

**UF1886**

***ADMINISTRACIÓN DEL  
GESTOR DE DATOS EN  
SISTEMAS ERP-CRM***



# ÍNDICE

## **UNIDAD DIDÁCTICA 1. GESTIÓN DE MANTENIMIENTO EN ALMACÉN DE DATOS.. 9**

1. Procesos de los sistemas de almacén de datos .....	11
1.1. Definición de la estructura de datos .....	12
1.2. Definición de la estructura de procesos .....	18
1.3. Integración de procesos .....	25
2. Parámetros de los sistemas que influyen en el rendimiento .....	25
2.1. Definición de parámetros de configuración.....	26
2.2. Optimización de recursos de software y tiempos de ejecución .....	42
RECUERDA.....	47
Preguntas de Autoevaluación.....	49

## **UNIDAD DIDÁCTICA 2. EVALUACIÓN DEL RENDIMIENTO ..... 51**

1. Herramientas de monitorización y de evaluación del rendimiento: características y funcionalidades.....	53
1.1. Definición de las funcionalidades que se desea evaluar.....	61
1.2. Definición de herramientas para evaluar el rendimiento del sistema ERP .....	63
RECUERDA.....	71
Preguntas de Autoevaluación.....	73

## **UNIDAD DIDÁCTICA 3. RESOLUCIÓN DE PROBLEMAS EN EL ALMACÉN DE DATOS ..... 75**

1. Trazas del sistema (logs) .....	77
1.1. Definición de sistemas para trazar los procesos entre los sistemas ERP, CRM y almacén de datos .....	80
2. Incidencias: identificación y resolución.....	86
2.1. Definición de procesos para identificar incidencias, control y resolución.....	88
2.2. Documentación de las tareas e incidencias realizadas .....	95
RECUERDA.....	97
Preguntas de Autoevaluación.....	99

## **UNIDAD DIDÁCTICA 4. TRANSPORTE DE COMPONENTES ENTRE ENTORNOS DE DESARROLLO, PRUEBA Y EXPLOTACIÓN EN ALMACÉN DE DATOS ..... 101**

1. El sistema de transmisión de información.....	103
1.1. Características en la transmisión de datos .....	104
1.2. Herramientas que intervienen en la transmisión de datos, configuración de parámetros .....	115
2. Entornos de desarrollo .....	119
2.2. Gestión del transporte entre los entornos de desarrollo, pruebas y explotación .....	123
RECUERDA.....	125
Preguntas de Autoevaluación.....	127

<b>UNIDAD DIDÁCTICA 5. ERRORES EN LA EJECUCIÓN DEL TRANSPORTE .....</b>	<b>129</b>
1. Tipos de errores.....	131
1.1. Verificación de la sintaxis .....	131
1.2. Identificación de incompatibilidades entre componentes.....	134
1.3. Documentación de errores encontrados.....	137
2. Resolución de problemas.....	137
2.1. Utilización de manuales para la resolución de problemas .....	138
2.2. Documentación de las soluciones aportadas.....	139
RECUERDA.....	141
Preguntas de Autoevaluación.....	143
<b>UNIDAD DIDÁCTICA 6. PROCESOS DE EXTRACCIÓN DE DATOS EN SISTEMAS DE ALMACÉN DE DATOS .....</b>	<b>145</b>
1. Características y funcionalidades.....	147
1.1. Definición de la estructura que interviene en los procesos de extracción de datos.....	152
1.2. Tratamiento de la información y transformación de estos datos para facilitar la toma de decisiones.....	153
1.3. Interpretación de resultados .....	154
2. Procedimientos de ejecución.....	155
2.1. Definición de procedimientos de extracción de datos entre los sistemas ERP, CRM y la base de datos.....	155
RECUERDA.....	161
Preguntas de Autoevaluación.....	163
<b>UNIDAD DIDÁCTICA 7. INCIDENCIAS EN EL PROCESO DE EXTRACCIÓN DE DATOS .....</b>	<b>165</b>
1. Trazas de ejecución.....	167
1.1. Establecimiento de trazas para estudiar incidencias .....	169
2. Resolución de incidencias .....	171
2.1. Solución y documentación de incidencias.....	171
RECUERDA.....	175
Preguntas de Autoevaluación.....	177
<b>ACTIVIDADES PRÁCTICAS .....</b>	<b>179</b>
Actividad Práctica RP3 .....	181
<b>RESPUESTAS A LAS PREGUNTAS DE AUTOEVALUACIÓN .....</b>	<b>183</b>

# UD1 Gestión de mantenimiento en almacén de datos



**UF1886 Administración  
del Gestor de Datos en  
Sistemas ERP-CRM**



# 1. Procesos de los sistemas de almacén de datos

Las Empresas han evolucionado en los últimos años hacia el planteamiento de estrategias basadas en los objetivos comprometidos con el cliente. Los mercados, cada vez más maduros y exigentes, demandan hoy más que nunca calidad de servicio y respuesta individualizada a necesidades puntuales de la clientela. Además, la reducción de los costes en el almacenamiento y procesamiento de información, ha permitido la aparición de potentes bases de datos que realizan tratamientos de información en tiempos de respuesta cada vez menores.

Los anteriormente destacados constituyen factores que han llevado a las Organizaciones a plantearse un mejor aprovechamiento y gestión de la información, uno de los activos que ha demostrado tener más importancia a la hora de fundamentar su éxito.

