el simple hecho de ser gratuito, no es ni mucho menos libre. Por ejemplo, Internet Explorer de Microsoft es un programa gratuito pero no es libre, ya que no da a sus usuarios la posibilidad de estudiarlo (incluyendo el acceso a su código fuente), ni de mejorarlo, ni de hacer públicas estas mejoras con el código fuente correspondiente, de manera que todo el mundo se pueda beneficiar. Internet Explorer es un programa propietario -en cuanto a las libertades- y gratuito -en cuanto a su costo-. Existe una distinción fundamental entre los programas que garantizan los derechos de distribución y

modificación, el software libre, y los que no los garantizan que consideramos propietarios. Respecto al coste, cualquier software libre se puede vender, siempre y cuando se respeten las libertades originales que lo definen. Por ejemplo, la empresa frances Mandrake o la norteamericana Novell venden distribuciones de GNU/Linux, y se trata de software libre porque conserva las libertades que lo definen.

Open Source (código abierto)

Durante el año 1998 se lanzó la Open Software Initiative y propusieron el uso de término open source (código abierto) en contraposición al término free software (software libre) como término más atractivo al entorno empresarial. El término free software en el mundo anglófono creaba una situación incómoda debido a la doble acepción que en inglés tiene el término free (que puede significar gratuito o libre).

Se creó una lista de condiciones que debe cumplir un programa para poder ser considerado Open Source. Estas condiciones son muy similares y, de hecho están basadas, en las directrices de software libre de Debian. Éstas condiciones también son aplicables a cualquier programa que sea software libre y pueden ayudarnos a matizar sus implicaciones:

1. 1. Libre distribución. No se puede impedir la venta o distribución del programa o parte de él. Así mismo, tampoco se puede exigir el pago de un canon o tasa a cambio de su distribución por parte de terceros.
2. Código fuente. El programa debe incluir su código fuente y no se puede restringir su redistribución.
3. Trabajos derivados. No debe impedirse realizar modificaciones o trabajos derivados del programa y debe permitirse que éstos sean distribuidos bajo los mismos términos del software original.
4. Integridad del código de fuente original. Puede exigirse que una versión modificada del programa tenga un nombre y número de versión diferente que el programa original para poder proteger al autor original de la responsabilidad de estas versiones.
5. No discriminación contra personas o grupos. Las condiciones de uso del programa no pueden discriminar contra una persona o un grupo de personas.
6. No discriminación contra usos. No se puede negar a ninguna persona hacer uso del programa para ningún fin como, por ejemplo, comercial o militar.
7. Distribución de la licencia. Los derechos del programa deben aplicarse a todos quienes se redistribuyen el programa sin ninguna condición adicional.
8. La licencia no debe ser específica de un producto. Los derechos garantizados al usuario del programa no deben depender de que el programa forme parte de una distribución o paquete particular de software.
9. La licencia no debe restringir otro software. La licencia no debe poner restricciones en otros programas que se distribuyen junto con el software licenciado.
10. La licencia debe ser tecnológicamente neutra. No puede existir ninguna disposición de la licencia que obligue al uso de una tecnología concreta.

Ventajas y desventajas del software libre y del software propietario

Software libre

Ventajas:

- Bajo costo de adquisición y libre uso.
- Innovación tecnológica.
- Requisitos de hardware menores y durabilidad de las soluciones.
- Trabajo de forma cooperativa.
- Independencia del proveedor.

- Adaptación del software.
- Lenguas minoritarias, traducción, uso e impulso de difusión.

Desventajas:

- La curva de aprendizaje es mayor.
- El software libre no tiene garantía proveniente del autor.
- Los contratos de software propietario no se hacen responsables por daños económicos, y de otros tipos por el uso de sus programas.
- Se necesita dedicar recursos a la reparación de errores.
- No existen compañías únicas que respalden toda la tecnología.
- La mayoría de la configuración de hardware no es intuitiva.
- Únicamente los proyectos importantes y de trayectoria tienen buen soporte, tanto de los desarrolladores como de los usuarios.
- El usuario debe tener nociones de programación.
- La diversidad de distribuciones, métodos de empaquetamiento, licencias de uso, herramientas con un mismo fin, etc, pueden crear confusión.

Software Propietario

Ventajas:

- Control de calidad.
- Recursos de investigación.
- Personal altamente capacitado.
- Uso común por los usuarios.
- Software para aplicaciones muy específicas.
- Difusión de publicaciones acerca del uso y aplicación del software.
- Curva de aprendizaje menor.
- Soporte de las herramientas por diversas compañías.

Desventajas:

- Cursos de aprendizaje costosos.
- Secreto del código fuente.
- Soporte técnico ineficiente.
- Costosa la adaptación de un módulo del software a necesidades particulares.
- Derecho exclusivo de innovación.
- Ilegalidad de copias sin licencia.
- Imposibilidad de compartir.
- Quedar sin soporte técnico.
- Descontinuación de una línea de software.
- Dependencia de proveedores.
- Costo elevado de licencias.
- Prácticas monopolísticas.

Copyright y copyleft

Copyright

El símbolo del copyright “ © ”, es usado para indicar que una obra está sujeta al derecho de autor. El derecho de autor es un conjunto de normas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores (los derechos de autor), por lo solo hecho de la creación de una obra literaria, artística o científica, tanto publicada o que todavía no se haya publicado. Una obra pasa al dominio público cuando los derechos patrimoniales han expirado. Esto sucede habitualmente trascurrido un plazo desde la muerte del autor (post mortem auctoris).

Derecho de autor y Copyright

El derecho de autor y copyright constituyen dos concepciones sobre la propiedad literaria y artística. El primero proviene de la familia del derecho continental, particularmente del derecho francés, mientras que el segundo proviene del derecho anglosajón (common law). El derecho de autor se basa en la idea de un derecho personal del autor, fundado en una forma de identidad entre el autor y su creación. El derecho moral está constituido como emanación de la persona del autor; reconoce que la obra es expresión de la persona del autor y así se le protege. La protección del copyright se limita estrictamente a la obra, sin considerar atributos morales del autor en relación con su obra, excepto la paternidad; no lo considera como un autor propiamente tal, pero tiene derechos que determinan las modalidades de utilización de una obra.

Campo de aplicación

La protección del derecho de autor abarca únicamente la expresión de un contenido, pero no las ideas. Para su nacimiento no necesita de ninguna formalidad, es decir, no requiere de la inscripción en un registro o el depósito de copias, los derechos de autor nacen con la creación de la obra. Son objeto de protección las obras originales, del campo literario, artístico y científico, cualquiera que sea su forma de expresión, soporte o medio. Entre otras:

- Libros, folletos y otros escritos
- Obras dramáticas o dramático-musicales
- Obras coreográficas y las pantomimas
- Composiciones musicales con o sin letra
- Obras musicales y otras grabaciones sonoras
- Obras cinematográficas y otras obras audiovisuales
- Obras de dibujo, pintura, arquitectura, escultura, grabado, litografía
- Obras fotográficas
- Ilustraciones, mapas, planos, croquis y obras plásticas relativos a la geografía, a lo topográfico, a la arquitectura o a las ciencias
- Programas informáticos

Los derechos de autor

Generalmente le da al dueño del derecho de autor el derecho exclusivo para hacer y para autorizar a otros a utilizar su obra.

La protección del derecho de autor existe desde que la obra es creada de una forma fijada. El derecho de autor sobre una obra creada se convierte inmediatamente en propiedad del autor que creó dicha obra. Sólo el autor o aquellos cuyos derechos derivan del autor pueden reclamar propiedad. Los autores de una obra colectiva son co-dueños del derecho de autor de dicha obra a menos que haya un acuerdo que indique lo contrario. El derecho de autor de cada contribución individual de una publicación periódica o en serie, o cualquier otra obra colectiva, existen a parte del derecho de autor de una obra colectiva en su totalidad y están conferidos inicialmente al autor de cada contribución.

Copyleft

El símbolo del copyleft es “(ɔ)”, es utilizado como la contrapartida del símbolo del copyright, sin embargo no posee reconocimiento legal.

El término copyleft describe un grupo de licencias que se aplican a una diversidad de trabajos tales como el software. Una licencia copyleft se basa en las normas sobre el derecho de autor, las cuales son vistas por los defensores del copyleft como una manera de restringir el derecho de hacer y redistribuir copias de un trabajo determinado, para garantizar que cada persona que recibe una copia o una versión derivada de un trabajo,

pueda a su vez usar, modificar y redistribuir tanto el propio trabajo como las versiones derivadas del mismo. Así, y en un entorno no legal, el copyleft puede considerarse como opuesto al copyright.

Métodos de aplicar copyleft

La práctica habitual para conseguir este objetivo de explotación sin trabas, copia y distribución de una creación o de un trabajo y sus derivados es la de distribuirlo junto con una licencia. Dicha licencia debería estipular que cada propietario de una copia del trabajo pudiera:

1. Usarla sin ninguna limitación
2. (Re)distribuir cuantas copias desee
3. Modificarla de la manera que crea conveniente

Estas tres libertades, sin embargo, no son suficientes aún para asegurar que un trabajo derivado de una creación sea distribuido bajo las mismas condiciones no restrictivas: con este fin, la licencia debe asegurar que el propietario del trabajo derivado lo distribuirá bajo el mismo tipo de licencia. Otras condiciones de licencia adicionales que podrían evitar posibles impedimentos al uso sin trabas, distribución y modificación del trabajo incluirían:

- Asegurar que las condiciones de la licencia copyleft no pueden ser revocadas
- Asegurar que el trabajo y sus derivados son siempre puestos a disposición de manera que se facilite su modificación, para el software, esta facilidad suele asociarse a la disponibilidad del código fuente, donde incluso la compilación de dicho código debería permitirse sin ninguna clase de impedimento.
- Idear un sistema más o menos obligatorio para documentar adecuadamente la creación y sus modificaciones, por medio de manuales de usuario, descripciones, etc.

Tipos de Licencias de Software

Definiciones

- **Licencia** : contrato entre el desarrollador de un software sometido a propiedad intelectual y a derechos de autor y el usuario, en el cual se definen con precisión los derechos y deberes de ambas partes. Es el desarrollador, o aquél a quien éste haya cedido los derechos de explotación, quien elige la licencia según la cual distribuye el software.
- **Patente** : conjunto de derechos exclusivos garantizados por un gobierno o autoridad al inventor de un nuevo producto (material o inmaterial) susceptible de ser explotado industrialmente para el bien del solicitante por un periodo de tiempo limitado.
- **Derecho de autor o copyright** : forma de protección proporcionada por las leyes vigentes en la mayoría de los países para los autores de obras originales incluyendo obras literarias, dramáticas, musicales, artísticas e intelectuales, tanto publicadas como pendientes de publicar.
- **Software libre** : proporciona libertad de
 - ejecutar el programa, para cualquier propósito
 - estudiar el funcionamiento del programa, y adaptarlo a sus necesidades
 - redistribuir copias
 - mejorar el programa, y poner sus mejoras a disposición del público, para beneficio de toda la comunidad
- **Software de fuente abierta** : sus términos de distribución cumplen los criterios de
 - distribución libre
 - inclusión del código fuente
 - permitir modificaciones y trabajos derivados en las mismas condiciones que el software original

- integridad del código fuente del autor, pudiendo requerir que los trabajos derivados tengan distinto nombre o versión
 - no discriminación a personas o grupos
 - sin uso restringido a campo de actividad
 - los derechos otorgados a un programa serán válidos para todo el software redistribuido sin imponer condiciones complementarias
 - la licencia no debe ser específica para un producto determinado
 - la licencia no debe poner restricciones a otro producto que se distribuya junto con el software licenciado
 - la licencia debe ser tecnológicamente neutral
- **Estándar abierto** : basado en los principios de
 - disponibilidad
 - maximizar las opciones del usuario final
 - sin tasas sobre la implementación
 - sin discriminación de implementador
 - permiso de extensión o restricción
 - evitar prácticas predadoras por fabricantes dominantes
 - **Software de dominio público** : aquél que no está protegido con copyright.
 - **Software con copyleft** : software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando lo redistribuyen o modifican, o sea, la versión modificada debe ser también libre.
 - **Software semi-libre** : aquél que no es libre, pero viene con autorización de usar, copiar, distribuir y modificar para particulares sin fines de lucro.
 - **Freeware** : se usa comúnmente para programas que permiten la redistribución pero no la modificación (y su código fuente no está disponible).
 - **Shareware** : software con autorización de redistribuir copias, pero debe pagarse cargo por licencia de uso continuado.
 - **Software privativo** : aquél cuyo uso, redistribución o modificación están prohibidos o necesitan una autorización.
 - **Software comercial** : el desarrollado por una empresa que pretende ganar dinero por su uso.

Desarrollos de software libre

- **Motivación ética** : abanderada por la Free Software Foundation -partidaria del apelativo *libre* -, que argumenta que el software es conocimiento, debe poderse difundir sin trabas y que su ocultación es una actitud antisocial y que la posibilidad de modificar programas es una forma de libertad de expresión.
- **Motivación pragmática** : abanderada por la Open Source Initiative -partidaria del apelativo *fuente abierta* -, que argumenta ventajas técnicas y económicas, apartando el término “free” para poder evitar así la posible confusión entre “libre” y “gratis”.

Tipos de licencias

La siguiente tabla va a mostrar una comparativa de las licencias más importantes para software no propietario, indicando si son compatibles con la licencia GNU (GPL) y si están aprobadas por la Open Source Initiative.

Nombre	Descripción	Compat. GNU [7]	Certific. OSI [8]
Academic Free (AFL)	Libre, sin copyleft, con patentes.	No	Hasta 2.1
Apache Software	Libre y abierta, con patentes.	No	Sí
Apple Public Software (APSL)	Libre, permite enlazar con ficheros propietarios.	No	Sí
Artistic	Puede agregarse a software comercial (licencia de Perl).	No	Sí
Clarified Artistic o Artistic 2	Libre, abierta, corrige los problemas de la versión 1.	Sí	Sí
BSD Modificada	Simple, libre, abierta	Sí	Sí
BSD Original (BSD)	Permisiva, sin copyleft, con cláusula de advertencia.	No	No
Common Development and Distribution (CDDL)	Libre, sin copyleft, con patentes, con propiedad intelectual.	No	Sí
Common Public (CPL)	Libre, con patentes.	No	Sí
Dominio Público	Estado sin registrar (sin licencia), permisivo, sin copyleft.	Sí	-
Eclipse Public (EPL)	Libre, con patentes (menos agresiva que CPL).	No	Sí
Eiffel Forum (EFL)	Libre y abierta (la versión 1 no es compatible con GPL).	v2	Sí
EU DataGrid Software	Libre, permisiva, sin copyleft.	Sí	Sí
Expat	Libre, simple, permisiva y si copyleft (similar a la MIT).	Sí	Sí
GNU Public (GPL)	Libre, abierta, con copyleft.	Sí	Sí
GNU Reducida (LGPL)	GPL sin copyleft, permite enlazar con módulos no libres.	Sí	Sí
IBM Public	Libre, con patentes.	No	Sí
Intel Open Software	Libre (ha dejado de usarse).	Sí	Sí
Jabber	Libre, abierta, no permite relicenciar en GPL).	No	Sí
Lucent Public (Plan9)	Libre, incompatible GPL.	No	Sí
MIT/X Window	Libre, permisiva, copyleft limitado.	Sí	Sí
Mozilla Public (MPL)	Libre, copyleft limitado, no enlazable con GPL.,	No	Sí
Netscape Public (NPL)	Como MPL pero puede usar código propietario.	No	No
Nokia Open Source	Similar a MPL.	No	Sí
OpenLDAP	Libre, permisiva, sin copyleft.	v2.7	No
Open Software (OSL)	Libre, abierta, con copyleft reducido (según FSF).	No	Sí
Perl	Licencia dual AL/GPL.	Sí	
PHP	Libre, sin copyleft (similar a BSD Original).	No	Sí
Python	Libre (compatible GPL).	Sí	Sí
Q Public (QPL)	Libre, sin copyleft, no enlazable con GPL salvo explícito (Qt abierto usa GPL).	No	Sí
Reciprocal Public	No gratuito, notificación de modificaciones al	No	Sí

Elección del tipo de licencia

Diferencias entre licenciar y relicenciar versiones de una aplicación original en 3 tipos de licencias más usadas en software libre: BSD, GPL y MPL.

- **BSD** : una aplicación licenciada con BSD permite que otras versiones pueden tener otros tipos de licencias, tanto propietarias, como BSD o GPL.
- **GPL** : esta licencia aplica la necesidad de *copyleft*, haciendo que las nuevas versiones de la aplicación sean siempre libres y licenciadas bajo GPL.
- **MPL** : aplica licencias dobles al código fuente y a los ejecutables, obligando a devolver al autor los fuentes modificados y permitiendo licenciar los binarios como propietarios.

Técnicas de evaluación de alternativas y análisis de viabilidad. Personal, procedimientos, datos, software, hardware. Presupuestación y control de costes de un proyecto informático.

Introducción

Mientras que el Plan de Sistemas de Información tiene como objetivo proporcionar un marco estratégico que sirva de referencia para los Sistemas de Información de un ámbito concreto de una organización, el objetivo del Estudio de Viabilidad del Sistema es el análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo, que tenga en cuenta restricciones económicas, técnicas, legales y operativas. La solución obtenida como resultado del estudio puede ser la definición de uno o varios proyectos que afecten a uno o varios sistemas de información ya existentes o nuevos. Para ello, se identifican los requisitos que se ha de satisfacer y se estudia, si procede, la situación actual.

A partir del estado inicial, la situación actual y los requisitos planteados, se estudian las alternativas de solución. Dichas alternativas pueden incluir soluciones que impliquen desarrollos a medida, soluciones basadas en la adquisición de productos software del mercado o soluciones mixtas. Se describe cada una de las alternativas, indicando los requisitos que cubre.

Una vez descritas cada una de las alternativas planteadas, se valora el impacto en la organización, la inversión a realizar en cada caso y los riesgos asociados. Esta información se analiza con el fin de evaluar las distintas alternativas y seleccionar la más adecuada, definiendo y estableciendo su planificación.

Análisis de Viabilidad

El propósito de este proceso es analizar un conjunto concreto de necesidades, con la idea de proponer una solución a corto plazo. Los criterios con los que se hace esta propuesta no serán estratégicos sino tácticos y relacionados con aspectos económicos, técnicos, legales y operativos.

Los resultados del Estudio de Viabilidad del Sistema constituirán la base para tomar la decisión de seguir adelante o abandonar. Si se decide seguir adelante pueden surgir uno q₁₄₉^{B2}

varios proyectos que afecten a uno o varios sistemas de información. Dichos sistemas se desarrollarán según el resultado obtenido en el estudio de viabilidad y teniendo en cuenta la cartera de proyectos para la estrategia de implantación del sistema global.

Se ha considerado que este proceso es obligatorio, aunque el nivel de profundidad con el que se lleve a cabo dependerá de cada caso. La conveniencia de la realización del estudio de la situación actual depende del valor añadido previsto para la especificación de requisitos y para el planteamiento de alternativas de solución. En las alternativas se considerarán soluciones "a medida", soluciones basadas en la adquisición de productos software del mercado o soluciones mixtas.

Para valorar las alternativas planteadas y determinar una única solución, se estudiará el impacto en la organización de cada una de ellas, la inversión y los riesgos asociados.

El resultado final de este proceso son los productos relacionados con la solución que se propone para cubrir la necesidad concreta que se planteó en el proceso, y depende de si la solución conlleva desarrollo a medida o no:

- Contexto del sistema (con la definición de las interfaces en función de la solución).
- Impacto en la organización.
- Coste / beneficio de la solución.
- Valoración de riesgos de la solución.
- Enfoque del plan de trabajo de la solución.
- Planificación de la solución.

Solución propuesta:

- Descripción de la solución.
- Modelo de descomposición en subsistemas.
- Matriz de procesos / localización geográfica.
- Matriz de datos / localización geográfica. Entorno tecnológico y comunicaciones.
- Estrategia de implantación global del sistema.
- Descripción de los procesos manuales.