Los avances tecnológicos subsanan el continuo desencuentro que hasta ahora ha existido entre la tecnología y las reglas de negocio. Ciertos avances tecnológicos permiten hoy obtener rendimientos máximos en el uso de tecnologías hardware, sistemas de información para la dirección (EIS, DSS), procesamiento analítico on-line (OLAP), MDA (Análisis Multidimensional), Data Mart (DM) y Data Warehouse (DW).

En concreto, el **almacén de datos** o Data Warehouse es una herramienta tecnológica que permite mejorar la gestión de la información almacenada en las grandes bases de datos empresariales intentando resolver los problemas clásicos con los que se enfrentan los sistemas de información tradicionales como costes, redundancia de datos, dispersión de la información, escasa visión global corporativa y falta de fiabilidad.

Una adecuada implantación de este tipo de herramienta puede redundar en mejoras de desempeño empresarial, materializadas en considerables reducciones de costes, mejor comportamiento de la empresa y fomentar procesos de cambio en la Organización.

## 1.1. Definición de la estructura de datos

Una estructura de datos, o un tipo de datos estructurado, es un tipo de dato construido a partir de otros. Un dato de tipo estructurado está compuesto por una serie de datos de tipos elementales y alguna relación existente entre ellos. Normalmente, la relación suele ser de orden aunque puede ser de cualquier otro tipo.

Se dice que una estructura de datos que es homogénea cuando todos los datos elementales que la forman son del mismo tipo. En caso contrario, se dice que la estructura es heterogénea. Por ejemplo, el tipo de datos complejo es una estructura homogénea, tanto la parte real como la imaginaria se representan con datos reales.

Siempre que se utilice un dato en un programa debe estar determinado su tipo, para que el traductor sepa como debe tratarlo y almacenarlo. En el caso de datos de tipos elementales, el tipo de dato determina el espacio que se utiliza en memoria. Esto, puede no ocurrir si el dato es de un tipo estructurado.

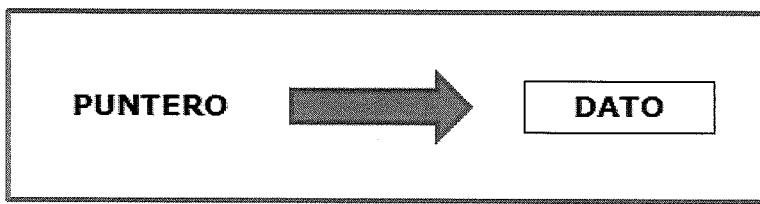
Una estructura de datos que siempre ocupa el mismo espacio en memoria, se dice que es **estática**. Por el contrario, si la memoria asignada a una determinada estructura de datos va variando durante la ejecución del programa, es decir, se realiza una asignación dinámica de memoria, se dice que es una estructura de datos **dinámica**.

### 1.1.1. Estructura de datos estáticas

Dentro de los tipos de estructuras de datos que ocupan siempre el mismo espacio en memoria tenemos los punteros, las cadenas y los arrays.

#### Punteros

Un puntero es un dato que indica la posición de otro dato. Su utilidad se pone de manifiesto en la construcción de estructuras de datos, ya que son ellos los que proporcionan los lazos de unión entre los elementos que constituyen las estructuras. Son importantes los punteros al principio y al final de la estructura.



Si, ocasionalmente, se necesita que un puntero no señale a ningún dato, se dice que el puntero tiene un valor nulo (puntero nulo).

### Cadenas

Una cadena es una secuencia de caracteres que se interpretan como un dato único. Las cadenas pueden tener longitud fija o variable. La longitud de la cadena se indica tanto por el número de caracteres que contiene ésta, indicado al principio de la misma, como por un carácter especial denominado fin-de-cadena. Sobre datos de tipo cadena se pueden realizar las siguientes operaciones:

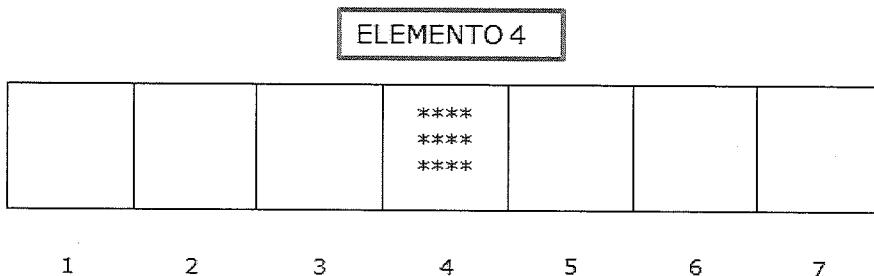
- **Concatenación:** consiste en formar una cadena a partir de dos ya existentes, juntando los caracteres de ambas.
- **Extracción de subcadena:** permite formar una cadena a partir de otra ya existente. La subcadena se forma tomando un tramo consecutivo de la cadena inicial.
- **Comparación de cadenas:** es posible comparar dos cadenas. Se considera menor aquella en que el primer carácter en que difieren ambas es menor.
- **Obtención de la longitud:** la longitud de una cadena es un dato de tipo entero, cuyo valor es el número de caracteres que contiene ésta.

### Arrays

El *array* (también llamado formación o matriz), es la estructura de datos más usual. Existe en todos los lenguajes de programación y en algunos es de las pocas estructuras de datos existentes (BASIC y FORTRAN).

Un array es una estructura de datos formada por una cantidad fija de datos del mismo tipo, cada uno de los cuales tiene asociado uno, o más índices, que determinan de forma única la posición del dato en el array.

Podemos imaginar un array como una estructura de celdas donde se pueden almacenar valores. En la figura podemos ver una matriz de un sólo índice que toma valores de 1 a 7.