Si la alternativa incluye desarrollo:

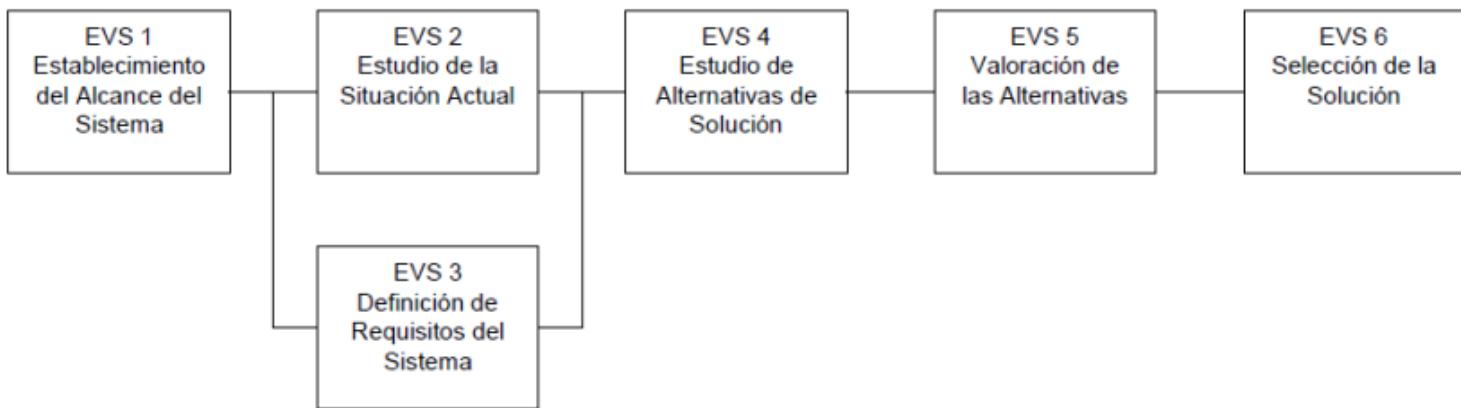
- Modelo abstracto de datos / modelo de procesos.
- Modelo de negocio / modelo de dominio.

Si la alternativa incluye un producto software estándar de mercado:

- Descripción del producto.
- Evolución del producto.
- Costes ocasionados por el producto.
- Estándares del producto.
- Descripción de adaptación si es necesaria.

Si en la organización se ha realizado con anterioridad un Plan de Sistemas de Información que afecte al sistema objeto de este estudio, se dispondrá de un conjunto de productos que proporcionarán información a tener en cuenta en todo el proceso.

Las actividades que engloba este proceso se recogen en la siguiente figura, en la que se indican las actividades que pueden ejecutarse en paralelo y las que precisan para su realización resultados originados en actividades anteriores.



Actividad EVS 1: Establecimiento del alcance del sistema

En esta actividad se estudia el alcance de la necesidad planteada por el cliente o usuario, o como consecuencia de la realización de un PSI, realizando una descripción general de la misma. Se determinan los objetivos, se inicia el estudio de los requisitos y se identifican las unidades organizativas afectadas estableciendo su estructura.

Se analizan las posibles restricciones, tanto generales como específicas, que puedan condicionar el estudio y la planificación de las alternativas de solución que se propongan.

Si la justificación económica es obvia, el riesgo técnico bajo, se esperan pocos problemas legales y no existe ninguna alternativa razonable, no es necesario profundizar en el estudio de viabilidad del sistema, analizando posibles alternativas y realizando una valoración y evaluación de las mismas, sino que éste se orientará a la especificación de requisitos, descripción del nuevo sistema y planificación.

Se detalla la composición del equipo de trabajo necesario para este proceso y su planificación. Finalmente, con el fin de facilitar la implicación activa de los usuarios en la definición del sistema, se identifican sus perfiles, dejando claras sus tareas y responsabilidades.

Tarea		Productos	Técnicas y Prácticas	Participantes
EVS 1.1	Estudio de la Solicitud	<ul style="list-style-type: none"> - Descripción General del Sistema - Catálogo Objetivos EVS - Catálogo de Requisitos 	<ul style="list-style-type: none"> - Catalogación - Sesiones de trabajo 	<ul style="list-style-type: none"> - Comité de Dirección - Jefe de Proyecto - Analistas
EVS 1.2	Identificación del Alcance del Sistema	<ul style="list-style-type: none"> - Descripción General del Sistema: <ul style="list-style-type: none"> o Contexto del Sistema o Estructura Organizativa - Catálogo de Requisitos - Catálogo de Usuarios 	<ul style="list-style-type: none"> - Diagrama de Flujo de Datos - Diagrama de Descomposición Funcional - Catalogación - Sesiones de trabajo 	<ul style="list-style-type: none"> - Comité de Dirección - Jefe de Proyecto - Analistas
EVS 1.3	Especificación del Alcance del EVS	<ul style="list-style-type: none"> - Catálogo de Objetivos del EVS - Catálogo de Usuarios - Plan de Trabajo 	<ul style="list-style-type: none"> - Catalogación - Sesiones de trabajo 	<ul style="list-style-type: none"> - Comité de Dirección - Jefe de Proyecto - Analistas

Tarea EVS 1.1: Estudio de la solicitud

Se realiza una descripción general de la necesidad planteada por el usuario, y se estudian las posibles restricciones de carácter económico, técnico, operativo y legal que puedan afectar al sistema. Antes de iniciar el estudio de los requisitos del sistema se establecen los objetivos generales del Estudio de Viabilidad, teniendo en cuenta las restricciones identificadas anteriormente.

Si el sistema objeto de estudio se encuentra en el ámbito de un Plan de Sistemas de Información vigente, se debe tomar como referencia el catálogo de requisitos y la arquitectura de información resultante del mismo, como información adicional para la descripción general del sistema y determinación de los requisitos iniciales.

Productos

- De entrada
 - Catálogo de Requisitos del PSI (PSI 9.2)
 - Arquitectura de Información (PSI 9.2)
 - Solicitud (externo)
- De salida
 - Descripción General del Sistema
 - Catálogo de Objetivos del EVS
 - Catálogo de Requisitos

Prácticas

- Catalogación
- Sesiones de trabajo

Participantes

- Comité de Dirección
- Jefe de Proyecto
- Analistas

Tarea EVS 1.2: Identificación del alcance del sistema

Se analiza el alcance de la necesidad planteada y se identifican las restricciones relativas a la sincronización con otros proyectos, que puedan interferir en la planificación y futura puesta a punto del sistema objeto del estudio. Esta información se recoge en el catálogo de requisitos.

Si el sistema pertenece al ámbito de un Plan de Sistemas de Información, se debe tener en cuenta la arquitectura de información propuesta para analizar el alcance del sistema e identificar los sistemas de información que quedan fuera del ámbito del estudio. Además, se estudia el plan de proyectos, para determinar las posibles dependencias con otros proyectos.

Una vez establecido el alcance, se identifican las unidades organizativas afectadas por el sistema, así como su estructura y responsables de las mismas. Para determinar los responsables se tiene en cuenta a quiénes afecta directamente y quiénes pueden influir en el éxito o fracaso del mismo.

Productos

- De entrada
 - Plan de Proyectos (PSI 9.2)
 - Arquitectura de Información (PSI 9.2)
 - Descripción General del Sistema (EVS 1.1)
 - Catálogo de Objetivos del EVS (EVS 1.1)
 - Catálogo de Requisitos (EVS 1.1)
- De salida
 - Descripción General del Sistema:
 - Contexto del Sistema
 - Estructura Organizativa

- Catálogo de Requisitos:
 - Requisitos Relativos a Restricciones o Dependencias con Otros Proyectos
- Catálogo de Usuarios

Técnicas

- Diagrama de Flujo de Datos
- Diagrama de Descomposición Funcional

Prácticas

- Catalogación
- Sesiones de trabajo

Participantes

- Comité de Dirección
- Jefe de Proyecto
- Analistas

Tarea EVS 1.3: Especificación del alcance del EVS

En función del alcance del sistema y los objetivos del Estudio de Viabilidad del Sistema, se determinan las actividades y tareas a realizar. En particular, hay que decidir si se realiza o no el estudio de la situación actual y, en el caso de considerarlo necesario, con qué objetivo. Si el sistema pertenece al ámbito de un Plan de Sistemas de Información, los criterios que pueden orientar sobre la necesidad de llevar a cabo el estudio de la situación actual dependen de la arquitectura de información propuesta, en cuanto a la identificación de los sistemas de información actuales, implicados en el ámbito de este estudio, que se haya decidido conservar.

Se identifican los usuarios participantes de las distintas unidades organizativas afectadas para la realización del Estudio de Viabilidad del Sistema, determinando previamente sus perfiles y responsabilidades.

Debe comunicarse el plan de trabajo a los usuarios identificados como implicados en el Estudio de Viabilidad, solicitando su aceptación y esperando su confirmación.

Productos

- De entrada
 - Arquitectura de Información (PSI 9.2)
 - Catálogo de Objetivos del EVS (EVS 1.1)
 - Descripción General del Sistema (EVS 1.2)
 - Catálogo de Usuarios (EVS 1.2)
- De salida
 - Catálogo de Objetivos del EVS:
 - Objetivos del Estudio de la Situación Actual
 - Catálogo de Usuarios
 - Plan de Trabajo

Prácticas

- Catalogación
- Sesiones de trabajo

Participantes

- Comité de Dirección
- Jefe de Proyecto

- Analistas

Actividad EVS 2: Estudio de la situación actual

La situación actual es el estado en el que se encuentran los sistemas de información existentes en el momento en el que se inicia su estudio. Teniendo en cuenta el objetivo del estudio de la situación actual, se realiza una valoración de la información existente acerca de los sistemas de información afectados. En función de dicha valoración, se especifica el nivel de detalle con que se debe llevar a cabo el estudio. Si es necesario, se constituye un equipo de trabajo específico para su realización y se identifican los usuarios participantes en el mismo.

Si se decide documentar la situación actual, normalmente es conveniente dividir el sistema actual en subsistemas. Si es posible se describirá cada uno de los subsistemas, valorando qué información puede ser relevante para la descripción.

Como resultado de esta actividad se genera un diagnóstico, estimando la eficiencia de los sistemas de información existentes e identificando los posibles problemas y las mejoras.

Tarea	Productos	Técnicas y Prácticas	Participantes
EVS 2.1 Valoración del Estudio de la Situación Actual	- Descripción de la Situación Actual: o Contexto del Sistema Actual o Descripción de los Sistemas de Información Actuales	- Diagrama de Flujo de Datos - Diagrama de Representación - Sesiones de Trabajo	- Jefe de Proyecto - Analistas - Directores de Usuarios
EVS 2.2 Identificación de los Usuarios Participantes en el Estudio de la Situación Actual	- Catálogo Usuarios	- Sesiones de Trabajo - Catalogación	- Jefe de Proyecto - Directores de Usuarios
EVS 2.3 Descripción de los Sistemas de Información Existentes	- Descripción de la Situación Actual: o Descripción Lógica del Sistema Actual o Modelo Físico del Sistema Actual (opcional) o Matriz Localización Módulos y Datos	- Modelo Entidad /Relación Extendido - Diagrama de Flujo de Datos - Diagrama de Clases - Diagrama de Interacción de Objetos - Matricial - Diagrama de Representación - Sesiones de Trabajo	- Analistas - Usuarios expertos - Equipo de Soporte Técnico
EVS 2.4 Realización del Diagnóstico de la Situación Actual	- Descripción de la Situación Actual: o Diagnóstico de la Situación Actual		- Analistas - Responsable de Mantenimiento

Tarea EVS 2.1: Valoración del estudio de la situación actual

En función de los objetivos establecidos para el estudio de la situación actual, y considerando el contexto del sistema especializado en la descripción general del mismo, se identifican los sistemas de información existentes que es necesario analizar con el fin de determinar el alcance del sistema actual. Asimismo, se decide el nivel de detalle con el que se va a llevar a cabo el estudio de cada uno de los sistemas de información implicados. En el caso de haber realizado un Plan de Sistemas de Información que afecte a dicho sistema, se toma como punto de partida para este análisis la arquitectura de información propuesta.

Para poder abordar el estudio, se realiza previamente una valoración de la información existente acerca de los sistemas de información afectados por el EVS. Se debe decidir si se realizan o no los modelos lógicos del sistema actual o si se describe el modelo físico, en función de los siguientes criterios:

- Si existen los modelos lógicos, y son fiables, se utilizan en la tarea Descripción de los Sistemas de Información Existentes (EVS 2.3).
- Si no existen dichos modelos, o no son fiables, se considera el tiempo de vida estimado para el sistema de información en función de la antigüedad, la obsolescencia de la tecnología o la falta de adecuación funcional para determinar si se obtienen los modelos lógicos y físicos del sistema actual o por el contrario no se elabora ningún modelo.

La información relativa a los sistemas de información que se decida analizar, se obtiene mediante sesiones de trabajo con los Directores de Usuarios y el apoyo de los profesionales de Sistemas y Tecnologías de la Información y Comunicaciones (STIC) que se considere necesario.

Productos

- De entrada
 - Información Existente del Sistema Actual (externo)
 - Arquitectura de Información (PSI 9.2)
 - Catálogo de Objetivos del EVS (EVS 1.3)
 - Descripción General del Sistema (EVS 1.2)
- De salida
 - Descripción de la Situación Actual:
 - Contexto del Sistema Actual
 - Descripción de los Sistemas de Información Actuales

Técnicas

- Diagrama de Flujo de Datos

Prácticas

- Diagrama de Representación
- Sesiones de Trabajo

Participantes

- Jefe de Proyecto
- Analistas
- Directores de Usuarios

Tarea EVS 2.2: Identificación de usuarios participantes en el estudio de la situación actual

En función del nivel de detalle establecido para el estudio de la situación actual, se identifican los usuarios participantes de cada una de las unidades organizativas afectadas por dicho estudio. Se informa a los usuarios implicados en el Estudio de la Situación Actual, se solicita su aceptación y se espera su confirmación.

Productos

- De entrada
 - Descripción General del Sistema (EVS 1.2)
 - Catálogo de Usuarios (EVS 1.3)
 - Descripción de la Situación Actual (EVS 2.1)

- De salida
 - Catálogo de Usuarios

Prácticas

- Catalogación
- Sesiones de Trabajo

Participantes

- Jefe de Proyecto
- Directores de Usuarios

Tarea EVS 2.3: Descripción de los sistemas de información existentes

En esta tarea se describen los sistemas de información existentes afectados, según el alcance y nivel de detalle establecido en la tarea Valoración del Estudio de la Situación Actual (EVS 2.1), mediante sesiones de trabajo con los usuarios designados para este estudio.

Si se ha decidido describir los sistemas a nivel lógico, y si existe un conocimiento suficiente de los sistemas de información a especificar, puede hacerse directamente, aplicando las técnicas de modelización y siguiendo un método descendente. Si no se dispone del conocimiento suficiente, se construyen los modelos a partir de la descripción del modelo físico, es decir, de forma ascendente.

Si se tiene que describir el modelo físico, se puede hacer mediante un Diagrama de Representación en el que se recojan todos los componentes físicos y sus referencias cruzadas. Otra opción es describir el modelo físico de forma más detallada, para lo que es necesaria la utilización de herramientas de tipo scanner.

Es conveniente indicar la localización geográfica y física actual de los módulos y datos de los sistemas de información afectados, evaluando al mismo tiempo la redundancia en las distintas unidades organizativas.

Productos

- De entrada
 - Descripción de la Situación Actual (EVS 2.1)
 - Catálogo de Usuarios (EVS 2.2)
- De salida
 - Descripción de la Situación Actual:
 - Descripción Lógica del Sistema Actual
 - Modelo Físico del Sistema Actual (opcional)
 - Matriz de Localización Geográfica y Física de Módulos y Datos, incluidas las redundancias

Técnicas

- Diagrama de Flujo de Datos
- Modelo Entidad / Relación extendido
- Diagrama de Clases
- Diagrama de Interacción de Objetos
- Matricial

Prácticas

- Diagrama de Representación
- Sesiones de Trabajo

Participantes

- Analistas
- Usuarios Expertos
- Equipo de Soporte Técnico

Tarea EVS 2.4: Realización del diagnóstico de la situación actual

Con el fin de elaborar el diagnóstico de la situación actual se analiza la información de los sistemas de información existentes, obtenida en la tarea anterior y se identifican problemas, deficiencias y mejoras. Estas últimas deben tenerse en cuenta en la definición de los requisitos.

En el caso de haber realizado un Plan de Sistemas de Información, se considera la valoración realizada sobre los sistemas de información actuales que pertenecen al ámbito de este estudio.

Si se ha tomado la decisión de no describir la situación actual, se realiza un diagnóstico global justificando esta decisión

Productos

- De entrada
 - Descripción de la Situación Actual (EVS 2.3)
 - Catálogo de Objetivos del EVS (EVS 1.3)
 - Valoración de la Situación actual (EVS 5.3)
- De salida
 - Descripción de la Situación Actual:
 - Diagnóstico de Situación Actual

Participantes

- Analistas
- Responsable de Mantenimiento

Actividad EVS 3: Definición de requisitos del sistema

Esta actividad incluye la determinación de los requisitos generales, mediante una serie de sesiones de trabajo con los usuarios participantes, que hay que planificar y realizar. Una vez finalizadas, se analiza la información obtenida definiendo los requisitos y sus prioridades, que se añaden al catálogo de requisitos que servirá para el estudio y valoración de las distintas alternativas de solución que se propongan.

Tarea	Productos	Técnicas y Prácticas	Participantes
EVS 3.1	Identificación de las Directrices Técnicas y de Gestión	- Catálogo de Normas	- Jefe de Proyecto - Analistas - Usuarios Expertos
EVS 3.2	Identificación de Requisitos	- Identificación de Requisitos	- Jefe de Proyecto - Analistas - Usuarios Expertos
EVS 3.3	Catalogación de Requisitos	- Catálogo de Requisitos	- Jefe de Proyecto - Analistas - Usuarios Expertos

Tarea EVS 3.1: Identificación de las directrices técnicas y de gestión

La realización de esta tarea permite considerar los términos de referencia para el sistema en estudio desde el punto de vista de directrices tanto técnicas como de gestión. Si el

sistema en estudio pertenece al ámbito de un Plan de Sistemas de Información vigente, éste proporciona un marco de referencia a considerar en esta tarea.

Con este fin, se recoge información sobre los estándares y procedimientos que deben considerarse al proponer una solución, relativos a:

- Políticas técnicas:
 - Gestión de Proyectos (seguimiento, revisión y aprobación final).
 - Desarrollo de Sistemas (existencia de normativas, metodologías y técnicas de programación).
 - Arquitectura de Sistemas (centralizada, distribuida).
- Política de Seguridad (control de accesos, integridad de datos, disponibilidad de aplicaciones).
- Directrices de Planificación.
- Directrices de Gestión de Cambios.
- Directrices de Gestión de Calidad.

Productos

- De entrada
 - Catálogo de Normas del PSI (PSI 3.2)
 - Recopilación de Directrices Técnicas y de Gestión (externo)
- De salida
 - Catálogo de Normas

Prácticas

- Catalogación

Participantes

- Jefe de Proyecto
- Analistas
- Usuarios Expertos

Tarea EVS 3.2: Identificación de Requisitos

Para la obtención de las necesidades que ha de cubrir el sistema en estudio, se debe decidir qué tipo de sesiones de trabajo se realizarán y con qué frecuencia tendrán lugar, en función de la disponibilidad de los usuarios participantes.

Si se ha realizado el Estudio de la Situación Actual (EVS 2), puede ser conveniente seleccionar la información de los sistemas de información existentes que resulte de interés para el desarrollo de dichas sesiones de trabajo.

Una vez establecidos los puntos anteriores, se planifican las sesiones de trabajo con los usuarios participantes identificados al estudiar el alcance del Estudio de Viabilidad del Sistema (EVS 1.3), y se realizan de acuerdo al plan previsto. La información obtenida depende del tipo de sesión de trabajo seleccionado.

Productos

- De entrada
 - Descripción General del Sistema (EVS 1.2)
 - Catálogo de Requisitos (EVS 1.2)
 - Equipo de Trabajo del EVS (EVS 1.3)
 - Catálogo de Usuarios (EVS 2.2/1.3)
 - Descripción de la Situación Actual (EVS 2.4)
- De salida
 - Identificación de Requisitos

Prácticas

- Sesiones de Trabajo

Participantes

- Jefe de Proyecto
- Analistas
- Usuarios Expertos

Tarea EVS 3.3: Catalogación de Requisitos

Se analiza la información obtenida en las sesiones de trabajo para la Identificación de Requisitos, definiendo y catalogando los requisitos (funcionales y no funcionales) que debe satisfacer el sistema, indicando sus prioridades.

Se incluirán también requisitos relativos a distribución geográfica y entorno tecnológico.

Productos

- De entrada
 - Identificación de Requisitos (EVS 3.2)
 - Catálogo de Requisitos (EVS 1.2)
- De salida
 - Catálogo de Requisitos

Prácticas

- Catalogación

Participantes

- Jefe de Proyecto
- Analistas
- Usuarios Expertos

Actividad EVS 4: Estudio de alternativas de solución

Este estudio se centra en proponer diversas alternativas que respondan satisfactoriamente a los requisitos planteados, considerando también los resultados obtenidos en el Estudio de la Situación Actual (EVS 2), en el caso de que se haya realizado.