En el array de la figura siguiente utilizamos dos índices con valores entre 1 y 3, el primero, y entre 1 y 5, el segundo. Cada elemento de esta matriz está representado por un par ordenado de números, el valor de los dos índices.

ELEMENTO 1,4				
1				*****
2				
3				
1	2	3	4	5

En general, al número de índices del array se le denomina número de dimensiones del array. La dimensión de la formación está dada por los valores máximos de los índices y, el número total de elementos es el producto de estos valores máximos. En los dos ejemplos anteriores, el número de dimensiones de los arrays son 1 y 2, las dimensiones son (7) y (3; 5) y los números totales de elementos son 7 y 15, respectivamente.

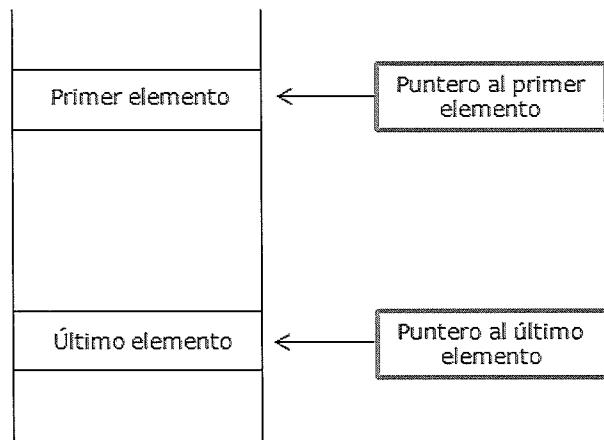
La principal operación que se puede realizar con los arrays es la selección, que consiste en especificar un elemento determinado del array. Esta operación se efectúa dando un valor para todos y cada uno de los índices del array. Con el elemento seleccionado se pueden realizar las operaciones propias de su tipo. Así, con cada elemento de un array real, una vez seleccionado, se pueden realizar las operaciones definidas para datos de tipo real (operaciones aritméticas).

### 1.1.2. Estructura Dinámicas de datos

Como ya se ha dicho, este tipo de estructuras ocupan un espacio en memoria que va evolucionando según el tamaño que dicha estructura vaya adquiriendo. Las estructuras de dinámicas que vamos a estudiar son: colas, pilas, listas encadenadas y árboles.

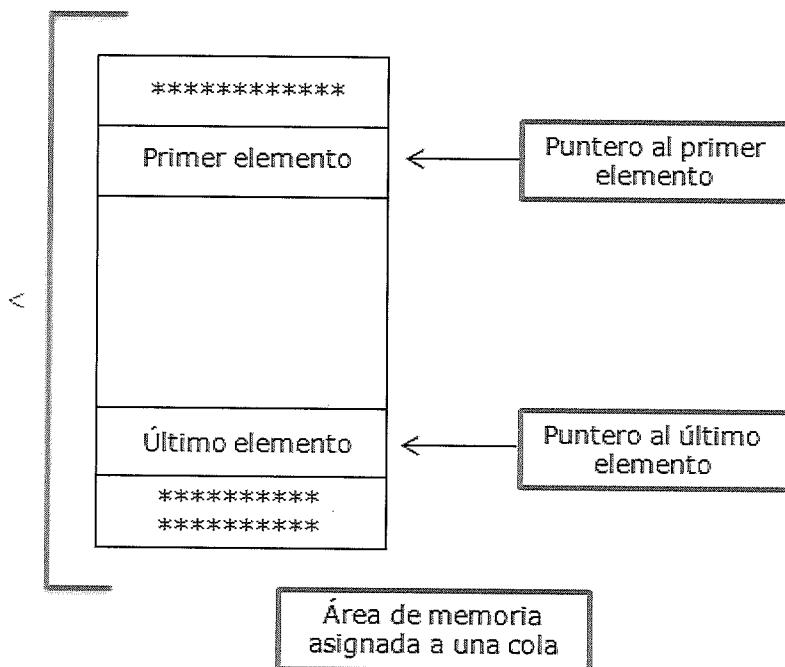
#### Colas (FIFO)

Una cola es una estructura de datos en la que el primer dato en entrar es el primer dato en salir. Es decir, es una estructura FIFO (*First In First Out*). Todo el mundo conocemos como funciona una cola, los nuevos se ponen al final, los servicios se prestan al principio y no está permitido "colarse". Las mismas reglas se aplican a las colas de datos almacenadas en la memoria de un computador.



Hay varias formas de implementar una cola en la memoria de un computador. Una forma simple consiste en almacenar los datos en posiciones de memoria adyacentes y utilizar punteros para el principio y el fin de la cola. Cuando un elemento se añade a la cola, el puntero de la parte posterior se ajusta para que señale al nuevo elemento. De manera similar, cuando un elemento se elimina de la cola, se ajusta el puntero delantero para que señale al nuevo primer elemento.

El problema de este método para implementar las colas es que las posiciones de memoria que ocupan, varían a medida que se añaden y eliminan elementos de la misma. La solución habitual consiste en asignar un área fija para almacenar la cola y permitir que se mueva en este área de manera circular. Un área de almacenamiento de esta forma se denomina buffer circular, y puede apreciarse en la figura siguiente.



Entre las aplicaciones que tienen las colas se encuentran el almacenamiento de datos en camino, entre un procesador y un periférico, o actuar como punto intermedio en las redes de comunicación de datos.