Teniendo en cuenta el ámbito y funcionalidad que debe cubrir el sistema, puede ser conveniente realizar, previamente a la definición de cada alternativa, una descomposición del sistema en subsistemas.

En la descripción de las distintas alternativas de solución propuestas, se debe especificar si alguna de ellas está basada, total o parcialmente, en un producto existente en el mercado. Si la alternativa incluye un desarrollo a medida, se debe incorporar en la descripción de la misma un modelo abstracto de datos y un modelo de procesos, y en orientación a objetos, un modelo de negocio y un modelo de dominio.

Tarea		Productos	Técnicas y Prácticas	Participantes
EVS 4.1	Preselección de Alternativas de Solución	<ul style="list-style-type: none"> - Descomposición Inicial del Sistema en Subsistemas (opcional) - Alternativas de Solución a Estudiar 	<ul style="list-style-type: none"> - Diagrama de Representación 	<ul style="list-style-type: none"> - Jefe de Proyecto - Analistas - Técnicos de sistemas
EVS 4.2	Descripción de las Alternativas de Solución	<ul style="list-style-type: none"> - Catálogo de Requisitos - Alternativas de solución a estudiar: <ul style="list-style-type: none"> o Catálogo de Requisitos (cobertura) o Modelo de Descomposición en Subsistemas o Matriz Procesos / Localización Geográfica o Matriz Datos / Localización Geográfica o Entorno Tecnológico y Comunicaciones o Estrategia de Implantación Global del Sistema <p>Si la alternativa requiere desarrollo:</p> <ul style="list-style-type: none"> o Modelo Abstracto de Datos / Modelo de Procesos (En caso de <i>Estructurado</i>) o Modelo de Negocio / Modelo de Dominio (En caso de <i>Orientación a Objetos</i>) <p>Si la alternativa incluye producto software estándar:</p> <ul style="list-style-type: none"> o Descripción del Producto o Previsión de Evolución del Producto o Costes Ocasionados por Producto o Estándares del Producto o Descripción de Adaptación (si es necesaria) 	<ul style="list-style-type: none"> - Matricial - Modelo Entidad/ Relación extendido - Diagrama de Flujo de Datos - Casos de Uso - Diagrama de Clases - Catalogación - Diagrama de Representación 	<ul style="list-style-type: none"> - Jefe de Proyecto - Analistas - Usuarios Expertos - Técnicos de sistemas - Responsables de Seguridad - Especialistas en Comunicaciones

Tarea EVS 4.1: Preselección de Alternativas de Solución

Una vez definidos los requisitos a cubrir por el sistema, se estudian las diferentes opciones que hay para configurar la solución. Entre ellas, hay que considerar la adquisición de productos software estándar del mercado, desarrollos a medida o soluciones mixtas.

Dependiendo del alcance del sistema y las posibles opciones, puede ser conveniente realizar inicialmente una descomposición del sistema en subsistemas. Se establecen las posibles alternativas sobre las que se va a centrar el estudio de la solución, combinando las opciones que se consideren más adecuadas.

Productos

- De entrada
 - Información de Productos Software del Mercado (externo)
 - Descripción General del Sistema (EVS 1.2)
 - Descripción de la Situación Actual (EVS 2.4)
 - Catálogo de Requisitos (EVS 3.3)
- De salida
 - Descomposición Inicial del Sistema en Subsistemas (opcional)
 - Alternativas de Solución a Estudiar

Prácticas

- Diagrama de Representación

Participantes

- Jefe de Proyecto
- Analistas
- Técnicos de Sistemas

Tarea EVS 4.2: Descripción de las Alternativas de Solución

Para cada alternativa propuesta, se identifican los subsistemas que cubre y los requisitos a los que se da respuesta. Se deben considerar también aspectos relativos a la cobertura geográfica (ámbito y limitaciones) de procesos y datos, teniendo en cuenta a su vez la gestión de comunicaciones y control de red.

En la definición de cada alternativa, se propone una estrategia de implantación teniendo en cuenta tanto la cobertura global del sistema como la cobertura geográfica. Si la alternativa incluye desarrollo se describe el modelo abstracto de datos y el modelo de procesos, y en el caso de Orientación a Objetos, el modelo de negocio y, opcionalmente, el modelo de dominio. Se propone el entorno tecnológico que se considera más apropiado para la parte del sistema basada en desarrollo y se describen los procesos manuales.

Si la alternativa incluye una solución basada en producto se analiza su evolución prevista, adaptabilidad y portabilidad, así como los costes ocasionados por licencias, y los estándares del producto. Igualmente se valora y determina su entorno tecnológico.

Productos

- De entrada
 - Descripción General del Sistema (EVS 1.2)
 - Descripción de la Situación Actual (EVS 2.4)
 - Catálogo de Requisitos (EVS 3.3)
 - Descomposición Inicial del Sistema en Subsistemas (EVS 4.1) (opcional)
 - Alternativas de Solución a Estudiar (EVS 4.1)
- De salida
 - Catálogo de Requisitos (actualizado)
 - Alternativas de Solución a Estudiar:
 - Catálogo de Requisitos (cobertura)
 - Modelo de Descomposición en Subsistemas
 - Matriz Procesos / Localización Geográfica
 - Matriz Datos / Localización Geográfica
 - Entorno Tecnológico y Comunicaciones
 - Estrategia de Implantación Global del Sistema
 - Descripción de Procesos Manuales
 - Si la alternativa incluye desarrollo:
 - Modelo Abstracto de Datos / Modelo de Procesos
 - Modelo de Negocio / Modelo de Dominio (en caso de Orientación a Objetos)
 - Si la alternativa incluye un producto software estándar de mercado:
 - Descripción del Producto
 - Evolución del Producto
 - Costes Ocasionados por Producto
 - Estándares del Producto
 - Descripción de Adaptación (si es necesaria)

Técnicas

- Matricial
- Diagrama de Flujo de Datos
- Modelo Entidad / Relación extendido
- Diagrama de Clases
- Casos de Uso

Prácticas

- Catalogación
- Diagrama de Representación

Participantes

- Jefe de Proyecto
- Analistas
- Usuarios Expertos
- Técnicos de Sistemas
- Responsable de Seguridad
- Especialistas en Comunicaciones

Actividad EVS 5: Valoración de la alternativas

Una vez descritas las alternativas se realiza una valoración de las mismas, considerando el impacto en la organización, tanto desde el punto de vista tecnológico y organizativo como de operación, y los posibles beneficios que se esperan contrastados con los costes asociados. Se realiza también un análisis de los riesgos, decidiendo cómo enfocar el plan de acción para minimizar los mismos y cuantificando los recursos y plazos precisos para planificar cada alternativa.

Tarea		Productos	Técnicas y Prácticas	Participantes
EVS 5.1	Estudio de la Inversión	<ul style="list-style-type: none">- Valoración de Alternativas:<ul style="list-style-type: none">◦ Impacto en la Organización de Alternativas◦ Coste / Beneficio de Alternativas	<ul style="list-style-type: none">- Análisis Coste / Beneficio	<ul style="list-style-type: none">- Jefe de Proyecto- Analistas-
EVS 5.2	Estudio de los Riesgos	<ul style="list-style-type: none">- Valoración de Alternativas:<ul style="list-style-type: none">◦ Valoración de Riesgos	<ul style="list-style-type: none">- Impacto en la Organización	<ul style="list-style-type: none">- Jefe de Proyecto- Analistas
EVS 5.3	Planificación de Alternativas	<ul style="list-style-type: none">- Plan de Trabajo de Cada Alternativa:<ul style="list-style-type: none">◦ Enfoque del Plan de Trabajo de Cada Alternativa◦ Planificación de Cada Alternativa	<ul style="list-style-type: none">- Planificación	<ul style="list-style-type: none">- Jefe de Proyecto- Analistas

Tarea EVS 5.1: Estudio de la Inversión

Para cada alternativa de solución propuesta, se valora el impacto y se establece su viabilidad económica. Para ello, se realiza un análisis coste/beneficio que determina los costes del sistema y los pondera con los beneficios tangibles, cuantificables directamente, y con los beneficios intangibles, buscando el modo de cuantificarlos.

Productos

- De entrada
 - Alternativas de Solución a Estudiar (EVS 4.2)

- De salida
 - Valoración de Alternativas:
 - Impacto en la Organización de Alternativas
 - Coste / beneficio de Alternativas

Técnicas

- Análisis Coste / Beneficio

Participantes

- Jefe de Proyecto
- Analistas

Tarea EVS 5.2: Estudio de los Riesgos

Para cada alternativa se seleccionan los factores de situación que habrá que considerar, relativos tanto a la incertidumbre como a la complejidad del sistema. Se identifican y valoran los riesgos asociados y se determinan las medidas a tomar para minimizarlos.

Productos

- De entrada
 - Alternativas de Solución a Estudiar (EVS 4.2)
 - Valoración de Alternativas (EVS 5.1)
- De salida
 - Valoración de Alternativas:
 - Valoración de Riesgos

Prácticas

- Impacto en la Organización

Participantes

- Jefe de Proyecto
- Analistas

Tarea EVS 5.3: Planificación de Alternativas

En función del análisis de riesgos realizado en la tarea anterior, y para cada una de las alternativas existentes:

- Se determina el enfoque más adecuado para llevar a buen fin la solución propuesta en cada alternativa.
- Se realiza una planificación, teniendo en cuenta los puntos de sincronismo con otros proyectos en desarrollo o que esté previsto desarrollar, según se ha recogido en el catálogo de requisitos.

De esta manera se garantiza el cumplimiento del plan de trabajo en los restantes procesos del ciclo de vida.

Productos

- De entrada
 - Catálogo de Requisitos (EVS 4.2)
 - Alternativas de Solución a Estudiar (EVS 4.2)
 - Valoración de Alternativas (EVS 5.2)

- De salida
 - Plan de Trabajo de Cada Alternativa:
 - Enfoque del Plan de Trabajo de Cada Alternativa
 - Planificación de Cada Alternativa

Técnicas

- Planificación

Participantes

- Jefe de Proyecto
- Analistas

Actividad EVS 6: Selección de la solución

Antes de finalizar el Estudio de Viabilidad del Sistema, se convoca al Comité de Dirección para la presentación de las distintas alternativas de solución, resultantes de la actividad anterior. En dicha presentación, se debaten las ventajas de cada una de ellas, incorporando las modificaciones que se consideren oportunas, con el fin de seleccionar la más adecuada. Finalmente, se aprueba la solución o se determina su inviabilidad.

Tarea		Productos	Técnicas y Prácticas	Participantes
EVS 6.1	Convocatoria de Presentación	- Plan de Presentación de Alternativas	- Presentación	- Jefe de Proyecto
EVS 6.2	Evaluación de Alternativas y Selección	<ul style="list-style-type: none"> - Plan de Presentación de Alternativas - Catálogo de Requisitos - Solución Propuesta: <ul style="list-style-type: none"> ◦ Descripción de la Solución ◦ Contexto del Sistema (con la definición de las interfaces) ◦ Impacto en Organización de la Solución ◦ Coste / Beneficio de la Solución ◦ Valoración de Riesgos de la Solución ◦ Enfoque del Plan de Trabajo de la Solución ◦ Planificación de la Solución 	<ul style="list-style-type: none"> - Presentación - Sesiones de Trabajo 	<ul style="list-style-type: none"> - Comité de Dirección - Jefe de Proyecto - Analistas
EVS 6.3	Aprobación de la Solución	- Aprobación de la Solución		<ul style="list-style-type: none"> - Comité de Dirección - Jefe de Proyecto

Tarea EVS 6.1: Convocatoria de la Presentación

Se efectúa la convocatoria de la presentación de las distintas alternativas propuestas, adjuntando los productos de la actividad anterior con el fin de que el Comité de Dirección pueda estudiar previamente su contenido. Se espera confirmación por parte del Comité de Dirección de las alternativas a presentar.

Productos

- De entrada
 - Catálogo de Usuarios (EVS 2.2/1.3)
 - Alternativas de Solución a Estudiar (EVS 4.2)
 - Valoración de Alternativas (EVS 5.2)
 - Plan de Trabajo de Cada Alternativa (EVS 5.3)
- De salida
 - Plan de Presentación de Alternativas

Prácticas

- Presentación

Participantes

- Jefe de Proyecto

Tarea EVS 6.2: Evaluación de las Alternativas y Selección

Una vez recibida la confirmación de qué alternativas van a ser presentadas para su valoración, se efectúa su presentación al Comité de Dirección, debatiendo sobre las ventajas e inconvenientes de cada una de ellas y realizando las modificaciones que sugiera dicho Comité, hasta la selección de la solución final.

Productos

- De entrada
 - Descripción General del Sistema (Contexto del Sistema) (EVS 1.2)
 - Catálogo de Requisitos (EVS 4.2)
 - Alternativas de Solución a Estudiar (EVS 4.2)
 - Valoración de Alternativas (EVS 5.2)
 - Plan de Trabajo de Cada Alternativa (EVS 5.3)
 - Plan de Presentación de Alternativas (EVS 6.1)
- De salida
 - Plan de Presentación de Alternativas
 - Catálogo de Requisitos (Actualizado en Función de la Cobertura de la Solución)
 - Solución Propuesta:
 - Descripción de la Solución:
 - Modelo de Descomposición en Subsistemas
 - Matriz Procesos / Localización Geográfica
 - Matriz Datos / Localización Geográfica
 - Entorno Tecnológico y Comunicaciones
 - Estrategia de Implementación Global del Sistema
 - Descripción de Procesos Manuales
 - Si la alternativa incluye desarrollo:
 - Modelo Abstracto de Datos / Modelo de Procesos
 - Modelo de Negocio / Modelo de Dominio
 - Si la alternativa incluye un producto software estándar del mercado:
 - Descripción del Producto
 - Evolución del Producto
 - Costes Ocasionados por Producto
 - Estándares del Producto
 - Descripción de Adaptación (si es necesaria)
 - Contexto del Sistema (con la Definición de las Interfaces en Función de la Solución)
 - Impacto en la Organización de la Solución
 - Coste / Beneficio de la Solución
 - Valoración de Riesgos de la Solución
 - Enfoque del Plan de Trabajo de la Solución
 - Planificación de la Solución

Prácticas

- Presentación
- Sesiones de Trabajo

Participantes

- Comité de Dirección
- Jefe de Proyecto
- Analistas

Tarea EVS 6.3: Aprobación de la Solución

El Comité de Dirección da su aprobación formal o determina la inviabilidad del sistema, por motivos económicos, de funcionalidad como resultado del incumplimiento de los requisitos identificados en plazos razonables o de cobertura de los mismos, etc.

Productos

- De entrada
 - Catálogo de Requisitos (EVS 6.2)
 - Solución Propuesta (EVS 6.2)
- De salida
 - Aprobación de la Solución

Participantes

- Comité de Dirección
- Jefe de Proyecto

Técnicas de Evaluación de Alternativas

Técnicas de Análisis Coste / Beneficio

Objetivos

La técnica de análisis coste/beneficio tiene como objetivo fundamental proporcionar una medida de los costes en que se incurre en la realización de un proyecto y comparar dichos costes previstos con los beneficios esperados de la realización de dicho proyecto.

Esta medida o estimación servirá para:

- Valorar la necesidad y oportunidad de acometer la realización del proyecto.
- Seleccionar la alternativa más beneficiosa para la realización del proyecto.
- Estimar adecuadamente los recursos económicos necesarios en el plazo de realización del proyecto.

Es de destacar la necesidad cada vez mayor de guiarse por criterios económicos y no sólo técnicos para la planificación de trabajos y proyectos. Por ello se hace una primera introducción sobre las técnicas y métodos de evaluación de conceptos económicos, con el fin de proporcionar a los profesionales criterios que les ayuden en la planificación de proyectos y evaluación de alternativas.

Conceptos

Punto de amortización (Break-Even Point)

Es el momento en el tiempo en que el conjunto de beneficios obtenidos por la explotación del nuevo sistema iguala al conjunto de costes de todo tipo que ha ocasionado. A partir del punto de amortización (Break-Even Point), el sistema entra en fase de aportar beneficios netos a la organización.

Periodo de amortización (PayBack)

Es el periodo de tiempo que transcurre desde que los costes con máximos hasta que se alcanza el punto de amortización (Break-Even Point), es decir, en cuanto el sistema empieza a aportar beneficios. Cuanto menor sea el periodo de amortización (Payback) de un Sistema, más atractivo será para la organización acometer su implantación.

Retorno de Inversión - ROI (Return of Investment)

Es el rendimiento de la inversión expresada en términos de porcentaje. Se calcula mediante la fórmula siguiente:

$$ROI = 100 \times (\text{Beneficio Neto Anual} - \text{Coste Desarrollo Anualizado}) / \text{Inversión Promedio}$$

Siendo:

- **Beneficio Neto Anual** : Beneficio neto que aporta el sistema como consecuencia de su uso, es decir los beneficios obtenidos más los gastos no incurridos. Deben restársele los gastos operacionales anuales y los de mantenimiento del sistema.
- **Coste Desarrollo Anualizado** : Total del coste inicial de desarrollo del sistema, dividido por los años que se supone que va a ser operativo.
- **Inversión Promedio** : Total de la inversión realizada (costes de desarrollo, hardware, software, etc.) dividido por dos.

Descripción

Para la realización del análisis coste/beneficio se seguirán los siguientes pasos:

Producir estimaciones de costes/beneficios

Se realizará una lista de todo lo que es necesario para implementar el sistema y una lista de los beneficios esperados del nuevo sistema. En un análisis de costes y beneficios se deberán considerar aquellos aspectos tangibles, es decir, medibles en valores como dinero, tiempo, etc, y no tangibles, es decir, no ponderables de una forma objetiva. En general, los costes suelen ser medibles y estimables en unidades económicas, no así en cuanto a los beneficios, los cuales pueden ser tangibles o no tangibles.

Entre los beneficios no tangibles pueden estar:

- El aumento de cuentas debido a un mejor servicio a los clientes.
- La mejora en la toma de decisiones debido a una mejora en el soporte informático.

La valoración de dichos beneficios se deberá estimar de una forma subjetiva y será realizada por las áreas correspondientes.

A menudo es conveniente dividir los costes estimados a lo largo del proyecto, para ofrecer una información más detallada de la distribución de los recursos de cara a la dirección.

En la estimación de costes se considerarán, los siguientes aspectos:

- **Adquisición de hardware y software** : El que sea preciso para el desarrollo, implantación y normal funcionamiento del sistema. Se debe considerar la saturación de máquinas o sistemas actuales como consecuencia de la entrada en vigor del nuevo sistema.
- **Gastos de mantenimiento de hardware y software** anteriores.
- **Gastos de comunicaciones** : Líneas, teléfonos, correo, etc.
- **Gastos de instalación** : Cableado, acondicionamiento de sala, recursos humanos y materiales, gastos de viaje, etc.
- **Coste de desarrollo** del sistema.
- **Gastos del mantenimiento del sistema** : Coste anual.

- **Gastos de consultoría** : En caso de requerirse algún consultor externo en cualquier etapa del proyecto.
- **Gastos de formación** : De todo tipo (Desarrolladores, Operadores, Implantadores, Usuario Final, etc).
- **Gastos de material** : Papel, tóner, etc
- **Costes derivados de la curva de aprendizaje** : De todo el personal involucrado: Desarrolladores, Técnicos de Sistemas, Operadores, y desde luego, Usuarios.
- **Costes financieros** , de publicidad, etc

En la estimación de beneficios se pueden considerar cuestiones como las siguientes:

- **Incremento de la productividad** : Ahorro o mejor utilización de recursos humanos.
- **Ahorro de gastos de mantenimiento** del sistema actual.
- **Ahorros de adquisición y mantenimiento de hardware y software** , o reutilización de plataformas sustituidas.
- **Incremento de ventas o resultados, disminución de costes** : Producidos por una mejora de la gestión (rotación de stock "just in time", analítica de clientes, etc).
- **Ahorro de material de todo tipo** : Sustituido por datos electrónicos que proporciona el sistema, como por ejemplo: papel correo, etc.
- **Beneficios financieros** .
- **Otros beneficios tangibles** : Ahorro de recursos externos, consultoría, formación, etc.
- **Beneficios intangibles** : Incremento de la calidad del producto o servicio, mejora de la imagen de la compañía, mejora en la atención al cliente, mejora en la explotación, etc.

Determinar la viabilidad del proyecto y su aceptación

Se basará en uno de los métodos siguientes:

Retorno de la inversión

Este método consisten en calcular el coste y el beneficio anual, conociendo el coste total al inicio del proyecto "C0", para determinar en qué año se recupera el coste total inicialmente estimado.

<u>AÑO</u>	<u>COSTE</u>	<u>BENEFICIO</u>	<u>BENEFICIO NETO</u>
0	C0	0	
1	C1	B1	B1 - C1
2	C2	B2	B2 - C2
...			
n	Cn	Bn	Bn - Cn

El año de recuperación de la inversión se produce cuando $\Sigma \text{Beneficio Neto} = C0$.