### Pilas (LIFO)

Una pila es una colección ordenada de datos a los que sólo se puede acceder por un extremo, denominado tope o cima de la pila. La pila es una estructura en la que el último elemento en entrar será el primero en salir, es decir, es lo que se denomina estructura LIFO (*Last In First Out*).

Podemos comparar esta estructura con una pila de platos colocada sobre un muelle. Cuando se añade un nuevo plato en lo alto de la pila, los demás bajan, cuando se retira un plato de la pila, los demás suben.

Igual que en el caso de las colas, van a existir dos punteros, uno que indica la posición tope de la pila, denominado puntero de pila, y otro que señala su base, denominado base de pila, y que mantiene el mismo valor mientras existe la pila. Cuando la pila está vacía el puntero de pila tiene el mismo valor que la base de pila.

La pila es una de las estructuras más importantes en computación. Se usa en cálculos, para pasar de un lenguaje de computador a otro y, para transferir el control de una parte del programa a otra.

Las operaciones que se pueden realizar tanto con las colas como con las pilas, son las siguientes:

- **Añadir o eliminar un elemento:** si es una cola podremos añadirlo o eliminarlo al final de la misma, y si es una pila al principio.
- **Acceder al primer elemento:** normalmente es el único al que se va a poder acceder directamente.
- **Acceder al elemento siguiente del último procesado:** este es el mecanismo normal de acceso tanto a colas como a pilas.
- **Saber si está vacía:** están vacías si no contienen ningún elemento.

## **Listas encadenadas**

Una lista es un conjunto ordenado de datos. Los elementos de la lista pueden insertarse o eliminarse en cualquier punto de la misma, por lo que es menos restrictiva que una pila o una cola.

La forma más sencilla de implementar una lista es hacer uso de un puntero que señale desde un dato al siguiente. También hay un puntero que señala al primer elemento de la lista, mientras que para el último se emplea un puntero nulo.

Una estructura de este tipo se denomina lista encadenada. Cada elemento de la lista consiste en una parte de datos y un puntero.

Una variación sobre la idea de una lista es el caso en el que el puntero del final de la lista señale al primer elemento. Esto crea lo que se denomina una lista circular.

Si los elementos de la lista están en orden alfabético o numérico, dichas listas se conocen como listas ordenadas.

## **Árboles**

Un árbol es una estructura que implica una jerarquía, en la que cada elemento está unido a otros bajo él. Cada dato en un árbol es un nodo de dicho árbol. El nodo más alto se denomina **raíz**. Cada nodo puede estar conectado a uno o más subárboles, que también responden a la estructura de un árbol. Un nodo, en la parte inferior, del que no cuelgue ningún subárbol se denomina nodo terminal u hoja.

Un tipo especial de árboles muy usados en computación son los árboles binarios. En ellos, de cada nodo pueden colgar, a lo más, dos subárboles, denominados subárbol derecho y subárbol izquierdo, que también son árboles binarios.

La forma usual de representar los árboles supone el uso de punteros. En un árbol binario cada nodo está constituido por una parte de datos y dos punteros. Uno, o ambos punteros, pueden tener un valor nulo si del nodo no cuelgan subárboles.

Son muy utilizados en informática. Las partes de muchos programas se enlazan como si se tratara de árboles. Los árboles se utilizan para representar operaciones aritméticas, y en búsquedas y ordenaciones.

## 1.2. Definición de la estructura de procesos

Un proceso es básicamente un programa en ejecución. Un programa ejecutable es un conjunto de instrucciones y datos almacenados en un fichero. Cuando lo que tiene ese programa se carga en la memoria y se pone en ejecución, se convierte en un **proceso**.

Un proceso es una entidad formada por los siguientes elementos principales:

- Una **imagen binaria** de un programa, cargada total o parcialmente en la memoria física. La imagen binaria está formada por las instrucciones y datos del programa.
- Un **área de memoria** para almacenar datos temporales, conocida como pila. La imagen binaria y la pila son el programa en sí mismo, pero para que el SO pueda controlar el programa hacen falta una serie de estructuras de datos. Las estructuras fundamentales son:
  - La tabla de páginas para traducir las direcciones virtuales generadas por el proceso en las direcciones físicas en la que se encuentra almacenado.
  - Una estructura de control, conocida como PCB, para que el sistema operativo pueda controlar su ejecución.

### 1.2.1. Estados de un proceso

Un proceso pasa por varios estados durante su ejecución. Los estados posibles para un proceso se muestran en la siguiente imagen:



Como vemos en el esquema, los nodos (nuevo, listo, etc.) representan los estados y los arcos, las acciones o eventos que llevan a un cambio de estado.

Definición de los estados:

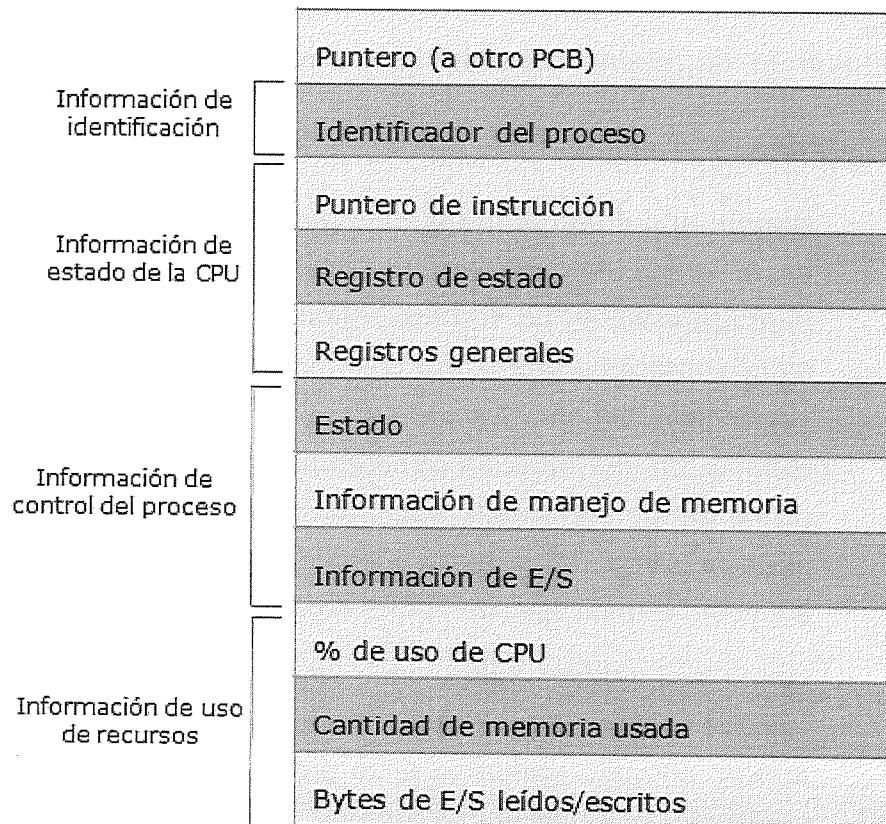
- **Nuevo**: el proceso se acaba de crear, pero aún no ha sido admitido en el grupo de procesos ejecutables por el sistema operativo. Habitualmente en un sistema operativo multitarea como Windows, nada más que un proceso se crea, éste resulta admitido, pasando al estado listo. Sin embargo, esto no tiene por qué ser siempre así. Por ejemplo, en una situación de sobrecarga temporal del sistema, el SO puede decidir retardar la admisión de los procesos nuevos. Así se alivia la carga del sistema, ya que hasta que un proceso no es admitido, éste no compite por los recursos del sistema.
- **Listo**: el proceso está esperando ser asignado al procesador para su ejecución. Una CPU clásica (con un solo núcleo) solo se puede dedicar en cada momento a un proceso. Los procesos que están preparados para ejecutarse permanecen en estado listo hasta que se les concede la CPU. Entonces pasan al estado "En ejecución".

- **En ejecución:** el proceso tiene la CPU y ésta ejecuta sus instrucciones.
- **En espera:** el proceso está esperando a que ocurra algún suceso, como por ejemplo la terminación de una operación de E/S.
- **Terminado:** el proceso ha sido sacado del grupo de procesos ejecutables por el sistema operativo. Después de que un proceso es marcado como terminado se liberarán los recursos utilizados por ese proceso, por ejemplo, la memoria.

### **1.2.2. Bloque de control de proceso o Process Control Block (PCB)**

Es una estructura de datos que permite al sistema operativo controlar diferentes aspectos de la ejecución de un proceso.

El PCB se organiza en un conjunto de campos en los que se almacena información de diversos tipos. Los campos típicamente mantenidos en el PCB de un proceso se muestran en la figura siguiente:



La información típica mantenida en el PCB puede clasificarse en cuatro categorías:

- **Información de identificación.** Esta información está integrada básicamente por el identificador del proceso (PID), que es un número que identifica al proceso. Este número es diferente para todos los procesos que se encuentran en ejecución.
- **Información de estado de la CPU.** Se trata de un conjunto de campos que almacenan el estado de los registros de la CPU cuando el proceso es suspendido.
- **Información de control del proceso.** Se trata de un conjunto de información que es utilizada por el sistema operativo para controlar diversos aspectos de funcionamiento del proceso. Pertenecen a esta categoría de información los siguientes campos:
  - Estado del proceso: Listo, en ejecución, etc.
  - Información de manejo de memoria: Como por ejemplo, la dirección física de memoria en la que se ubica la tabla de páginas del proceso.
  - Información de E/S: Lista de ficheros abiertos, ventanas utilizadas, etc.
- Información de uso de recursos. Se trata de un conjunto de información relativa a la utilización realizada por el proceso de los recursos del sistema, como por ejemplo, el porcentaje de utilización de la CPU, la cantidad de memoria usada o los bytes de E/S escritos y leídos por el proceso.

### **1.2.3. Planificación de procesos**

El objetivo de los sistemas multitarea es mantener múltiples programas en ejecución simultáneamente, pero como la CPU sólo puede ejecutar un programa de cada vez, hay que decidir quién se ejecuta en cada momento.

Se denomina planificación (scheduling) al mecanismo utilizado por el sistema operativo para determinar qué proceso (entre los presentes en el sistema) debe ejecutarse en cada momento.

## Planificación en sistemas de tiempo compartido

Los sistemas operativos más importantes del mercado actual (Windows, Linux, Mac OS y todas las versiones de Unix) se consideran **sistemas operativos de tiempo compartido**.

- **Objetivo prioritario de estos sistemas:** garantizar que el tiempo de respuesta de los programas se mantiene en unos valores admisibles para los usuarios. Cuando un usuario interacciona con un programa y le da una orden, quiere que el programa responda en un tiempo razonable. Para conseguir esto hay que hacer que el resto de programas que se encuentren en ejecución no monopolicen la CPU. Para ello, hay que ir repartiendo la CPU entre todos los programas, y además muy rápidamente, para que cada programa tenga una fracción del recurso CPU cada muy poco tiempo.
- **Esquema de funcionamiento:** a cada proceso en ejecución se le asigna un quantum, que representa el tiempo máximo que puede estar ocupando la CPU. Entonces un proceso abandona la CPU, o bien cuando se bloquea por una operación de E/S (pasando al estado "en espera"), o bien cuando expira su quantum (pasando al estado "listo".)