Valor Actual

Este método permite tener en cuenta que un gasto invertido durante un cierto tiempo produce un beneficio.

El método consiste en determinar el dinero que es viable invertir inicialmente para que se recupere la inversión en un periodo de tiempo definido previamente.

El resultado depende del tipo de interés (r) utilizado en la evaluación.

Se debe calcular, en primer lugar, el beneficio neto que se obtendrá cada año. Dicho beneficio no es real, ya que se debe estimar el valor real de dicha cantidad en el año n .

Para ello se aplica la fórmula:

$$\text{Valor Actual} = \text{Beneficio neto} / (1 + r/100)^n \quad n = \text{año} \quad 1, \dots, i$$

Se debe estudiar en cuántos años se recupera la inversión realizada inicialmente, o bien, si en un periodo de años fijado previamente se retorna la inversión y, por tanto, es viable el proyecto.

Si la inversión es el C_0 , se determinará la viabilidad del proyecto consultando la siguiente tabla:

<u>AÑO</u>	<u>COSTE</u>	<u>BENEFICIO</u>	<u>VALOR ACTUAL</u>
0	C_0		
1	C_1	B_1	$V.A_1 = (B_1 - C_1) / (1 + r/100)$
2	C_2	B_2	$V.A_2 = (B_2 - C_2) / (1 + r/100)$
...			
n	C_n	B_n	$V.A_n = (B_n - C_n) / (1 + r/100)$

El proyecto será viable si $\sum V.A_i > C_0$ a lo largo del periodo fijado.

Técnicas basadas en la teoría de la decisión multicriterio discreta

La Selección de un sistema informático, entre varias alternativas posibles, a fin de cubrir unas necesidades previas es un factor crítico de éxito de un sistema informático.

Esta tarea puede y suele ser bastante compleja en la realidad debido a múltiples razones. En primer lugar por las sutilezas técnicas de la materia, que impiden una fácil y nítida visión global del conjunto. En segundo lugar, por la dispersión y variedad de fuentes de los datos que han de constituir la información de base del problema. Por último por la dificultad de estructurar todo ello, junto con las frecuentemente diversas opiniones de expertos y directivos, de forma que pueda tomarse una decisión final.

Existen diferentes metodologías de análisis para afrontar el problema de manera más o menos cuantitativa, en un intento de hacerlo más racional y objetivo. Tradicionalmente se ha propuesto el análisis Coste-Beneficio, pero sus limitaciones son ya bien conocidas (necesidad de traducción a unidades monetarias, criterio único de evaluación), como para seguir utilizándolo.

Otra metodología muy utilizada últimamente es la denominada metodología multicriterio, que recogen la multiplicidad de aspectos y de puntos de vista que inciden en la evaluación de los sistemas que compiten por ser seleccionados. Este es el marco natural de la denominada Decisión Multicriterio Discreta (DMD).

Definición de criterios

El objetivo final de cualquier proceso de evaluación de bienes y/o servicios informáticos es la selección de la mejor alternativa posible escogida entre las existentes. Debe partirse de una enumeración y enunciado de las alternativas (a efectos operativos estas serán las ofertas presentadas por las empresas, los proyectos candidatos o las soluciones posibles). Las alternativas son completamente disjuntas y exhaustivas, es decir no caben en principio soluciones mixtas mezcla de otras alternativas (no obstante podemos considerar variantes dentro de la alternativa presentada por una empresa, que se introducirán en el proceso evaluativo como una oferta más). Llamaremos A_i a una alternativa genérica, con una variación entre 1 y m, (i = 1, M).

Por otra parte tenemos los criterios (también denominados atributos o características) que son los elementos en los que se basará el proceso de decisión, con su selección y posterior ponderación el decisor esta definiendo qué características de las alternativas le resultan importantes y en qué medida. Constituyen un conjunto discreto (C, 0=1, n). Los criterios deben ser ilustrativos de la característica que se quiera medir, cuando su número es muy grande hay que establecer un árbol de jerarquías entre ellos.

Por último tenemos las evaluaciones o puntuaciones X_{ij} de cada alternativa i respecto a cada criterio j, constituyendo en su conjunto la denominada matriz de decisión y que sirve para definir las alternativas en función de sus criterios. Por otro lado tenemos los pesos W, agrupados en el llamado vector de pesos (W₁ W_n) que representa la importancia que el decisor otorga a cada criterio.

Asignación de pesos

Desde el punto de vista de la DMD, estimar unos pesos W, que reflejen la importancia relativa de cada criterio j para el decisor, es una cuestión bastante delicada. La naturaleza de los pesos W_i como una cuantificación de la estructura de preferencias del decisor hace necesario "extraerlos" del mismo por algún procedimiento. Esto plantea, más en unos métodos que en otros, importantes problemas ya que, como es bien conocido, las inercias psicológicas del ser humano producen peligrosos sesgos e inconsistencias.

Los principales métodos son:

- Método de las utilidades relativas: Partiendo de unas estimaciones provisionales, ir afinando dichas estimaciones mediante comparaciones binarias de subgrupos de criterios.
- Método AHP (Analytic Hierarchy Process - Proceso Jerárquico Analítico): Comparaciones binarias de todos los criterios detallados.
- Método Delphi: Es el método del consenso. Consiste en consensuar entre todos los participantes decidores.
- Método de la entropía: Se utiliza cuando se quiere disminuir la subjetividad de los métodos anteriores. Determinar cual es la importancia relativa que tiene un determinado criterio. La importancia relativa se determina al estar directamente relacionada con la información intrínseca promedio generada por el conjunto de alternativas y por la asignación subjetiva que le otorgue el decisor. Información intrínseca promedio del criterio C_j es:

$$Ij = \frac{1}{\sum (Xij * \ln Xij)} \quad Xij \text{ son las puntuaciones normalizadas}$$

Puntuación de las ofertas

Una vez puntuadas los criterios de las diferentes alternativas, se hace preciso en muchos métodos (como el de ponderación lineal) el trasladar las puntuaciones brutas otorgadas a una escala normalizada por dos motivos fundamentales:

- Como estamos manejando un espacio multivariable hay que homogeneizar las puntuaciones para su comparación: esto es, considerarlas todas sobre la misma escala.
- Es razonable trabajar con escalas de dimensión suficientemente pequeña para simplificación de cálculos.

Con la normalización buscamos que las evaluaciones m de cada alternativa i correspondientes a un cierto criterio j sean comparables con las correspondientes a otros criterios. Llamaremos (X_1, \dots, X_m) al vector de puntuaciones de todas las alternativas sobre un criterio, el cual queremos transformar a uno normalizado (Y_1, Y_2, \dots, Y_m).

Los métodos de normalización más utilizados son los siguientes:

- Se otorga un cero a la mínima puntuación y un 1 a la máxima y el resto de las puntuaciones proporcionales a su valor en ese rango que es muy amplio.

$$X_{ni} = \frac{X_{sni} - \min X_{sni}}{\max X_{sni} - \min X_{sni}}, \quad \text{no se suele usar}$$

- La alternativa con valor máximo alcanza el 1 en esta escala, pero la mínima no alcanza el cero si ella misma no es cero. Este método es el más utilizado.

$$X_{ni} = \frac{X_{sni}}{\max X_{sni}}$$

- Este método mantiene la proporcionalidad pre y postnormalización:

$$X_{ni} = \frac{X_{sni}}{\sum X_{sni}}, \quad \text{usado para la entropía}$$

Selección de las alternativas

- Lexicográfico: Considerar el criterio de mayor peso y elegir aquella alternativa que para ese criterio tenga mayor puntuación. Si hay igualdad se toma el siguiente criterio en peso y así sucesivamente. Es un método sencillo, teniendo además la ventaja de no requerir comparabilidad intercriterios, un inconveniente es que no utiliza toda la información disponible.
- Promethee (pertenece al conjunto de métodos “relaciones de superación”): Ignora la cuantía de la diferencia sólo señala si existe o no, y al trabajar con los pesos de los criterios, considera si esa diferencia se ha hallado en un Criterio más o menos importante para el decisor.
- Concordancia: comparaciones binarias de las alternativas, como información del decisor exigen tan sólo un preorden en las evaluaciones por cada criterio, y unos

pesos en escala cardinal o incluso ordinal en algunas variantes. El procedimiento esencial de todos ellos gira alrededor de construir un coeficiente de concordancia cik para cada par de alternativas i,k. Dicho cik suele definirse como la suma de pesos de los criterios en que la alternativa i es superior a la k más la mitad de los pesos en los que ambas sean consideradas iguales.

- Permutación: La idea básica es la de comparar cada permutación posible de las alternativas, considerada como una ordenación de las mismas, con la información (ordinal) que aporta para cada criterio la matriz de decisión. Para cada permutación se calcula un llamado índice de evaluación, atendiendo a lo bien que concuerda con la información que proporcionan los datos, y aquella permutación que lo tenga máximo es la elegida. Entre sus ventajas figuran su flexibilidad cara al decisor (método cualitativo), y entre sus inconvenientes el que su dificultad de cálculo crece con m.
- Ponderación lineal (pertenece al conjunto de métodos “utilidad multiatributo”): consiste en calcular cual es el valor de cada alternativa y se elige la que tenga mayor valor. Para calcular el valor se emplea la fórmula: $V(A_i) = \sum X_{ij}W_j$. El problema fundamental es una buena estimación de los pesos. Necesita normalización previa de las puntuaciones. Entre sus ventajas podemos citar las siguientes: Procesa bien los fenómenos económicos, ya que suelen ser lineales, es un método muy intuitivo (el decisor lo comprende bien, ha demostrado su utilidad en otros contextos de decisión financieros, comerciales) y es el primer método para implantar en organizaciones poco tecnificadas. En cuanto a sus inconvenientes deben citarse: El ser de relativa facilidad en su manipulación vía pesos o vía evaluaciones, tener un enfoque absolutamente compensatorio lo que tiende a favorecer a las alternativas que son medianías y los resultados no son significativos sin una cuidadosa elección de escalas de medida de las evaluaciones.

Documática. Gestión y archivo electrónico de documentos. Sistemas de gestión documental y de contenidos. Sindicación de contenido. Sistemas de gestión de flujos de trabajos. Búsqueda de información: robots, spiders, otros. Posicionamiento y buscadores (SEO)

Introducción

En una organización, la información susceptible de almacenamiento crece a un ritmo exponencial. Dicho crecimiento hace necesario solucionar el problema de su adecuada gestión, ya que a partir de un cierto volumen se hace imprescindible un sistema organizativo que posibilite la localización de la información que se precise en cualquier momento.

Podemos clasificar la información que es necesario manejar de la siguiente manera:

- Información estructurada: se trata de información que se puede subdividir en campos. Nos estamos refiriendo por ejemplo a los registros de las tablas de las BDR.
- Información no estructurada: es información en la que no se puede encontrar una estructura interna. Hablamos por ejemplo de fotos, archivos de texto, archivos de vídeo, páginas web, etc. Incluimos en este apartado los documentos de cualquier tipo.

El ámbito de este tema se circunscribe al segundo tipo de información.

El desarrollo de los sistemas automatizados de recuperación de información se inició con el objetivo de facilitar el manejo de la enorme cantidad de literatura científica surgida desde los años 40; posteriormente esta disciplina se extendió a otros ámbitos fuera de los científicos.

Otlet es considerado el precursor de la gestión de documentación automática (documática) con su obra *Traité de Documentation*, publicada en 1934, en la que expone los principios y relaciones de la Tecnología documental. Otlet identifica los componentes fundamentales del moderno concepto de Documentación Automática (o Automatizada), distinguiendo estas tres premisas principales:

- Establece una teoría sobre la organización, las herramientas y los soportes tecnológicos para sustentar esta nueva disciplina.
- Aplicación práctica del proceso documental: la Documentación ocupa un lugar preponderante en la organización.
- Objetivo: satisfacer las necesidades informativas del usuario.

Posteriormente en los años 50, los especialistas se centran en el problema de la búsqueda y recuperación de información, acuñándose el término *Information Retrieval* (recuperación de información). La recuperación de información es el conjunto de tareas mediante las cuales el usuario localiza y accede a los recursos de información que son pertinentes para la resolución del problema planteado. En un sistema documático, el proceso de recuperación de la información sigue en general el esquema siguiente:

- El usuario formula una necesidad de conocimiento.
- Se interroga al sistema gestor documental (SGD).
- El SGD devuelve una lista de referencias.
- Si lo que buscamos no está en la lista se realiza una segunda búsqueda y empieza el proceso de nuevo.

A finales de los años 60 se da un nuevo paso en la evolución de la documática, con la introducción de la *Information Science* (Ciencia de la Información) como ciencia integradora de la teoría, proceso y práctica documental con otras ciencias complementarias, como la cibernetica, la informática, la teoría de la información y la comunicación, etc.

El desarrollo de nuevas teorías ha traído, de la mano de la Ciencia de la Información, la aparición de la disciplina *Information Management* (Gestión de la Información y la Documentación en las Organizaciones), en la que desempeñan un papel fundamental las telecomunicaciones y la informática, íntimamente relacionadas con los sistemas de información, en el marco de redes complejas de información.

Archivo Electrónico de Documentos

Como ya hemos visto, el archivo electrónico de documentos o documentación automática consiste en la gestión de grandes volúmenes de información no estructurada (texto, imágenes, gráficos, sonidos, etc).

Adicionalmente, será necesario gestionar cierta información que permita localizar el documento cuando sea necesario; así, los documentos han de ser sometidos a un proceso de **indización**.

El otro gran proceso involucrado en un sistema de gestión documental es la **recuperación de la información**. Abarca el conjunto de tareas mediante las que un usuario recupera la información relevante en respuesta de una necesidad cognitiva.

Indización

Consiste en extraer los conceptos clave del texto de un documento. Su objetivo es definir el contenido de un documento mediante un conjunto de conceptos que especifican el tema o temas de que trata.

La indización conlleva dos procesos fundamentales:

1. Extraer los conceptos informativos de cada documento.
2. Traducirlos a un lenguaje documental.

El lenguaje documental es el que se usa para la interrogación del SGD. En función del lenguaje documental que utilice, podemos clasificar los SGD en dos grandes grupos:

- Sistemas de lenguaje libre o free-text. Permiten hacer búsquedas en lenguaje natural. Un ejemplo es el buscador de Internet Google.
- Sistemas basados en lenguajes controlados. En este caso, los términos que contiene un lenguaje documental son de dos clases:
 - Términos preferentes o descriptores (descriptors, keywords): son aquellos que deben utilizarse en la indización y en la recuperación. Representan términos precisos y únicos.
 - Términos no preferentes (no-descriptors): no pueden asignarse a los documentos ni la indización, ni realizar consultas utilizando estos.

En cuanto a la indización, hay que tener en cuenta que la cantidad de términos que representen a un documento no indica la calidad de la indización; no por muchos términos es más precisa, cuántos más términos representan a un documento aumenta la **exhaustividad** (mayor probabilidad de que se seleccione ese documento) y disminuye la **precisión** (conceptos que realmente identifican al documento).

Si se cae en excesiva exhaustividad o precisión, se pueden producir dos fallos a la hora de realizar una búsqueda documental:

- **Ruido** : documentos que el sistema ha seleccionado y que en realidad no responden a la pregunta. Esto es consecuencia de indicar los documentos con más términos de los que se debiera.
- **Silencio** : documentos que al hacer la búsqueda no han sido seleccionados y sin embargo responden a la pregunta formulada. Es consecuencia de la falta de precisión, es decir, no indizar los términos correctos.

Etapas de la indización

Hablamos de sistema indizador como el encargado de realizar el proceso de indización. Existen aplicaciones en que este proceso es manual, realizado por un operador, pero en otras el operador es ayudado por un sistema informático, por ser un proceso totalmente automático.

Las distintas fases de las que consta el proceso de indización son las siguientes:

1. Examen del documento . El examen será más o menos extenso según el tipo de documento y su forma física; en general, el sistema indizador tendrá que asegurarse de leer toda la información y no olvidar ninguna parte. En el caso de un documento de texto, éstas son las partes del texto que habrá de tener en cuenta por orden de importancia:

- título
- resumen
- introducción, capítulos y conclusiones
- ilustraciones y gráficos

- palabras subrayadas o impresas en otra tipografía

2. Identificación del documento . El sistema indizador aplicará una serie de criterios para identificar los conceptos esenciales para la descripción del tema, eligiendo los más acordes con las necesidades del centro o servicio en que se esté indizando.

En la selección de los conceptos se persiguen dos objetivos principales:

- Exhaustividad: no dejar de indizar nada que pueda ser importante.
- Pertinencia: la información ha de ser representativa del documento.

Para la identificación de los conceptos esenciales se pueden emplear los siguientes métodos:

- Sistema full-text: consiste en extraer todas las palabras clave, a excepción de aquellas que se encuentren en una lista de palabras vacías (aquellas que no aportan información, como los determinantes, preposiciones, etc). Es el sistema que se utiliza habitualmente para los sistemas documentales free-text.
- Indización mediante lenguajes controlados: el universo de las palabras a indizar está restringido, utilizándose una lista de descriptores.
- El método estadístico: seleccionar los conceptos más significativos mediante el análisis de las frecuencias de los términos del documento.
- El método sintáctico: utiliza técnicas de análisis morfológico y semántico para captar la estructura del texto. Utilizado sobre todo en la investigación sobre el procesamiento de lenguaje natural.

3. Traducción de los términos . Consiste en la traducción de los conceptos extraídos del documento al lenguaje documental utilizado, es decir, a términos de indización:

- Si utilizamos un lenguaje documental controlado, habrán de traducirse a los convenientes descriptores.
- Si utilizamos texto libre, habrá que comprobar que los conceptos extraídos están aceptados en las distintas fuentes de referencia:
 - diccionarios y encyclopedias
 - libros de texto y manuales
 - tesauros
 - etc

Los tesauros

Los tesauros que se acaban de citar son diccionarios que muestran la equivalencia entre los términos o expresiones del lenguaje natural y los términos normalizados del lenguaje documental, así como las relaciones semánticas que existen entre ellos.

Los tesauros en España están definidos en la norma UNE 50-106-90, la cual no es de obligado cumplimiento, pero proporciona un marco para la comunicación entre centros y para facilitar el trabajo en equipo.

Los elementos principales de un tesauro son los siguientes:

- Unidades lexicales. A su vez se subdividen en varios tipos:
 - descriptores
 - términos equivalentes o sinónimos. Son aquellos cuya presencia es útil en el tesauro, pero que no se pueden utilizar en la indización, pues remiten o envían a un descriptor. Pueden ser de dos clases:
 - sinónimos lingüísticos: se traducen directamente por un descriptor y tienen exactamente el mismo significado que el descriptor elegido.

- sinónimos documentales o quasi-sinónimos: agrupan en un solo descriptor varios términos que tienen un significado próximo, aunque no es exactamente el mismo.
- infraconceptos: términos que no tienen sentido por sí solos y que se añaden a los descriptores para formar nuevos descriptores. Ejemplo: infra, multi, super, etc.
- palabras herramienta o instrumento: descriptores que no tienen significado exacto si van solos. Son términos como: comparación, evaluación, método.
- Relaciones entre unidades lexicales. Existen las siguientes clases de relaciones:
 - Relaciones de equivalencia o sustitución: son aquellas que relacionan un sinónimo con un descriptor.
 - Relaciones de jerarquía: expresan relaciones de superioridad y subordinación entre descriptores. A su vez pueden ser:
 - relaciones genéricas: en las que existe un término genérico que representa un concepto en el que están contenidos los términos específicos.
 - relaciones partitivas o relaciones todo-parte: en las que se expresa que un término se compone de otros.
 - Relaciones asociativas o de vecindad: indican las analogías que pueden existir entre dos descriptores.
 - Relaciones de definición: que relacionan un descriptor con su uso o aplicación.

Los tesauros se utilizan para eliminar ambigüedades y facilitar la indización, pero también son utilizados en el proceso de recuperación de la información que se verá posteriormente.

Sistemas de Indización

En función de cuál es el resultado de la indización, es decir, cómo se organiza la información resultado de la indización de los documentos, podemos establecer las siguientes categorías:

- **Ficheros planos** : (a) la información referente a la indización de uno o más documentos son almacenados en un fichero (generalmente en formato de texto ASCII). La búsqueda sobre estos ficheros planos se llevan a cabo generalmente por medio de la localización de patrones de texto.
- **Ficheros inversos** : (b) son un tipo de fichero índice donde la estructura de cada ítem (entrada) del fichero es, generalmente: descriptor, identificador de documento, identificador de campo, donde el identificador de documento es único para cada documento y el identificador de campo es un término que nos indica dentro de qué campo del documento aparece el descriptor. Algunos sistemas incluyen también información acerca de la localización en el documento del párrafo y frase de los términos utilizados para proceder a interrogar la BD. La búsqueda se realiza, corrientemente, por medio de la localización de los términos solicitados en el fichero inverso.