## Colas de planificación

Son unas estructuras de datos que organizan los PCBs de los procesos que se encuentran cargados en el sistema en función de su estado.

El SO planifica los procesos en función de la información mantenida en estas colas.

Estas estructuras se forman enlazando los PCBs de los procesos mediante punteros.

Existen dos tipos de colas:

- **Cola de procesos listos:** contiene a los procesos que se encuentran en el estado "listo". Debe indicarse una vez más que estos procesos son los que están preparados para ser asignados a la CPU.

- **Cola de dispositivo:** contiene los procesos que están esperando por un determinado dispositivo. Estos procesos se encuentran en el estado "En espera". Cada dispositivo tiene una cola asignada. Hay muchos dispositivos, como por ejemplo el disco, que son intensivamente utilizados por muchos procesos. Los procesos deben esperar ordenadamente para poder utilizar este recurso

### **Concepto de cambio de contexto**

Es el hecho de abandonar la ejecución de un proceso y poner en marcha otro proceso.

El cambio de contexto requiere salvar el estado que tienen los registros de la CPU justo antes de que ésta abandone el proceso que se saca de ejecución. Así, después, se podrá reanudar la ejecución de este proceso, justo en el punto en el que se suspendió su ejecución. El estado de los registros de la CPU se salva en el PCB del proceso.

### **Concepto de swapping (intercambio)**

Se trata de un mecanismo que permite sacar procesos de ejecución, salvándolos en el disco, para luego volver a ponerlos en ejecución cuando sea requerido.

El objetivo del "swapping" es aliviar al sistema, cuando su carga de trabajo es demasiado alta, suspendiendo temporalmente en el disco unidades de trabajo (procesos). Cuando la carga del sistema baja, se ponen de nuevo en ejecución los procesos temporalmente suspendidos. Al final se conseguirá mejorar el rendimiento global del sistema multitarea.

Debe observarse la clara diferencia existente entre los mecanismos del "cambio de contexto" y del swapping.

#### **1.2.4. Operaciones sobre procesos**

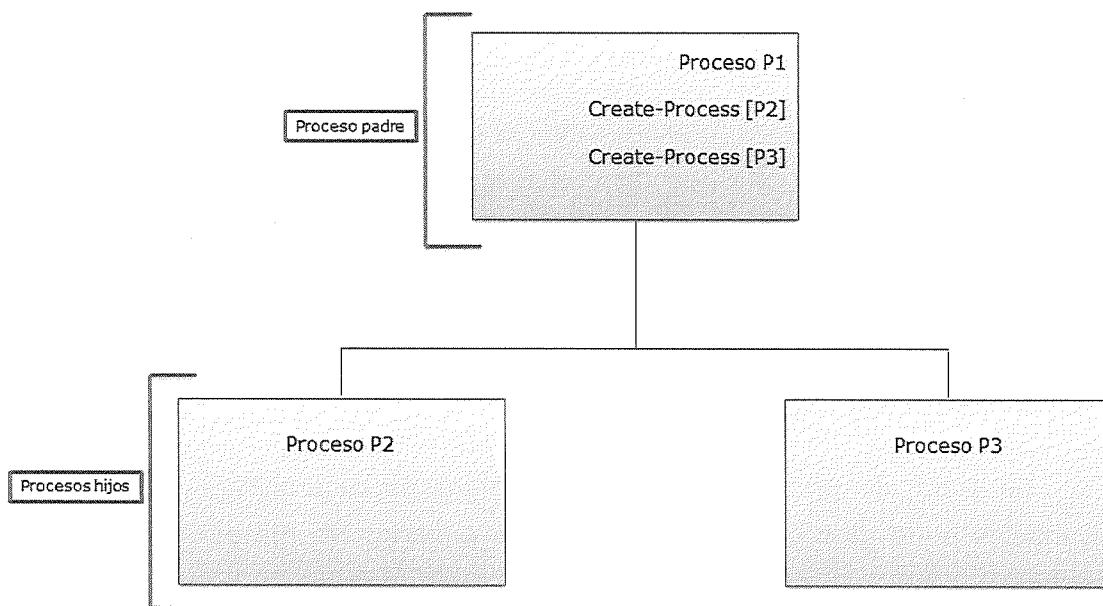
Los procesos tienen que poder ser creados y eliminados dinámicamente en el sistema. Debido a ello, el sistema debe proporcionar facilidades para llevar a cabo estas acciones con los procesos. Las funcionalidades básicas se indican a continuación.

## Creación de procesos

Todo sistema operativo debe proporcionar un servicio Create-Process, que será utilizado por un proceso para crear otro proceso.

Al proceso que solicita el servicio Create-Process se le denomina proceso padre, y al proceso que es creado mediante este servicio, proceso hijo.

A continuación mostraremos un ejemplo.



Este mecanismo de generación de procesos tiene como consecuencia que las relaciones de parentesco entre los procesos existentes en un sistema tenga estructura de árbol.

## Terminación de procesos

Un proceso puede terminar por sí mismo, o bien puede ser terminado por otro proceso, que generalmente sólo puede ser su proceso padre.

Un proceso termina por sí mismo llamando a un servicio del sistema, denominado normalmente Exit o Exit-Process.

Un proceso puede terminar la ejecución de un proceso hijo llamando a un servicio del sistema, conocido normalmente como Abort o Terminate-Process.

### 1.3. Integración de procesos

En la mayoría de las ocasiones los procesos son entidades totalmente aisladas: llevan a cabo su trabajo sin tener que comunicarse con otros procesos o programas. Sin embargo las cosas en la realidad no son tan sencillas. En muchas ocasiones, los programas o procesos necesitan intercambiar información entre sí.

La cooperación entre procesos requiere que estos se comuniquen. A continuación se indican los mecanismos básicos de comunicación:

- **Memoria compartida:**

- Se basa en que los procesos que desean comunicarse comparten una misma región de memoria física. Para llevar a cabo la comunicación, uno escribe y otro lee de la región de memoria compartida.
- Los procesos utilizan servicios del sistema operativo para compartir la región.

- **Paso de mensajes:**

- Los procesos utilizan una pareja de servicios del sistema operativo para comunicarse. Estos servicios son conocidos habitualmente como Send y Receive.
- Para llevar a cabo la comunicación un proceso ejecuta la función Send y el otro Receive, intercambiando de esta forma un bloque de información que recibe el nombre de mensaje.

## 2. Parámetros de los sistemas que influyen en el rendimiento

Los ERP están creados para adaptarse a las necesidades de cada empresa. Esto se logra en el momento de la configuración o parametrización de los procesos de acuerdo a las necesidades de cada caso concreto.

## 2.1. Definición de parámetros de configuración

Un **parámetro** es "un dato o factor que se toma como necesario para analizar o valorar la situación", "una variable que, en una familia de elementos, sirve para identificar cada uno de ellos mediante su valor numérico" según la definición de la Real Academia Española.

En la configuración del sistema operativo es importante tener en cuenta los parámetros así como los usuarios.

Es por ello que a continuación se procederá a explicar en detalle:

- Definición de los parámetros de configuración del sistema operativo.
- Definición de usuarios en el sistema operativo.

La velocidad de la CPU es un factor de rendimiento clave de la plataforma de sistema.

La función principal de los sistemas operativos consiste en gestionar recursos de hardware y software de manera común.

Un **sistema operativo** es, en primer lugar, un administrador de recursos, por ello es importante poder determinar con qué efectividad administra sus recursos un sistema determinado.

Generalmente hay un gran potencial de mejora en el uso de los recursos existentes, pero:

- Muchas instalaciones realizan muy poco o ningún control y evaluación.
- Cuando se hacen controles específicos se generan grandes cantidades de datos que muchas veces no se sabe cómo interpretar.

Las instalaciones rara vez cuentan con personal versado en las técnicas de análisis de rendimiento.

Durante los primeros años del desarrollo de las computadoras el hardware representaba el costo dominante de los sistemas y debido a ello los estudios de rendimiento se concentraban en el hardware.

Actualmente y según la tendencia apreciable:

- El software representa una porción cada vez mayor de los presupuestos informáticos.
- El software incluye el S. O. de multiprogramación / multiproceso, sistemas de comunicaciones de datos, sistemas de administración de bases de datos, sistemas de apoyo a varias aplicaciones, etc.
- El software frecuentemente oculta el hardware al usuario creando una máquina virtual, que está definida por las características operativas del software.

Un software deficiente y / o mal utilizado puede ser causa de un rendimiento pobre del hardware, por lo tanto es importante controlar y evaluar el rendimiento del hardware y del software.

### **Tendencias importantes que afectan a los aspectos del rendimiento**

Con los avances en la tecnología de hardware los costos del mismo han decrecido drásticamente y todo hace suponer que esta tendencia continuará.

Los costos de trabajo (personal) han ido aumentando:

- Significan un porcentaje importante del costo de los sistemas informáticos.
- Se debe reformular el aspecto del rendimiento del hardware base y medirlo de manera más adaptada a la productividad humana.

El advenimiento del microprocesador en la década de 1.970:

- Ha permitido bajar considerablemente el costo de los ciclos de cpu.
- Ha desplazado el foco de atención de la evaluación del rendimiento a otras áreas donde los costos no disminuyeron proporcionalmente; ej.: utilización de dispositivos de entrada / salida.

También influyen en los puntos de vista sobre la evaluación del rendimiento aspectos tales como:

- Construcción de redes.
- Procesamiento distribuido.

Las conexiones se hacen con redes y no sólo con computadoras específicas:

- Se puede disponer de cientos o miles de sistemas de computación.
- Se puede acceder a complejos sistemas de comunicaciones de datos.

### **Necesidad del control y de la evaluación del rendimiento**

Los objetivos corrientes en la evaluación del rendimiento generalmente son:

- **Evaluación de selección:** el evaluador debe decidir si la adquisición de un sistema de computación es apropiada.
- **Proyección del rendimiento:** el evaluador debe estimar el rendimiento de un:
  - Sistema inexistente.
  - Nuevo sistema.
  - Nuevo componente de hardware o de software.
- **Control del rendimiento:** el evaluador acumula datos del rendimiento de un sistema o componente existente para:
  - Asegurar que el sistema cumple con sus metas de rendimiento.
  - Ayudar a estimar el impacto de los cambios planeados.
  - Proporcionar los datos necesarios para tomar decisiones estratégicas.

En las primeras fases del desarrollo de un nuevo sistema se intenta predecir:

- La naturaleza de las aplicaciones que correrán en el sistema.
- Las cargas de trabajo que las aplicaciones deberán manejar.

Durante el desarrollo e implementación de un nuevo sistema se intenta determinar:

- La mejor organización del hardware.
- Las estrategias de administración de recursos que deberán implantarse en el S. O.
- Si el sistema cumple o no con sus objetivos de rendimiento.