Document	Text
1	Pease porridge hot, pease porridge cold,
2	Pease porridge in the pot,
3	Nine days old.
4	Some like it hot, some like it cold,
5	Some like it in the pot,
6	Nine days old.

(a) Example text; each line is one document

Number	Term	Text
1	cold	1,4
2	days	3,6
3	hot	1,4
4	in	2,5
5	it	4,5
6	like	4,5
7	nine	3,6
8	old	3,6
9	pease	1,2
10	porridge	1,2
11	pot	2,5
12	some	4,5
13	the	2,5

(b) Inverted file for text of (a)

- Los ficheros de patrones de bits contienen hileras de dígitos binarios, patrones de bits que representan a los documentos. Existen varias formas de construir estos patrones de bits. Un método común consiste en la división de los documentos en bloques lógicos, e identificar los términos de indexación que contiene cada bloque. Cada palabra es desglosada para traducirse en una hilera de bits (es decir, un patrón de bits con algunos de los bits "puesto a 1"). Los patrones de bits de cada palabra en un bloque son agrupados para crear un bloque de patrones. Los bloques de signaturas se concatenen posteriormente para producir el patrón de bits del documento. La búsqueda se lleva a cabo por medio de la comparación entre los patrones de bits de las interrogaciones con los patrones de bits de los documentos de la BD.
- Los grafos (redes) son colecciones ordenadas de nodos conectados por arcos y se usan para representar documentos de diversas formas y maneras. Un ejemplo es el grafo denominado **red semántica**, que representa las relaciones semánticas que se establecen en el texto, relaciones que se pierden a menudo en otros sistemas de indexación. Aunque constituyen un campo interesante para el estudio, resultan bastante difíciles de llevar a la práctica y requieren excesivo esfuerzo manual para el proceso de la representación de las colecciones de documentos.

Recuperación de la Información

La recuperación de la información es el conjunto de tareas mediante las cuales un usuario recupera la información **relevante**, para dar respuesta a su necesidad cognitiva. Es decir, un documento será relevante, si satisface la necesidad de conocimiento del usuario. Esto supone una gran diferencia con los sistemas gestores de BD, en los que el criterio de éxito de una interrogación a la BD es la exactitud y corrección de los datos, en ningún caso depende de las subjetividad del usuario.

Uno de los problemas con los que nos encontramos, al interrogar un SGD, es que el usuario concibe su necesidad de conocimiento en "lenguaje natural", el cual ha de ser traducido al lenguaje documental que entiende el sistema. Por lo tanto, puede producirse una pérdida de eficiencia en la traducción. Por ello se dice que el tipo de recuperación que se puede producir en la interrogación a un SGD es **aproximada o probabilística**, es decir, ante una misma necesidad de conocimiento se pueden obtener múltiples respuestas dependiendo de la habilidad ante una misma necesidad de conocimiento se pueden obtener múltiples respuestas dependiendo de la habilidad del usuario para traducirla al lenguaje documental que entiende el sistema. Hay que hacer notar que esto supone otra

diferencia relevante con los SGBD tradicionales, en los que la información que devuelve el sistema es **determinista**, ya que ante una misma necesidad de información siempre devolverá el mismo resultado.

Métricas de Eficiencia

Al igual que ocurría en el proceso de indexación, a la hora de la recuperación de la información no se puede ser exhaustivo y preciso al mismo tiempo, ya que si uno de los parámetros aumenta el otro disminuye, como podemos representar gráficamente de la siguiente manera:



Por ello, para medir la eficiencia de un sistema de recuperación de la información se establecen una serie de parámetros, que enunciaremos a continuación basándonos en la tabla siguiente:

	Relevantes	No Relevantes
Extraídos	A	B
No extraídos	C	D

La tabla pretende reflejar, para una consulta a un SGD:

- A: documentos relevantes que han sido devueltos por el SGD.
- B: documentos no relevantes que han sido devueltos por el SGD, lo que hemos definido anteriormente como ruido.
- C: documentos relevantes que no han sido devueltos y que deberían haber sido extraídos, lo que hemos llamado silencio.
- D: documentos no relevantes y que no han sido extraídos.

Definimos entonces las siguientes métricas:

- Índice de pertinencia o precisión: mide cuantos documentos devueltos son los considerados relevantes por el usuario: $A / (A + B)$. Es en definitiva una medida de la calidad de la información obtenida.
- Índice de exhaustividad o de respuesta: mide el porcentaje de documentos que han sido devueltos sobre el total de la base documental: $A / (A + C)$. Es una medida de la cantidad de la información obtenida.
- Tasa de ruido: mide el porcentaje de documentos que carecen de interés y han sido devueltos por el sistema: $B / (A + B)$.

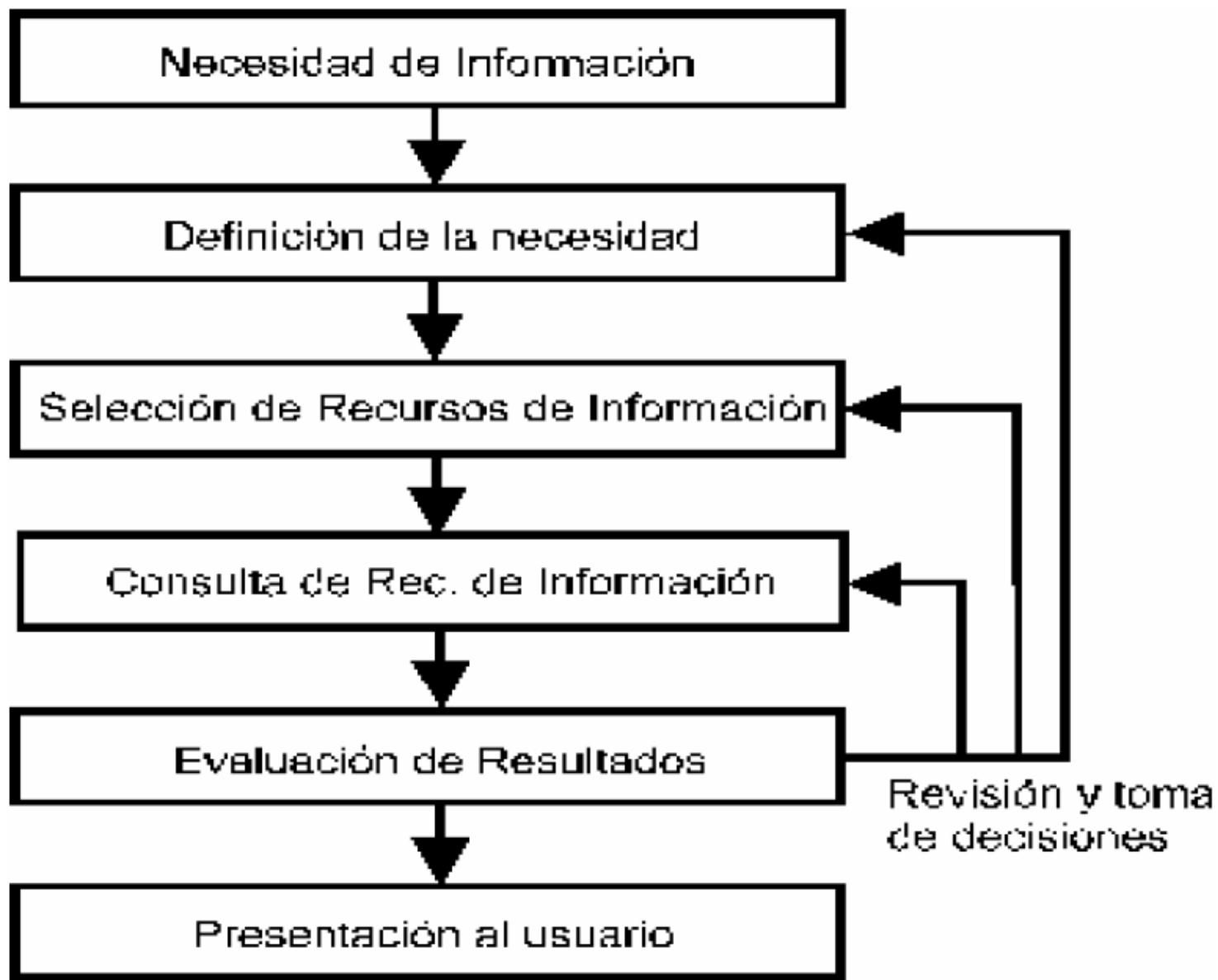
El Proceso de Recuperación de la Información

Un proceso de recuperación, al que podríamos considerar "genérico", seguiría las siguientes fases:

1. Definición de las necesidades informativas del usuario.
2. Selección y ordenación de las fuentes a utilizar.

3. Traslación de las necesidades del usuario al lenguaje documental propio de la fuente a utilizar en cada caso. Es posible, además, encontrar fuentes en las que no se utilice ningún tipo de vocabulario controlado, en cuyo caso resultará necesario afinar el trabajo terminológico.
4. Traducción de la expresión de lenguaje documental al lenguaje de interrogación propio de cada sistema.
5. Ejecución de las expresiones del lenguaje de interrogación obtenidas.
6. Consulta de las respuesta obtenidas, para analizar su pertinencia o no a la cuestión planteada.
7. Replanteamiento, si procede, de las expresiones utilizadas, si los resultados obtenidos no son pertinentes.
8. Selección y obtención de los documentos que respondan a las necesidades manifestadas por el usuario.
9. Transmisión del resultado, preparado adecuadamente, al usuario.

Este proceso se puede plasmar gráficamente como aparece en la figura:



Organización Funcional de los Sistemas Documáticos

En los Sistemas de Gestión Documental (SGD) se pueden identificar una serie de subsistemas funcionales. Un SGD puede incorporar todos ellos o sólo algunos. Además, hay SGD's que permiten integrar subsistemas de otros fabricantes:

- **Sistemas de Gestión de Bases de Datos Documentales (SGBDD)** : son sistemas que incorporan todas las características de los SGBD tradicionales, incluyendo la

creación y mantenimiento de BD Documentales (adecuadas para información no estructurada), usuarios, controles de seguridad, e incluso lenguajes propios de programación. Estos sistemas están basados en sistemas de archivo y ficheros inversos, los cuales son una modalidad de organización de los datos especialmente apropiada para la información documental. Los rasgos más característicos de un SGBDD son:

- capacidad para almacenar información textual de longitud grande y variable.
- capacidad para recuperar con rapidez registros que responden a un criterio de búsqueda.
- capacidad para realizar búsquedas multicriterio sobre ficheros inversos utilizando lógica booleana.
- capacidad para administrar tesauros y diccionarios terminológicos.

Como ejemplos de sistemas de gestión de BD más representativos, podemos citar: BRS/ Search de BRS Information Technologies (uno de los más completos), Inmagic, CDS-Isis y su interfaz Winslsis, ...

- **Sistemas de indización** : anteriormente hemos visto el proceso de indización documental. Estos sistemas por lo tanto son aquellos encargados de realizar dicho proceso.
- **Sistemas de exploración o escáneres** : se trata de aplicaciones que son capaces de acceder a ficheros con diferentes formatos y buscar dentro de los mismos las cadenas de caracteres que respondan a lo expresado en la ecuación de búsqueda. Pueden encontrarse aplicaciones que combinen la exploración con la indexación, como dtSearch.
- **Sistemas de gestión bibliográfica** : sistema especializado para la gestión y mantenimiento de bibliografías especializadas. Es una aplicación específica de los sistemas de gestión de bases documentales que permite, no sólo el almacenamiento y la recuperación de referencias bibliográficas, sino también la exportación de estas referencias en diferentes formatos de cita bibliográfica a diferentes procesadores de textos, sistemas de gestión de BD, etc.
- **Sistemas de recuperación de información (SRI)** : son aplicaciones que se encargan exclusivamente de recuperar información de BD documentales no modificables. Ponen a disposición del usuario potentes herramientas de búsqueda y de apoyo a la búsqueda, pero su funcionalidad queda reducida a la consulta y exportación de documentos.

Los SRI incorporan un **gestor de interrogación o motor de búsqueda**, el cual realiza búsquedas dentro de una BD de documentos. El motor de búsqueda recibe la interrogación del usuario (query), que consiste en una o varias palabras, realiza la búsqueda en la BD y extrae una lista ordenada de documentos que cumplen entera o parcialmente con la interrogación. El orden depende de una puntuación (score) que asocia el programa a cada documento cuando realiza la búsqueda y en cada caso varía. Un criterio para puntuar los resultados que usualmente se aplica es que cuanto más próximos en el documento aparecen los términos de búsqueda, mayor es la puntuación del documento.

Un SRI debe permitir la recuperación de la información contenida en los documentos de la BD a la que accede, a través de cualquier término existente en ella, mediante la formulación de ecuaciones de búsqueda que permitan combinar los términos según diferentes criterios. Existen sistemas que ofrecen la posibilidad de ejecutar las consultas sobre una o varias BD simultáneamente. Los documentos resultantes se agrupan en sets o conjuntos, susceptibles de combinación posterior.

El SRI ha de poseer algún tipo de mecanismo para la salida de la información, generalmente mediante edición en pantalla, impresión y redirección a ficheros de los documentos de interés para el usuario. Las órdenes de salida de información deben ofrecer la posibilidad de enviar ésta a diferentes destinos, así como los formatos de

presentación de los datos a utilizar (tamaño, campos, ...). Deben incluirse aquí las capacidades para ordenar, según diferentes criterios, los documentos resultantes. Otra función a considerar es la posibilidad de crear nuevas BD, tomando como base los documentos recuperados en un búsqueda previa.

Es interesante que el SRI incluya también herramientas que permitan analizar y procesar la respuesta obtenida, utilizando herramientas de análisis de frecuencias de los términos (es decir, cuántas veces aparece el término buscado en los documentos recuperados) o de coocurrencias (frecuencia con la que aparecen dos o más términos de búsqueda en los documentos recuperados).

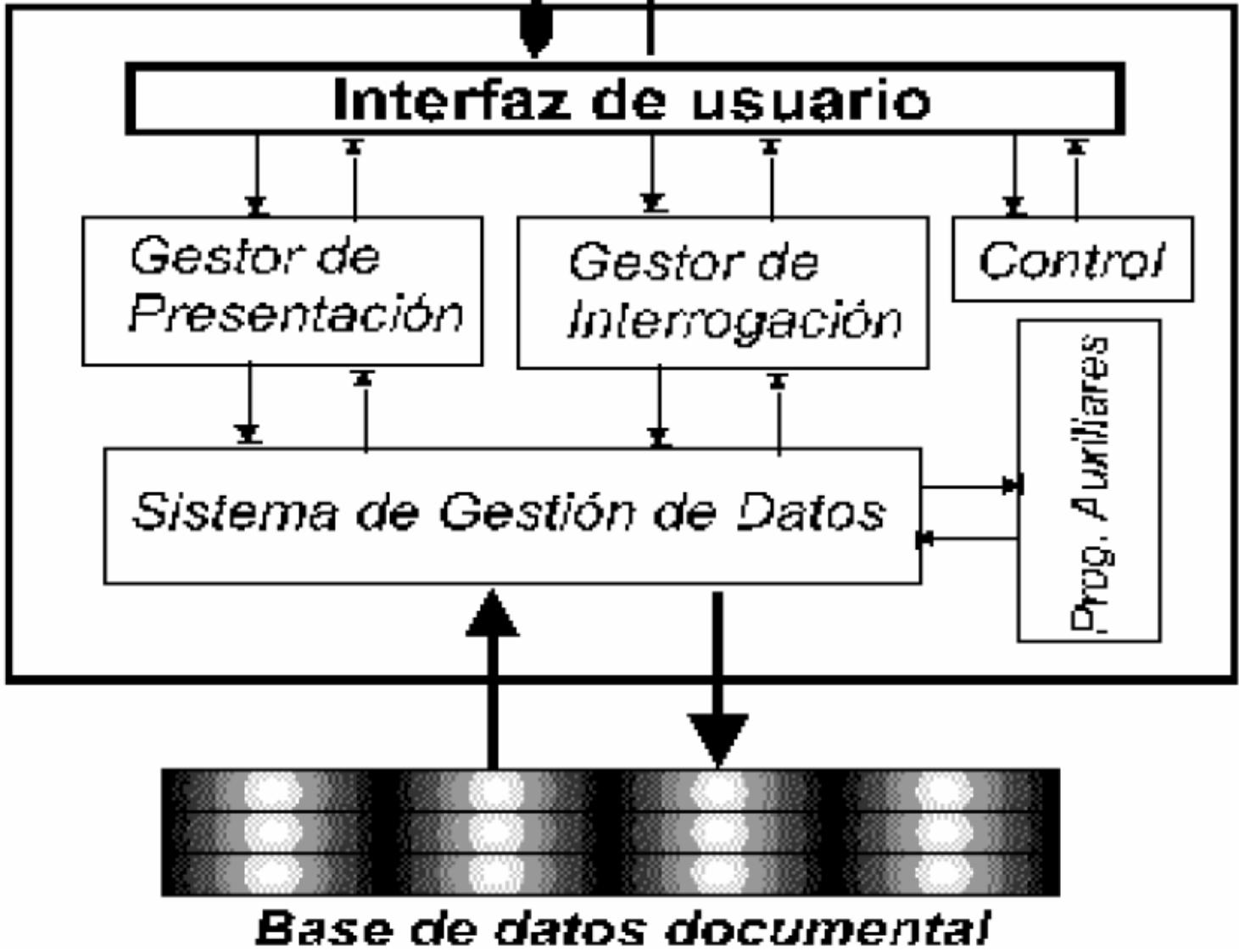
Otro posible subsistema de un SRI es aquel que permita definir los perfiles de búsqueda de los usuarios, así como realizar un seguimiento de las ecuaciones que ejecuten. Por ejemplo, la posibilidad de almacenar las ecuaciones de búsqueda que usualmente ejecutan, de manera que puedan ejecutarse en cualquier momento, se les llama normalmente "macros". Estas macros son ficheros susceptibles de edición y modificación, lo que facilita la recuperación de información con un mínimo esfuerzo de tiempo y coste.

Un elemento fundamental de un SRI es que incluya algún mecanismo de control terminológico, tanto para la entrada de datos como para su recuperación. Puede tratarse de un tesoro, de un glosario o de un diccionario terminológico.

Además se puede incluir una ayuda al usuario en todo momento, a través de mensajes y líneas de estado, especialmente durante el proceso de interrogación (interrogación asistida). En sistemas de recuperación en línea (teledocumentación), el sistema informa al usuario del tiempo de conexión, tareas ejecutadas, coste de la sesión, etc. Los mecanismos de ayuda al usuario, especialmente aquellos referidos a la evaluación y refinamiento de las búsquedas, son una de las principales áreas de investigación.

Por último, dependiendo de la configuración del sistema, éste puede ofrecer opciones de acceso multiusuario, niveles de seguridad, reorganización y recuperación de ficheros, etc.

Requerimientos del Usuario



- Sistemas hipertextuales: en su origen, los hipertextos e hipermedias eran una forma de organizar, acceder y explorar documentos de diferentes tipos, que posteriormente se han popularizado como motor y parte de tutoriales y presentaciones. Actualmente estos sistemas están volviendo a ser considerados como una forma válida y muy avanzada de gestionar documentación. Para que sea posible una existencia real de los conceptos de hipertexto e hipermedia, deben utilizarse aplicaciones que sean capaces de crear los vínculos y asociaciones entre los documentos. Las aplicaciones ofrecen unos elementos particulares que facilitan la creación y navegación por las estructuras hipertextuales:
 - Un conjunto de ficheros que contienen los documentos relacionados.
 - Ventanas de presentación de los documentos, las cuales son modificables en tamaño y posición.
 - Punteros o enlaces, que generalmente utilizan una representación gráfica distinta a la del resto del material informativo, en forma de color, iconos, botones... Así como dispositivos señaladores, que facilitan la selección y el acceso a los documentos mostrados en las ventanas.

- Herramientas de creación de enlaces y anotación de la navegación, lo que da al usuario la posibilidad de crear sus propias asociaciones y documentos.

Estas funcionalidades se integran en una herramienta que en el entorno hipertextual es conocida como “browser”, navegador o visualizador. El visualizador actúa como una interfaz, que muestra al usuario el contenido informativo de los documentos que selecciona, mediante la selección de enlaces. Suele completarse con la posibilidad de ejecutar búsquedas en el texto completo que contienen los documentos y/o búsquedas más rígidas utilizando lenguajes clásicos de interrogación. La interrogación, sea de texto, imágenes o sonidos, suele realizarse a través de la ejecución de patrones, que representan una necesidad dada de información por parte del usuario. Además, una completa aplicación para este ámbito debería ser capaz de generar mapas gráficos de la estructura hipertextual y utilizar estas representaciones para acceder directamente a los documentos deseados.

La visión que obtiene el usuario mediante el visualizador es una visión transparente, integrada, en la que no resulta complicado navegar de un documento a otro. Esta aparente facilidad no debe ocultar que los documentos pueden encontrarse en diferentes ficheros informáticos, e incluso en diferentes ordenadores, formando lo que se llama repositorio de información, que será tratado con más detalle en el próximo capítulo, por su relación con las BD multimedia.

Los sistemas y estructuras de hipermedia pueden además incorporar inteligencia embebida, es decir, ser capaces de ejecutar otras aplicaciones o de tomar decisiones con la actividad desarrollada por el usuario, tanto en la utilización de los enlaces como en el acceso a los contenedores.

- **Sistemas de Gestión Documental o de Gestión Electrónica de Documentos (GED)** : se trata de sistemas que pretenden ofrecer una solución integral para la documentación, especialmente administrativa y de gestión, que se utiliza en una organización dada (PRAX, 1994; LASSOURY, 1994). Incorporan funciones clásicas de gestión de BD y utilizan esquemas de obtención de una copia del documento original mediante escáner, almacenamiento óptico o magneto-óptico y un nivel básico de descripción textual del documento y de su contenido.
- **Sistemas o Gestores de Información Personal (Personal Information Systems/ Managers)** : son aquellos que integran, en un único entorno, todos los documentos, ficheros y relaciones entre ellos que son de interés para el trabajo de un usuario. Numerosos sistemas integrados de informatización ofrecen a sus usuarios un acceso homogéneo a los diferentes tipos de documentos y ficheros que manejan en su trabajo diario.
- **Sistemas compuestos** : se denomina así a aquellos que dan soporte a todas las tareas que se realizan en una unidad informativa, sea ésta un archivo, biblioteca o centro de documentación. Esto significa que cubren tanto la cadena documental como la gestión administrativa. Sirvan como ejemplo las aplicaciones de automatización de bibliotecas, como Absys o Libertas, o las aplicaciones de automatización de archivos, como la desarrollada para el Archivo de Indias de Sevilla. Normalmente, integran un motor documental, encargado de gestionar las BD documentales que cubren los catálogos, y un motor relacional, que cubre las tareas administrativas.

Optimización de Consultas y Recuperación de la Información

Lenguajes de Interrogación y Operadores

Un lenguaje de interrogación puede definirse como un conjunto de órdenes, operadores y estructuras que, organizados conforme a unas normas lógicas, permiten la consulta de fuentes y recursos de información electrónica.

El resultado de la combinación de estos elementos, siguiendo las normas establecidas, es una expresión a la que se conoce con el nombre “ecuación”, capaz de interrogar el contenido de la fuente de información. La definición mínima de un lenguaje de interrogación y de sus componentes puede encontrarse en el borrador del la norma ISO 8777-1988.

Las normas lógicas que rigen un lenguaje de interrogación responden a cuestiones relacionadas con la coordinación de los elementos, es decir, con la formulación de ecuaciones. Estas normas funcionan como la sintaxis del lenguaje, es decir, especificarán el orden de los elementos, la disposición de las estructuras, sus posibilidades combinatorias, las prioridades en la ejecución y todo tipo de posibles funciones. Las órdenes serán aquellas palabras o abreviaturas que le indicarán al sistema las acciones a ejecutar (buscar la expresión, mostrar los documentos o registros resultantes, consultar el tesoro o los ficheros inversos, ejecutar un perfil de usuario, ...). Sin embargo, no todos los lenguajes de interrogación utilizan las mismas palabras como órdenes, aunque las órdenes ejecuten las mismas funciones. Existen intentos para homogeneizar la interrogación de las BD, como el lenguaje CCL (Common Command Language) promovido por la Unión Europea, que aún no han alcanzado el objetivo para el que fueron desarrollados. A este panorama se une la proliferación de interfaces gráficos de usuario, que sustituyen a las órdenes y las sintaxis tradicional, dejando al usuario (si éste lo desea) sólo la labor de introducir los términos y los operadores que expresan las relaciones existentes entre ellos.

En un lenguaje de interrogación, los operadores son los encargados de expresar las relaciones que mantienen entre sí los términos que definen (más adecuado sería decir que pueden definir) las necesidades informativas del usuario.

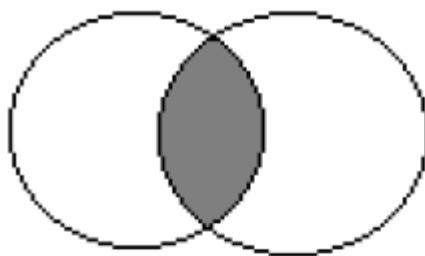
Pueden distinguirse diferentes tipos de operadores que se analizan a continuación.

Operadores Lógicos o Booleanos

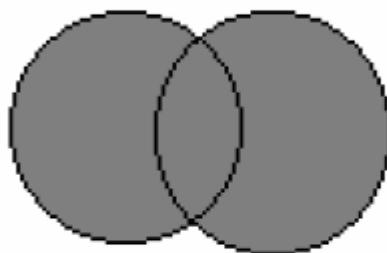
Los operadores lógicos, también llamados booleanos en honor a George Boole, precursor de la lógica simbólica y del álgebra de conjuntos, son los más utilizados en numerosos sistemas. El principio que rige la utilización de este tipo de operadores es que las relaciones entre conceptos pueden expresarse como relaciones entre conjuntos. Las ecuaciones de búsqueda pueden transformarse en ecuaciones matemáticas, que ejecutan operaciones sobre los conjuntos, lo que da como resultado otro conjunto. Los tres operadores básicos son el operador suma/unión (generalmente identificado como O/OR), el operador producto/intersección (identificado como Y/AND) y el operador resta/negación (identificado como NO/NOT). A su vez, estos operadores pueden combinarse entre sí generando operaciones más complejas, como el O exclusivo (elimina la intersección), etc.

No deben obviarse los problemas que plantean los operadores booleanos, independientemente de su potencia. En primer lugar, siempre se plantean en términos absolutos (es decir, selecciona el documento en función de si las palabras de búsqueda están o no están presentes, sin considerar el peso específico de cada término en el contexto). Por esa misma razón, es necesario un alto valor de precisión en los términos de

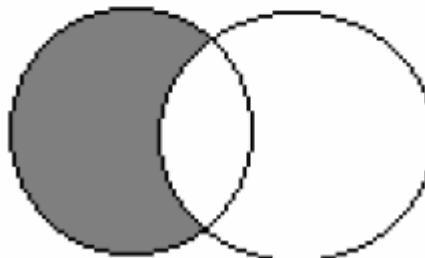
búsqueda utilizados. En segundo lugar, requieren claridad en la composición de las expresiones a buscar.



AND/Y
Producto lógico



OR/O
Suma Lógica



NOT/NO
Resta lógica

Los tres operadores booleanos básicos.

Operadores posicionales

La utilización de operadores posicionales pretende superar algunas de las limitaciones anteriormente citadas que ofrecen los operadores booleanos. Toman como punto de partida la consideración del valor de cada término dentro del contexto, es decir, de su relación con el resto. En definitiva lo que quiere decir es que la posición de los términos de búsqueda dentro del documento es significativa para valorar su utilidad. Los operadores posicionales pueden dividirse en dos tipos:

- Posicionales absolutos: Son aquellos que permiten buscar un término en un lugar dado del documento o registro. Por regla general, son operadores de campo, es decir, permiten al usuario fijar en qué campo o campos presentes en la estructura de BD debe aparecer el término buscado. La presencia del término en un campo dado (por ejemplo, en el campo título) puede ser una garantía de la adecuación del documento a los objetivos, en la mayor parte de las situaciones.
- Posicionales relativos: También llamados de proximidad, se trata de operadores que permiten establecer la posición de un término respecto a otro dado. Se considera que la cercanía entre los dos términos puede reflejar una íntima relación entre los conceptos reflejados por los mismos. Estos operadores permiten definir el nivel de proximidad entre los términos (mismo campo, línea, frase, número de términos significativos que los separa ...).

Operadores de Comparación

Especifican el rango de búsqueda, fijando unos límites para la misma. Estos límites pueden ser tanto numéricos como alfabéticos, correspondiendo los operadores a formas

del tipo “mayor que”, “menor o igual que”. Se utilizan principalmente en documentos que pueden contener datos numéricos.

Operadores de Truncamiento

Pueden darse situaciones en las cuales sea necesario utilizar no un término simple, sino también sus derivados, determinados por prefijación o sufijación, mínimas variantes léxicas, etc. Para facilitar este tipo de búsqueda se han introducido operadores de truncamiento, a los que también se llama máscaras. Se trata de operadores (normalmente se emplean símbolos como *, \$) cuya presencia puede sustituir a un carácter o a un conjunto de caracteres, situados a la izquierda, dentro o a la derecha del término en cuestión.

En los actuales sistemas de recuperación de información es posible encontrar todos estos tipos de operadores, que pueden combinarse entre sí, permitiendo crear ecuaciones complejas que reflejan con bastante precisión los conceptos y sus relaciones. La combinación de los operadores debe respetar un conjunto de reglas básicas en todos los sistemas, que establecen las prioridades y formas de ejecución de ecuaciones complejas, cuando éstas combinan más de dos conceptos. En primer lugar, los sistemas tienden a resolver, o ejecutar en primer lugar, aquellas expresiones que se relacionan utilizando el operador más restrictivo o prioritario. Por ejemplo, un operador posicional absoluto posee un nivel de restricción (una prioridad) mayor que un operador booleano, lo que significa que el sistema ejecutará antes la expresión cuyo operador es el posicional absoluto, combinando posteriormente el resultado con el operador booleano y su término relacionado.

Sin embargo, pueden darse expresiones en las cuales sea necesario variar estas prioridades y ordenar al sistema que ejecute en primer lugar expresiones con operadores de menor nivel de restricción, relacionando luego su resultado con términos a través de operadores más restrictivos. Para estas situaciones, se utilizan paréntesis, los cuales engloban a las expresiones que deben ejecutarse en primer lugar, independientemente de las prioridades fijadas por el sistema. La utilización de expresiones entre paréntesis hace posible, por ejemplo, que el resultado de una expresión con un operador booleano pueda ser combinada con un operador posicional absoluto. Además, los paréntesis pueden anidarse, resolviéndose las ecuaciones planteadas desde dentro hacia fuera, de la misma forma que las igualdades y polinomios matemáticos.

Estrategia de la Interrogación

Los lenguajes, sus órdenes y operadores son utilizados dentro del proceso de recuperación de información, la cual se encuentra almacenada en un repositorio, que suele ofrecer la forma de BD. La BD es consultada mediante la ejecución de búsquedas, expresiones que reúnen los elementos citados con anterioridad, y cuya resolución da como resultado aquellos elementos que responden a la lógica expresada en la búsqueda.

Con el concepto “estrategia de la interrogación” nos referimos a los posibles enfoques que se le puede dar a la planificación del proceso de recuperación de la información, tanto de la visión general de cómo se va a afrontar la búsqueda hasta la formulación de la ecuación concreta.

La estrategia debe ser un plan ideal de interrogación de la BD que incluya el objetivo de la búsqueda, el plan general y el plan específico de operación. El objetivo de la búsqueda se obtiene identificando qué tipo de información se necesita y sus características. Una vez definido el objetivo, debe establecerse un plan general de operación, que incluya una selección de la base o BD a consultar, las primeras aproximaciones a los términos a utilizar en las ecuaciones, así como las posibles relaciones lógicas. El plan específico de operación se pone en marcha una vez obtenidos los resultados del anterior y debe formular ecuaciones y utilizar términos con el mayor grado de precisión, establecer una

secuencia lógica con todo ello y redefinirlo si es preciso. Independientemente de ambos planes, resulta necesario conocer con anterioridad la respuesta a varias cuestiones que afectan a la interrogación de la BD, tales como el contenido y alcance de la BD, coste de consulta, lenguaje y operadores a utilizar durante las consultas, límites preestablecidos (por el usuario o el sistema)... Todas ellas afectan y modifican el enfoque del interrogador.

Tipos de Estrategia

En el momento actual, parece más adecuado utilizar el término para identificar el plan general de búsqueda. No existe una única ni perfecta aproximación a las estrategias de interrogación de BD. En la mayor parte de las ocasiones depende de la experiencia del usuario y de la calidad del contenido de los registros existentes en la BD, especialmente en lo que corresponde a su control terminológico. La estrategia depende, en gran medida, de la formación, intuición y experiencia del usuario. Tomando en consideración la intención del interrogador, la bibliografía señala que pueden existir varios tipos principales de búsqueda, que pueden clasificarse en dos grandes grupos, sin perjuicio de que puedan darse situaciones en las que se combinen:

- Categorización por objetivo:
 - Búsqueda de elemento conocido: se trata de búsquedas en las cuales el interrogador sabe cuál será la respuesta. Por ejemplo, en una biblioteca en la que estamos buscando un libro concreto (documento respuesta conocido) y realizamos la búsqueda por su ISBN.
 - Búsqueda de información específica: el interrogador busca una información específica dada, generalmente sobre un tema concreto y limitado, como trabajos publicados en un año o por un autor.
 - Búsqueda de información general: intenta buscar la información sobre una materia o asunto, de forma general, que obtenga una visión global del estado de la misma.
 - Exploración de la BD: se trata de conocer qué tipos de información y/o documentos se encuentran almacenados en la BD, a qué pueden responder y cómo pueden utilizarse.
- Categorización por plan de operación:
 - Búsqueda directa: se trata de una aproximación expeditiva, en la que se intenta resolver el problema con la formulación de una única consulta. Como puede deducirse, resulta difícil obtener buenos resultados con la misma.
 - Búsqueda “breve”: es una evolución de la anterior, en la que se trata de recuperar unos ítems significativos entre un gran número obtenido tras una sola ecuación.
 - Ampliación: comienza con ecuaciones muy restrictivas, que ofrezcan documentos pertinentes. Tras analizar la respuesta, el usuario puede ampliar o expandir las ecuaciones de búsqueda hasta recuperar toda la información existente. Puede ofrecer problemas si la ecuación inicial no es adecuada.
 - Restricción: opuesta a la anterior, formula ecuaciones que ofrecen resultados muy amplios, para posteriormente utilizar ecuaciones más restrictivas, hasta delimitar los documentos pertinentes.
 - Construcción de bloques: intenta establecer bloques de información que se correspondan con el objetivo de la búsqueda, para combinarlos entre sí de manera que se responda a la necesidad planteada de manera óptima.

La Exploración como Mecanismo de Recuperación

Las limitaciones inherentes al proceso de recuperación mediante ecuaciones han conducido a experimentar otras aproximaciones. Una de las más utilizadas es aquella que emplea la exploración, es decir, el acceso a los documentos mediante técnicas de visualización de parte de su contenido que puede ser relevante, y la posterior asociación con otros documentos de perfil similar. El usuario accede a un listado o enumeración de elementos descriptivos y, mediante un proceso de selección de elementos, va centrando el

objetivo de su búsqueda. Los criterios utilizados por el usuario se basan en la deducción y la asociación de conceptos (aproximación ésta similar a la que utiliza un sistema hipertextual) frente a la lógica de conjuntos que se plantea en un sistema de ecuaciones. Este tipo de representación es más adecuada para reflejar la polirepresentación que un concepto puede tener para un usuario individual. En cambio, la utilización de la exploración suele realizarse en entornos en los cuales el usuario no posee una idea clara de cuál debería ser la mejor táctica para aproximarse a la información que precisa. Por lo tanto, la cuestión clave a considerar en un sistema de exploración es combinar las ideas y esquemas del usuario con el esquema de organización de la información que ofrece el sistema. Ésta es la aproximación que pretenden desarrollar los enfoques cognitivos, poniendo su énfasis en el intermediario que debe existir entre el modelo del usuario y el modelo del sistema.

Revisión y Análisis de Resultados

El resultado de la ejecución de una ecuación de búsqueda es un conjunto de documentos que cumplen las condiciones expresadas en la ecuación. Se trata, a su vez, de un subconjunto del conjunto total de documentos existentes en el recurso o fuente de información consultado. Sin embargo, puede darse el caso de que la respuesta sea un número excesivamente elevado de documentos, o un número mínimo. Por otra parte, los documentos resultantes responden a la lógica y a las condiciones expresadas en la ecuación de búsqueda, lo cual no supone, como ya se ha señalado, que sean pertinentes a las necesidades del usuario. En realidad, es posible ejecutar ecuaciones perfectas, desde un punto de vista funcional (operadores, términos, ...), sin que los documentos resultantes reúnan las características que los harían deseables para el usuario.

Para superar esta posible distorsión en los resultados es necesario valorar y evaluar la respuesta a las ecuaciones planteadas. La primera modificación a realizar en la formulación de las ecuaciones afecta al número de respuestas obtenidas. En el caso de un excesivo número, se utilizan técnicas de restricción mediante la introducción de términos más específicos, se desechan términos generalistas o se limitan los truncamientos. En el caso de un número muy reducido, las acciones a tomar son las contrarias, es decir, utilización de términos más generales, incluyendo derivados y relacionados, limitación de los operadores más restrictivos, introducción de truncamientos, etc. Si se da la situación de ecuaciones correctas funcionalmente, pero sin respuesta adecuada, sería necesario replantear el proceso de recuperación, especialmente en la utilización de los lenguajes documentales y en la selección de fuentes.

Gestores de Contenido

Un **CMS (Content Management System)**, Sistema de Gestión de Contenidos o **Gestor de Contenidos**, es una aplicación web a la que podremos acceder a través de un navegador tras ser instalada en un servidor. A través de su panel de administración podremos crear, eliminar, modificar y en definitiva, gestionar el contenido de la “página web” (sitio web).

Por lo que también podríamos definirlo como una herramienta que nos permite la creación de una “página web” (sitio web) y su gestión por perfiles no técnicos.

¿Por qué surgieron los CMS?

No hace muchos años los sitios web estaban formados por páginas web estáticas codificadas en html. Existía la figura del webmaster, que era un técnico que se encargaba del mantenimiento de las páginas web del sitio. Por fortuna, las páginas web se modificaban pocas veces al año ya que todavía no se hacían blogs ni periódicos online que requieren una alta frecuencia de gestión del contenido. Además, modificar el contenido era tedioso, pues había que abrir el archivo html correspondiente a la página web en

cuestión que había que modificar y “bucear” entre el código html para realizar los oportunos cambios.

Un día surgió la necesidad de crear blogs, periódicos online y otros tipos de páginas web (sitios web) que requerían de frecuentes modificaciones. No se podían encargar todas las modificaciones al webmaster, había que encontrar alguna manera de que personas no técnicas pudieran crear y gestionar contenido de la página web (sitio web). Así aparecieron los CMS o Gestores de Contenidos.

Problemas, Beneficios y Ventajas de un CMS

Problemas de no usar un CMS

- Poca usabilidad de la interfaz.
- Pérdida de tiempo. Los tiempos para encontrar y editar una página son más largos.
- Solo pueden modificar contenidos personal con conocimientos HTML.
- Desorganización: Con una página sin CMS y con muchos contenidos puede ser un desastre localizar una página concreta de forma rápida.
- Necesidad de usar manuales de Dreamweaver, Frontpage, ...

Gracias al CMS podemos solucionar todos estos problemas, agilizando nuestro trabajo y permitiendo, sin muchos conocimientos técnicos, a cualquier persona a poder hacer uso de la página web de la empresa.

Beneficios del uso de un CMS

- Proceso de creación rápido y dinámico.
- Tiempo de ejecución más rápido para crear nuevas páginas y editar contenidos.
- Mayor consistencia del sitio web. Todo al alcance de tu mano.
- Mejora de la navegación del sitio.
- Mayor flexibilidad.
- Mayor seguridad.
- Menos contenido duplicado.
- Facilidad en la escalabilidad de la página web.
- Reducción de los costes de mantenimiento.

Ventajas

- **Ahorro de tiempo** : una de las mejores ventajas del uso de estos gestores es que tenemos la oportunidad de ahorrar tiempo en la creación, edición y administración de los contenidos. Sin necesidad de emplear otras herramientas para poder hacerlo.
- **Facilidad** : los gestores de contenido tienen la enorme ventaja de que pueden ser utilizados por las personas sin la necesidad de que tengan conocimientos en áreas del lenguaje de programación o diseño. La interfaz está hecha para que los usuarios empleen una herramienta con la cual puedan encontrar todo lo que necesitan al alcance de un solo click y de la forma más sencilla.
- **Creación** : los CMS permiten que las personas aún sin conocimientos en programación tengan la oportunidad de crear desde cero sus contenidos sin ayuda de nadie y de la forma que desean.
- **Diseño** : otra de las muchas ventajas que te ofrecen los gestores de contenidos es que tienes la posibilidad de elegir plantillas de diseño. Entre muchas que se encuentran para elegir según sean tus necesidades o gustos. No es necesario tampoco conocer sobre programación o diseño para tener un espacio web realmente estético e impactante, lo cual es muy importante.

Otra de las muchas ventajas que ofrecen, es que tienes la posibilidad de trabajar el SEO con ellos. Recordemos que para que un sitio web sea visible requiere de trabajo y posicionamiento para lograr el tráfico que necesita.

Front Office y Back Office del CMS

Los CMS se caracterizan por tener dos entornos:

- **Front Office** : es la **parte pública** de la página web (sitio web), a la que accedemos escribiendo la URL del sitio en la barra de direcciones del navegador web.
- **Back Office** : es la **parte privada** de la página web (sitio web) o lo que también se conoce como el **panel de administración** del sitio web. Desde aquí se puede gestionar el contenido del sitio web, su estructura, diseño y los diferentes elementos de configuración.

Para acceder al Back Office de un CMS habrá que escribir una url especial que dependerá del CMS utilizado. En el caso de WordPress habrá que añadir al nombre de dominio la palabra “wp-admin”, por ejemplo: <http://www.mipagina.es/wp-admin>.



Clasificación y Características de los CMS o Gestores de Contenidos

Los Gestores de Contenidos o CMS son aplicaciones web

Los Gestores de Contenidos son aplicaciones web especialmente diseñadas para crear páginas web. Las aplicaciones web son aquellas aplicaciones a las que se accede a través de un navegador web. Los Gestores de Contenidos o CMS como aplicaciones web que son, habitualmente necesitan de la compañía de una serie de elementos:

1. **Un servidor web** : Encargado de recibir las peticiones de los navegadores web de los clientes cuando solicitan una página web, de comunicarse con el módulo encargado de la ejecución del código y de enviar las páginas web resultado de la ejecución del código al navegador del cliente. El servidor web más utilizado es **Apache** .
2. **Módulo** encargado de ejecutar el código escrito en un lenguaje de programación y de enviar la página web resultante al servidor web (para la mayoría de CMS se utiliza el **módulo PHP** del servidor Apache).
3. **Un servidor de base de datos** . Encargado de almacenar los datos del sitio web. El más utilizado en los Gestores de Contenidos es sin duda el servidor de BD **MySQL** .

4. **Un lenguaje de programación** . El lenguaje de programación más utilizado para los Gestores de contenido más populares es **PHP** .

Clasificación de los CMS o Gestores de Contenidos por sus características

1. **Según el lenguaje de programación** empleado por el CMS para crear la página web, como por ejemplo Java, PHP, ASP.NET, Python, PERL. Tanto WordPress como los más conocidos gestores de contenidos están codificados en el Lenguaje de programación del lado del servidor PHP.
2. **Según la licencia** : Código abierto o Software propietario. Tanto el CMS WordPress como el resto de aplicaciones para crear páginas web más conocidas (Drupal, Joomla, Prestashop, etc) son Software abierto y gratuito.

Clasificación de los CMS o Gestores de Contenidos por su uso y funcionalidad

1. **Genéricos** : Tienen muchos posibles usos. Crear una página web corporativa, un blog, una tienda online, etc. Aquí podemos incluir CMS como: Joomla, Drupal, ... Y desde hace algún tiempo WordPress (comenzó siendo un Gestor de contenidos específico para la creación de Blogs).
2. **Blogs** : Son los CMS especialmente creados para la gestión de diarios personales. Son CMS de blogs WordPress, B2Evolution, Movable Type, Blogger, ...
3. **Comercio electrónico** : Son CMS creados específicamente para crear tiendas online. Algunos ejemplos son Magento, PrestaShop, Opencart, etc.
4. Existen **CMS específicos** para crear Foros, Wikis, CMS para cursos online como Moodle, etc.

Lista de los mejores CMS más utilizados

- **CMS WordPress** : Es el CMS más utilizado y mejor valorado para creación de blogs y webs. Está hecho en PHP y es gratuito.
- **CMS Drupal** : Es uno de los CMS más conocidos, es gratuito y open source. Está construido en PHP.
- **CMS Joomla** : Es otro CMS popular de código abierto y también creado en PHP. Es una evolución del CMS Mambo.
- **Prestashop CMS** : Es el CMS de ecommerce más conocido y mejor valorado. Podemos decir que es el WordPress de los ecommerce.
- **Magento CMS** : Es otro CMS para ecommerce de los más populares y mejor valorados. Ofrece muchos niveles de configuración. A diferencia de Prestashop, se requiere de conocimientos técnicos avanzados para utilizarlo.
- **Blogger** : aun hoy se sigue utilizando esta plataforma de gestión de contenidos, fue una de las primeras en hacer presencia en la red. Su forma de uso es gratuita y bastante sencilla, por lo que crear contenidos no genera ningún tipo de problemas.
- **LiveJournal** : dedicado a todas las personas que no cuentan con toda la experiencia requerida en el manejo de sitios web. Este gestor de contenidos permite que se puedan conectar blogs dependiendo de su temática, así como la clasificación de los mismos.

Como trabaja un CMS



Sindicación de Contenido

Se denomina Sindicación a la distribución masiva de contenidos en la web. A partir de la inclusión de algún nuevo contenido en un sitio, lo que se distribuye es una lista de enlaces junto con cierta cantidad de información adicional o metadata.

Los enlaces apuntarán a esos nuevos contenidos y la información adicional permitirá a los receptores evaluar si los contenidos son de su interés, en cuyo caso accederá a la versión completa simplemente siguiendo el enlace.

Los primeros sindicadores de contenido en línea fueron mega sitios de la magnitud de Yahoo y Excite. Su propuesta era muy clara: que sus visitantes pudieran acceder a información de orígenes muy diversos desde un lugar único.

Durante un tiempo, la sindicación resultó demasiado cara y trabajosa ya que se realizaba en base a la recuperación del título de cada página y la revisión de todo el HTML (que está concebido para mostrar contenidos pero no para organizarlos) para detectar los encabezados y enlaces para luego categorizarlos. Semejante tarea no estaba al alcance de cualquiera.

La gran novedad para la sindicación surgió de la utilización de archivos **XML**.

Conceptos

- **RSS** : Se corresponde con las siglas de Really Simply Syndication. Es un formato XML para la sindicación de contenidos. Es el más extendido, y permite distribuir contenidos sin necesidad de un navegador, utilizando un agregador de contenidos.



- **Agregador de contenidos** : Es el software que permite suscribirse a fuentes de noticias en RSS, por ello es también conocido como lector RSS o agregador de noticias.
- **Feed** : Es la fuente o canal web propiamente dicho, al que pueden suscribirse los usuarios.

Los archivos RSS

Un archivo RSS es la descripción estructural de un sitio web en formato XML.

RSS es un lenguaje surgido de la aplicación del metalenguaje XML. Por lo tanto, un archivo RSS no será más que un documento de texto compuesto por etiquetas acotadas entre los símbolos de mayor y menor, similares a las utilizadas XHTML.

El término RSS corresponde a **Rich Site Summary** o **Really Simple Syndication**.

Sindicación de Contenidos : Es el término técnico utilizado para designar un método o proceso que permite la notificación y envío de información recientemente publicada en la web. Por tanto, su principal objetivo es la organización y difusión de esta nueva información de un modo rápido y fiable. Parte del principio de suscripción, y se apoya en un conjunto de programas que permiten interpretar sus formatos.

Wikipedia dice : RSS son las siglas Really Simple Syndication (Sindicación Realmente Simple), un formato XML para sindicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS tales como Internet Explorer, entre otros (agregador).

Es interesante destacar que se trata de un formato que no está concebido para su visualización (como el HTML) sino para la interacción entre computadoras, ofreciendo la información en un formato estandarizado.

Para que este proceso resulte posible, un sitio web debe generar un feed o canal (el archivo RSS) que permanecerá alojado en el servidor tal como los demás archivos que lo componen.

Una vez que el feed está disponible, otros sistemas podrán accederlo y así enterarse de los nuevos contenidos que el sitio ofrece.

Hoy en día los sitios que permiten la creación y mantenimiento de blogs personales como Blogger y las aplicaciones que lo facilitan en cualquier dominio como WordPress han automatizado la generación de feeds, por lo que los usuarios solo deben manejar sus contenidos.

Sin demasiado misterio, los contenidos estarán entonces sindicados.

Para leer los feeds o canales RSS es necesario utilizar un tipo de programa denominado genéricamente agregador.

Los Lectores o Agregadores de feeds

Los archivos RSS, a diferencia de los XHTML, no son interpretados por los navegadores web y al abrirlos lo que hacen es mostrar el código XML que los compone.

Para visualizar directamente un feed es necesario utilizar un programa lector o agregador de feeds.

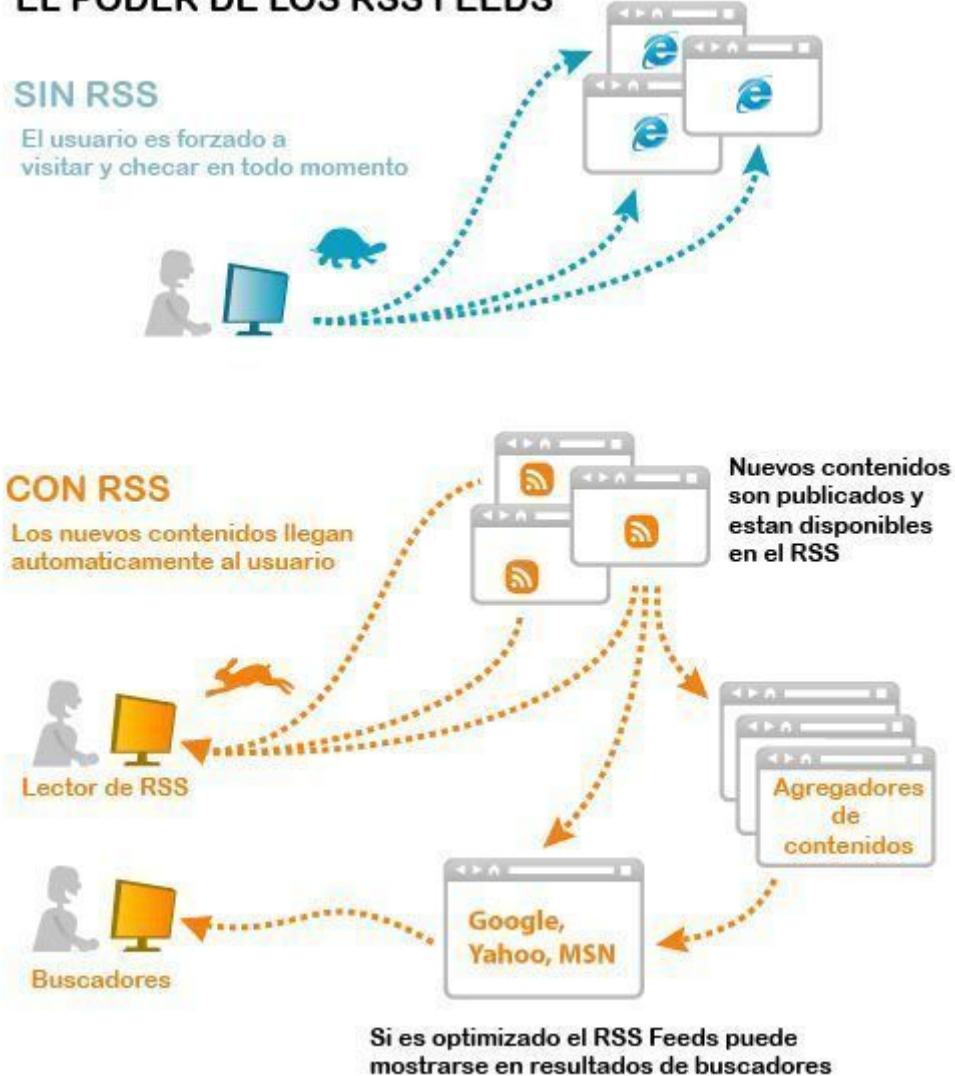
Hay distintos tipos de agregadores.

Las basados en web (usualmente denominados Portales) permiten la visualización en una página web. Un ejemplo típico de este tipo de agregador es Yahoo con su agregador MiYahoo! o el agregador de Bloglines.

Otros agregadores están integrados a clientes de correo o son clientes RSS exclusivamente.

Los agregadores ofrecen variedad de prestaciones especiales, tales como la inclusión de varios feeds relacionados en una única vista, el ocultamiento de entradas que ya han sido leídas y la categorización de feeds en áreas temáticas.

EL PODER DE LOS RSS FEEDS



©2007 Elliance, Inc. | www.elliance.com

Sistemas de Gestión de Flujos de Trabajos

Workflow o flujo de trabajo consiste en el estudio de aspectos operacionales de una actividad de trabajo, esto es, cómo se realizan y estructuran las tareas, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información y cómo se hace su seguimiento.

Una de las **aplicaciones de workflow** consiste en automatizar la secuencia de tareas, acciones o actividades para ejecutar el proceso, con el consiguiente seguimiento del estado de las etapas y las herramientas que son necesarias para gestionar esto. Esto a nivel real es muy sencillo y por eso es muy utilizado por las empresas.

Existen **tres tipos de actividad en los flujos de trabajo** : *actividades cooperativas, actividades colaborativas y actividades de coordinación* . También existen dos tipos de workflow principales: *workflow ad hoc* y *workflow procedimental* .

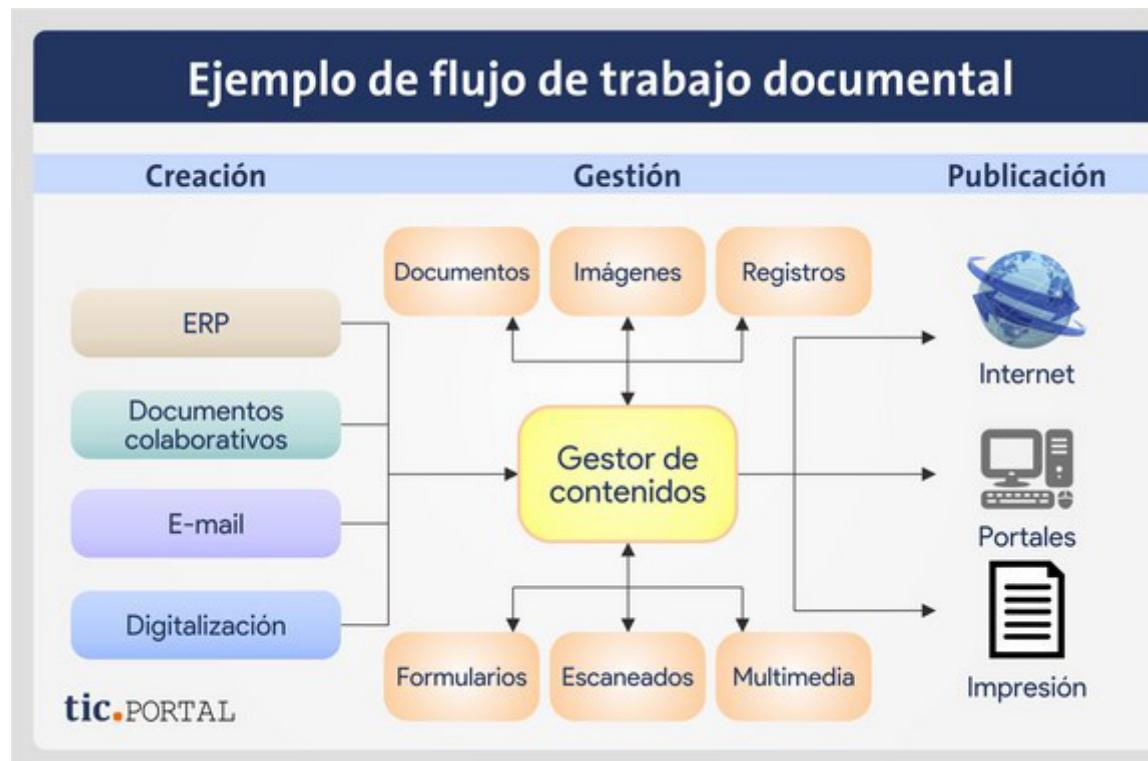
El principal **objetivo de los flujos de trabajo** consiste en reducir el tiempo y acelerar la realización de un trabajo mediante el acercamiento de procesos, personas y máquinas, incluso permitiendo trabajar en equipo desde diferentes lugares. Además de esto, puede facilitar la movilidad del personal, mecanizar y automatizar métodos y organización en la información, ofrecer mecanismos de control y seguimiento de procedimientos de la empresa, agilizar el proceso de intercambio de información y toma de decisiones de la empresa, independizar el flujo de trabajo y método de quien lo realiza, etc. Puede ser muy interesante en el trabajo de *gestión de stocks* o control de existencias así como también en la *gestión documental*.

Principalmente, el **workflow** busca seguir la realización y consecución de las tareas o trabajos por medio de una secuencia de tareas del proceso de negocio. De esta manera organiza y controla recursos, tareas y las reglas para completar este proceso buscando una mayor agilidad y la descentralización de actividades comerciales y administrativas principalmente.

Con esto se puede conseguir un control de todas las etapas a la vez que la **automatización de los procesos de trabajo**, por lo cual las tareas, información y documentos pasan por los participantes mediante unos procedimientos que se han establecido. Para ello en muchos casos se recurre a muchas aplicaciones informáticas y software que ayudan a controlar el flujo de trabajo en todos sus aspectos.

¿Qué es el flujo de trabajo y por qué es importante en un gestor documental?

En el contexto de los gestores documentales, se refiere al **movimiento automatizado de documentos** a través de una correlación de acciones relacionadas con el proceso empresarial. Dicho de una forma más sencilla, con un gestor documental que controla los flujos de trabajo cada documento queda ligado al estado en el que se encuentre en todo momento. Por ejemplo, una factura puede estar en diversos estados (recibida, aprobada, pagada, etc) y el administrador determinado podrá controlar en todo momento la situación de la misma.



El control de los flujos de trabajo supone la máxima automatización de los procesos empresariales y el control de las etapas, durante las cuales los documentos pasan de un

empleado a otro, según procedimientos previamente definidos. La etapa previa al control de flujos de trabajo es el control de flujos de información. Las empresas deben analizar **cómo la información llega, se almacena y se distribuye** por la compañía para generar un flujo de trabajo eficiente.

Beneficios del workflow management

No existen flujos de trabajo que funcionen de igual manera para todas las empresas. Sin embargo, muchas experimentan beneficios similares derivados.

- Mejora de la productividad del trabajo de los empleados con la automatización de procesos.
- Normalización de los métodos de trabajo mediante procedimientos preestablecidos.
- Optimización de la circulación de información interna.
- Ahorro de tiempo en tareas poco necesarias u obsoletas.

Sistemas de flujo de trabajo o workflow management system

Del mismo modo que el workflow management puede encontrarse dentro de un gestor documental, también se ha desarrollado como un sistema individual. Los sistemas de flujo de trabajo permiten automatizar y mejorar los procesos empresariales con el propósito de ahorrar tiempo y eliminar errores.

Entre las características esenciales que suelen presentar este tipo de sistemas destacan:

- Notificaciones por email: a través de las notificaciones por email, los administradores reciben información detallada del punto en el que se encuentra una tarea.
- SLA control status: se trata de una representación gráfica del estado de una tarea. Gracias a un código de colores se enfoca la importancia en aquellas etapas del proceso que necesitan de mayor atención o están experimentando algún problema.
- Formularios pre-completados: con el fin de evitar las pérdidas de tiempo a la hora de llenar formularios repetitivos, se aconseja la distribución de formularios parcialmente completos.
- Reasignación de tareas: no todos los procesos terminan funcionando de la forma en la que se planean. Por ello, y para evitar gastos económicos, el software de flujo de trabajo permite la reasignación de tareas.

Objetivos de los sistemas de flujo de trabajo (workflow)

Métodos y organización en el sistema de información

Uno de los principales objetivos de los sistemas de flujo de trabajo es reflejar, mecanizar y automatizar los métodos y la organización en el sistema de información. Y es que hay que tener en cuenta que hoy en día es esencial poder acceder a la información de forma fácil y eficaz y lo normal es que ésta esté en diferentes formatos, lo que puede provocar un problema de accesibilidad.

Procedimientos organizativos

El segundo objetivo de este tipo de sistemas es establecer los mecanismos de control y seguimiento de los procedimientos organizativos, algo que se consigue gracias a una normalización en la metodología de trabajo.

Método y flujo de trabajo

Por otro lado, los sistemas de flujo de trabajo tienen el objetivo de independizar el método y el flujo de trabajo de las personas que lo ejecutan.

Movilidad del personal

El cuarto objetivo de los sistemas de flujo de trabajo es facilitar la movilidad del personal. De hecho, permiten trabajar en equipo desde distintos lugares físicos.

Reingeniería de negocio

Otro de los objetivos es soportar procesos de reingeniería de negocio que es un método mediante el cual, en función de las necesidades del cliente, se rediseñan radicalmente los procesos principales de negocios, de principio a fin, con el objetivo de alcanzar mejoras espectaculares en medidas críticas de rendimiento, tales como costes, calidad, servicio y rapidez.

Toma de decisiones

El sexto objetivo es agilizar el proceso de intercambio de información y agilizar la toma de decisiones de una empresa, organización o institución. De hecho, con la implementación de este tipo de sistemas las decisiones son rápidas, ágiles y oportunas.

Servicio

También es importante tener en cuenta que con este tipo de sistemas se optimiza el servicio. En este sentido, hay que señalar que supone dar una respuesta más rápida a los clientes, además de transmitir una sensación de apuesta por la tecnología, lo que contribuye a motivar a los trabajadores.

Gestión del conocimiento

El último objetivo es la mejora de la gestión del conocimiento, una nueva cultura empresarial que se basa en gestionar las organizaciones situando los recursos humanos como el principal activo.

Aplicaciones/Sistemas Workflow, flujos de trabajo

Las aplicaciones Workflow automatizan la secuencia de acciones, actividades o tareas en la ejecución del proceso, permiten realizar un seguimiento de cada etapa del mismo y aportan las herramientas necesarias para su control o gestión del flujo de trabajo.

Un sistema Workflow va más allá y se caracteriza, principalmente, por una adecuada integración con sistemas de información actuales: BD, gestión documental, mensajería, ERP, etc, permitiendo la ampliación de un workflow, de un simple proceso a la integración de varios procesos de negocio interrelacionados.

En el mercado existen diversos tipos de herramientas Workflow, las principales son: **Workflow Corporativo, Workflow de Aplicación, Workflow Documental y Workflow de Producción**. Algunas de ellas se limitan a su área en particular y otras permiten la comunicación con aplicaciones externas de manera *síncrona* (esperando la respuesta antes de proseguir) y/o *asíncrona* (solamente deja un "mensaje" y recupera la respuesta más adelante).

Lenguajes de especificación de workflow

- **BPMN** (Business Procces Model and Notation): Modelo y Notación de Procesos de Negocio.
- **BPEL / WS-BPEL** (*Web Services* Business Process Execution Language): Lenguaje de ejecución de Procesos de Negocio *con Servicios Web* .
- **XPDL** (XML Process Definition Language): Lenguaje para la Definición de un Flujo de Trabajo.

- **YAML** (Yet Another Workflow Language): Lenguaje de workflow basado en patrones de Workflow.

Búsqueda de información: robots, spiders, otros

Un motor de búsqueda, también conocido como buscador, es un sistema informático que busca archivos almacenados en servidores web gracias a su “spider” (Web crawler). Un ejemplo son los buscadores de Internet (algunos buscan únicamente en la web, pero otros lo hacen además en noticias, servicios como Gopher, FTP, etc) cuando se pide información sobre algún tema. Las búsquedas se hacen con palabras clave o con árboles jerárquicos por temas; el resultado de la búsqueda es un listado de direcciones web en los que se mencionan temas relacionados con las palabras claves buscadas.

Como operan de forma automática, los motores de búsqueda contienen generalmente más información que los directorios. Sin embargo, estos últimos también han de construirse a partir de búsquedas (no automatizadas) o bien partir de avisos dados por los creadores de páginas (lo cual puede ser muy limitante). Los buenos directorios combinan ambos sistemas. Hoy en día Internet se ha convertido en una herramienta, para la búsqueda de información, rápida, para ello han surgido los buscadores que son un motor de búsqueda que nos facilita encontrar información rápida de cualquier tema de interés, en cualquier área de las ciencias, y de cualquier parte del mundo.

Se pueden clasificar en dos tipos:

1. **Índices temáticos** : Son sistemas de búsqueda por temas o categorías jerarquizados (aunque también suelen incluir sistemas de búsqueda por palabras clave). Se trata de BD de direcciones Web elaboradas “manualmente”, es decir, hay personas que se encargan de asignar cada página web a una categoría o tema determinado. Por ejemplo existen buscadores de fauna, flora, educación, música y de diferentes áreas.
2. **Motores de búsqueda** : Son sistemas de búsqueda por palabras clave. Son BD que incorporan automáticamente páginas web mediante “robots” de búsqueda en la red.

Clases de buscadores:

Buscadores jerárquicos (Arañas o Spiders)

Recorren las páginas recopilando información sobre los contenidos de las páginas. Cuando se busca una información en los motores, ellos consultan su BD y presentan resultados clasificados por su relevancia. De las webs, los buscadores pueden almacenar desde la página de entrada, a todas las páginas que residan en el servidor.

Si se busca una palabra, por ejemplo, “ordenadores”. En los resultados que ofrecerá el motor de búsqueda, aparecerán las páginas que contengan esta palabra en alguna parte de su texto.

Si consideran que un sitio web es importante para el usuario, tienden a registrarlas todas. Si no la consideran importante, sólo almacenan una o más páginas.

Cada cierto tiempo, los motores revisan los sitios, para actualizar los contenidos de su BD, por tanto puede que los resultados de la búsqueda estén desactualizados.

Los buscadores jerárquicos tienen una colección de programas simples y potentes con diferentes cometidos. Se suelen dividir en tres partes. Los programas que exploran la red -arañas (spiders)-, los que construyen la BD y los que utiliza el usuario, el programa que explota la BD.

Si se paga, se puede aparecer en las primeras páginas de resultados, aunque los principales buscadores delimitan estos resultados e indican al usuario que se trata de resultados esponsorizados o patrocinados. Hasta el momento, aparentemente, esta forma

de publicidad es indicada explícitamente. Los buscadores jerárquicos se han visto obligados a comercializar este tipo de publicidad para poder seguir ofreciendo a los usuarios el servicio de forma gratuita.

Ejemplo de arañas: Google, Bing, Hotbot.

Directarios

Una tecnología barata, ampliamente utilizada por gran cantidad de scripts en el mercado. No se requieren muchos recursos de informática. En cambio, se requiere más soporte humano y mantenimiento.

Los algoritmos son muchos más sencillos, presentando la información sobre los sitios registrados como una colección de directorios. No recorren los sitios web ni almacenan sus contenidos. Solo registran algunos de los datos de nuestra página, como el título y la descripción que se introduzcan al momento de registrar el sitio en el directorio.

Los resultados de la búsqueda, estarán determinados por la información que se haya suministrado al directorio cuando se registra el sitio. En cambio, a diferencia de los motores, son revisadas por operadores humanos, y clasificadas según categorías, de forma que es más fácil encontrar páginas del tema de nuestro interés.

Más que buscar información sobre contenidos de la página, los resultados serán presentados haciendo referencia a los contenidos y temática del sitio.

Su tecnología es muy barata y sencilla.

Ejemplo de directorios: Antiguos directorios, Open Directory Project, Yahoo!, Terra (antiguo Olé). Ahora, ambos utilizan tecnología de búsqueda jerárquica, y Yahoo! conserva su directorio.

Metabuscador

Permite lanzar varias búsquedas en motores seleccionados respetando el formato original de los buscadores. Lo que hacen, es realizar búsquedas en auténticos buscadores, analizan los resultados de la página, y presentan sus propios resultados, según un orden definido por el sistema estructural del metabuscador.

FFA - Enlaces gratuitos para todos

FFA (Free For All). Cualquiera puede inscribir su página durante un tiempo limitado en estos pequeños directorios. Los enlaces no son permanentes.

Buscadores verticales

Buscadores especializados en un sector concreto, lo que les permite analizar la información con mayor profundidad, disponer de resultados más actualizados y ofrecer al usuario herramientas de búsqueda avanzadas. Es importante resaltar que utilizan índices especializados de esta manera para acceder a la información de una manera más específica y fácil.

Ejemplos de este tipo de buscadores son: Trovit, Nestoria.

¿Qué es un crawler o arañas de la web y qué hacen?

¿Qué es un crawler?

El crawler, también conocido como araña de la web, es un software o webbot que se encarga de recorrer los enlaces de las páginas webs de una forma automática y sistemática.

¿Qué hace un crawler y cómo funciona?

Normalmente, un crawler dispone de un conjunto de inicial de URLs, conocidas como semillas, y va descargando las páginas web asociadas a las semillas y buscando dentro de éstas otras URLs.

Cada nueva URL encontrada se añade a la lista de URLs que la araña web debe visitar. Es decir, recoleta URL's para posteriormente procesarlas. Así, el motor de búsqueda creará un índice de las páginas descargadas para proporcionar búsquedas más rápidas.

Cuando un crawler visita un sitio web opta por una de estas dos alternativas:

- Buscar el archivo robots.txt y la meta etiqueta robots para ver las reglas que se han estipulado.
- Elaborar un índice de las páginas web que hay en su sitio. ¿Cómo? Explorando el contenido del texto visible, de varias etiquetas HTML y los hipervínculos en listados en la página.

Ejemplo: Googlebot

Diferencia entre los robots, spider y crawler

El ranking de los motores de búsqueda está basado en robots (arañas o crawlers).

Crawler

Se trata de un **software desarrollado para realizar una exploración en Internet** de una manera sistemática a través de la información percibida como relevante para su función. Capturan el texto de las páginas y los enlaces encontrados y por lo tanto permiten encontrar nuevas páginas. Es una de las bases de los motores de búsqueda, que son responsables de la indexación de sitios web, almacenarlos en la BD de los buscadores. Es también conocido como araña o Bot (robot).

El proceso que ejecuta un rastreador web se llama Web Crawler o rastreador .

Muchos sitios, en particular los motores de búsqueda utilizan rastreadores para mantener una BD actualizada. Los rastreadores web son usados básicamente para realizar una copia de todas las páginas visitadas para post-procesamiento por un motor de búsqueda que indexa las páginas descargadas para proporcionar búsquedas rápidas. Los rastreadores también se pueden utilizar para tareas de mantenimiento automatizadas en un sitio web, como la comprobación de enlaces o la validación de código HTML. Las spiders también pueden ser utilizadas para obtener los tipos específicos de información de páginas web, como direcciones de correo electrónico (más comúnmente como spam).

Los rastreadores de motores de búsqueda por lo general buscan información acerca de los permisos sobre el contenido. En especial hay dos maneras de bloquear un rastreador que indexe una página en particular (y los enlaces contenidos en ella). La primera, y más común, es a través del archivo robots.txt. La otra forma es a través de la etiqueta meta robots con el valor "noindex" o "nofollow", que sirve para no indexar (la página sí) y no por debajo (los enlaces en la página), respectivamente. También hay una tercera posibilidad, mucho menos explotado, que está utilizando el 'rel="nofollow"' para los enlaces, lo que indica que el rastreador que enlazan, en particular, no se debe seguir.

Araña

También conocido como Robot, Bot o Cadenas . Estos son los programas utilizados por los motores de búsqueda para navegar por Internet y descargar automáticamente contenido de sitios web. Metódicamente, expone el contenido que estime pertinente en el código fuente de los sitios, y almacena el resto en su BD. Por lo tanto, los motores de

búsqueda robots basados (arañas o crawlers) buscan en Internet después de la búsqueda de información y lo clasifican de acuerdo a los vínculos y también al contenido que se encuentra en las páginas de búsqueda, como el principal portal de búsqueda web, Google. Por lo tanto, cualquier página necesita ser trazada por el robot y por lo tanto pueden aparecer los resultados de búsqueda de los mecanismos implicados.

Posicionamiento y Buscadores (SEO)

¿Qué es?

Posicionamiento web, posicionamiento en buscadores o posicionamiento SEO se refiere a las técnicas que buscan que una página web aparezca en las primeras posiciones de los resultados en buscadores (Google, Yahoo, ...) para una serie de palabras o frases.

Conceptos

- **SEO** (Search Engine Optimization) o posicionamiento orgánico/natural.
- **SEM** (Search Engine Marketing) o posicionamiento de pago/publicitario.
- **SERPs** (Search Engine Results Page) o Página de resultados del Buscador.



Posicionamiento web natural u orgánico

Los buscadores proporcionan dos tipos de resultados: **enlaces patrocinados** o anuncios y **resultados orgánicos** o naturales:

- **Resultados Orgánicos, Posicionamiento “Gratis” o Posicionamiento Natural en Buscadores (SEO)** : los buscadores como Google aplican cierto criterio para decidir en qué orden deben aparecer los resultados de una búsqueda. Algunas de las características valoradas por los buscadores son, por ejemplo, la popularidad de la página web, su contenido, su velocidad de carga y otras cuestiones técnicas.
- **Enlaces Patrocinados, Posicionamiento “de Pago” o Marketing en Buscadores (SEM)** : la presencia de una página web en los resultados patrocinados se consigue con la compra de palabras clave al buscador (Google, Yahoo!, Bing, ...). Es importante destacar que el anunciante no paga por mostrar su anuncio, sólo paga cuando el usuario hace clic en él. A este tipo de publicidad se le llama también **PPC (Pago Por Clic)** .

Diferencias entre SEO y SEM

- Funcionamiento:
 - El criterio utilizado por los buscadores para mostrar los resultados naturales (SEO) es desconocido y las técnicas para mejorar el SEO están en las recomendaciones que de vez en cuando dan los propios buscadores y en la experiencia de quienes trabajan haciendo SEO.
 - Existen certificaciones oficiales de SEM, expedidas por los propios buscadores, que permiten formarse oficialmente en esta disciplina.
- Tiempo en obtener resultados:
 - Los resultados de las acciones para mejorar el SEO son observables a largo plazo.
 - Con el SEM se obtienen resultados de forma más inmediata.
- Garantías en la obtención de resultados:
 - En el SEO es imposible estimar, y mucho menos garantizar resultados.
 - En el SEM se puede estimar resultados.
- Costes:
 - La competencia en el SEO es tan alta que intentar aparecer en la primera página de resultados puede ser inútil, sobre todo para términos genéricos.
 - En el SEM, el precio de las palabras clave cambia en cada instante dependiendo de varios factores: competencia, país, idioma, ...
- Medición de resultados:
 - Es difícil medir con rigurosidad los resultados de las acciones para mejorar el SEO.
 - Los resultados del SEM se pueden medir con total precisión.

Objetivos del SEO

- **Definir las palabras claves** que son importantes para nuestra página, pues serán los términos utilizados por los usuarios para buscar información sobre contenido y soluciones que nosotros proveemos. Tenemos que tener en cuenta la Teoría del Long Tail aplicada a las búsquedas en la red; ya que, a pesar de que existe un número de búsquedas muy frecuentes, la mayoría de ellas son muy diferentes entre sí, y buscadores como Google se centraron en las pequeñas pero variadas búsquedas para obtener beneficios y componer su sistema de búsqueda.
- **Mejorar la visibilidad de la página web** . Los algoritmos empleados que emplean los buscadores para posicionar las páginas webs no son conocidos y van modificándose continuamente; consecuentemente, nadie puede tener la certeza de saber cómo posicionar en primer lugar una web en los SERPs, aunque se pueda trabajar para intentar aparecer en los primeros puestos. No obstante, se conocen

algunos de los aspectos que influyen en los algoritmos y que darán visibilidad a la web:

- Los propios de la programación de la página, que son “manipulables” y tenidos en cuenta para valorar la relevancia de la web, llamados **factores de relevancia on-page**.
 - Los que están relacionados con otras páginas webs a través de una estructura de vínculos que permiten navegar por toda la red de internet, llamados **actores de relevancia off-page**. En estos se incluye el **Social SEO**, que mayoritariamente está centrado en la capacidad de aportar enlaces entrantes desde los medios sociales hacia la web.
- **Aumentar el número de visitas** que están buscando lo que puede ofrecerle nuestra página; es decir, incrementar el tráfico cualificado que llega de los buscadores a la web.

Link Building

Linkbuilding o construcción de enlaces, es una técnica de SEO que consiste en conseguir que otras páginas web enlacen a la página que interesa que los buscadores consideren relevantes y la posicionen mejor en sus rankings. La técnica puede hacerse de manera natural, cuando otras webs enlazan sin previo acuerdo por algún hecho o dicho, o bien de manera artificial, cuando se simula que los enlaces se han conseguido de manera natural.

Esta se basa en el concepto de que uno de los factores que se incluyen dentro de la evaluación del ranking de una página es la cantidad de enlaces entrantes que tiene una página, concepto basado en el hecho de que el número de enlaces entrantes constituía uno de los factores evaluados en PageRank en 1999 por Google.

Las ventajas son:

- Posibilidad de medir la demanda y cantidad de personas que están buscando a través de una palabra clave.
- Efectividad del posicionamiento.
- Posicionamiento de la marca o branding.

Técnicas

- **Alta en directorios**: consiste en dar de alta la web en diferentes directorios, ya sean generales o temáticos.
- **Directorios de artículos**: consiste en escribir artículos para publicarlos en directorios que, a cambio del contenido, permiten incluir enlaces hacia una web.
- **Bookmarking**: se trata de guardar aquello que interesa posicionar en los buscadores en las diferentes webs de bookmarking.
- **Link baiting**: es una de las técnicas más valoradas por los buscadores pero una de las más difíciles de conseguir, ya que solo se consiguen cientos de enlaces a un artículo si este realmente aporta valor.
- **Intercambio de enlaces**: una buena forma de conseguir enlaces y una de las primeras que se empezaron a utilizar.
- **Compra de enlaces**: Más efectiva que el intercambio de enlaces pero también más cara. Para Google esta forma de conseguir enlaces es penalizable.
- **Enlaces desde foros**: otra forma para construir enlaces es de foros, agregando el link o enlace desde la firma del foro.
- **Otras técnicas**: envío de enlaces a bloggers, redes sociales, escribir revisiones, notas de prensa, entre otros.

Pasos para SEO y posicionamiento web

Posicionamiento a través de las Palabras clave

- Elige bien tus palabras clave.
- Comprueba la competencia.
- Mide la densidad de las palabras.
- Usa las palabras clave.
- Palabras clave en títulos y negrita.
- Mide y analiza tu posicionamiento natural para distintas palabras clave.

Configuración del Sitio

- Meta description y title.
- Url amigables y editadas.
- Creación y envío de sitemap a buscadores.
- Automatiza el envío de sitemaps.
- Transcribe el contenido audiovisual.
- Favicon.
- Evita el uso de cookies.
- Utiliza rel="autor".

Las imágenes

- Título y descripción.
- Especifica su tamaño.
- No escalar imágenes en Html.
- Optimización de imágenes para la web.
- Combinar imágenes usando CSS sprites.

Cuida los enlaces

- Anchor text diversificados.
- Comprueba los links rotos.
- Evita las redirecciones.
- Automatizar la búsqueda de links rotos.
- Conoce el PageRank.
- No enlaces contenido malicioso o ilegal.
- Busca en donde enlazan a tu competencia.
- Utiliza enlaces internos.

Evitar contenido duplicado

- Textos originales.
- No-index a nuestro contenido duplicado.
- Página inicial con sólo muestra.
- Descripción y Título Meta sin repetir.
- URL Canónica.
- Un mismo diseño para web y móvil.

Guía de Estilo

- Crear contenido divertidos y originales.
- Contenidos largos.
- Cuida a tus visitantes desde dispositivos.
- Publica periódicamente.
- Participa y conecta con tu comunidad.
- Protocolo después de cada artículo.

- Guest Blogging.
- Ofrece algún contenido de valor.
- Landing Page.
- Cuida al lector.

Evitar penalizaciones

- No poner palabras clave fuera de contexto.
- No poner texto escondido.
- Evita los errores de código que puedas.
- No te pases con el intercambio de enlaces.

Reduce el tiempo de carga de tu página

- Mide y mejora la velocidad de tu página.
- No abusar de los códigos en javascript.
- Elimina plugins de WordPress que no utilices.
- Pon javascript al final del código.
- Retrasar o diferir la carga de javascript.
- Ahorra y limpia tu código.
- Minimiza tu Css y Javascript.
- Combina tus Javascript.
- Usa la paginación.
- Reduce el número de consultas de DNS.
- Pocas llamadas http.
- Comprimir en gzip.
- Usar cache de la página.
- Usar cache para Javascript.
- Determinar una fecha de caducidad de la cache.
- Usar un CDN.
- No usar tablas anidadas en html.
- CSS externo.
- Javascript externo.
- Comprueba los tiempos de carga de cada página.

Herramientas imprescindibles

- Woorank.
- Web Ceo.
- Screaming Frog.
- All in One SEO Pack.
- SEO by Yoast.
- W3 Total Cache.
- SEO Chat Seo Tools.