Frecuentemente son necesarios procesos de configuración de los sistemas para que puedan servir a las necesidades.

Los procesos de sintonización del sistema tienden a mejorar el rendimiento en base a ajustar el sistema a las características de la instalación del usuario.

### **Mediciones del rendimiento**

El rendimiento expresa la manera o la eficiencia con que un sistema de computación cumple sus metas.

El rendimiento es una cantidad relativa más que absoluta pero suele hablarse de medidas absolutas de rendimiento, ej.: número de trabajos atendidos por unidad de tiempo.

Algunas mediciones son difíciles de cuantificar, ej.: facilidad de uso.

Otras mediciones son fáciles de cuantificar, ej.: accesos a un disco en la unidad de tiempo.

Las mediciones de rendimiento pueden estar:

- Orientadas hacia el usuario, ej.: tiempos de respuesta.
- Orientadas hacia el sistema, ej.: utilización de la cpu.

Algunas mediciones del rendimiento comunes son:

- **Tiempo de regreso:** tiempo desde la entrega del trabajo hasta su regreso al usuario (para procesamiento por lotes).
- **Tiempo de respuesta:** tiempo de regreso de un sistema interactivo.
- **Tiempo de reacción del sistema:** tiempo desde que el usuario presiona "enter" hasta que se da la primera sección de tiempo de servicio.

Las anteriores son cantidades probabilísticas y se consideran como variables aleatorias en los estudios de:

- Simulación.
- Modelado de sistemas.

Otras medidas del rendimiento utilizadas son:

- **Varianza de los tiempos de respuesta:** (o de otra de las variables aleatorias consideradas) es una medida de dispersión. Si es pequeña indica tiempos próximos a la media. Si es grande indica tiempos alejados de la media. Es una medida de la predecibilidad.
- **Capacidad de ejecución:** es la medida de la ejecución de trabajo por unidad de tiempo.
- **Carga de trabajo:** es la medida de la cantidad de trabajo que ha sido introducida en el sistema. El sistema debe procesar normalmente para funcionar de manera aceptable.
- **Capacidad:** es la medida de la capacidad de rendimiento máxima que un sistema puede tener siempre que el sistema esté listo para aceptar más trabajos y haya alguno inmediatamente disponible.
- **Utilización:** es la fracción de tiempo que un recurso está en uso. Es deseable un gran porcentaje de utilización pero éste puede ser el resultado de un uso ineficiente. Cuando se aplica a la cpu se debe distinguir entre; uso en trabajos productivos de aplicación y uso en sobrecarga del sistema.

## Técnicas de evaluación del rendimiento

### 1. Tiempos

Los tiempos proporcionan los medios para realizar comparaciones rápidas del hardware.

Una posible unidad de medida es el "mips": millón de instrucciones por segundo.

Los tiempos se usan para comparaciones rápidas; se utilizan operaciones básicas de hardware.

### 2. Mezclas de instrucciones

Se usa un promedio ponderado de varios tiempos de las instrucciones más apropiadas para una aplicación determinada; los equipos pueden ser comparados con mayor certeza de la que proporcionan los tiempos por sí solos.

Son útiles para comparaciones rápidas del hardware.

### *3. Programas del núcleo*

Un programa núcleo es un programa típico que puede ser ejecutado en una instalación.

Se utilizan los tiempos estimados que suministran los fabricantes para cada máquina para calcular su tiempo de ejecución.

Se corre el programa típico en las distintas máquinas para obtener su tiempo de ejecución.

Pueden ser útiles para la evaluación de ciertos componentes del software, por ejemplo compiladores; pueden ayudar a determinar qué compilador genera el código más eficiente.

### *4. Modelos analíticos*

Son representaciones matemáticas de sistemas de computación o de componentes de sistemas de computación.

Generalmente se utilizan los modelos de:

- Teoría de colas.
- Procesos de Markov.

Requieren un gran nivel matemático del evaluador y son confiables solo en sistemas sencillos, ya que en sistemas complejos los supuestos simplificadores pueden invalidar su utilidad y aplicabilidad.

### *5. Puntos de referencia (o programas de comparación del rendimiento)*

Son programas reales que el evaluador ejecuta en la máquina que se está evaluando.

Generalmente es un programa de producción:

- Típico de muchos trabajos de la instalación.
- Que se ejecuta con regularidad.

El programa completo se ejecuta en la máquina real con datos reales.

Se deben seleccionar cuidadosamente los puntos de referencia para que sean representativos de los trabajos de la instalación.

Los efectos del software pueden experimentarse directamente en vez de estimarse.

#### *6. Programas sintéticos*

Combinan las técnicas de los núcleos y los puntos de referencia.

Son programas reales diseñados para ejercitar características específicas de una máquina.

#### *7. Simulación*

Es una técnica con la cual el evaluador desarrolla un modelo computarizado del sistema que se está evaluando.

Es posible preparar un modelo de un sistema inexistente y ejecutarlo para ver cómo se comportaría en ciertas circunstancias; se puede evitar la construcción de sistemas mal diseñados.

Los simuladores son muy aplicados en la industria espacial y de transportes.

Los simuladores pueden ser:

- **Manejados por eventos:** son controlados por los eventos producidos en el simulador según distribuciones probabilísticas.
- **Manejados por libreto:** son controlados por datos obtenidos de forma empírica y manipulados cuidadosamente para reflejar el comportamiento anticipado del sistema simulado.

#### *8. Control del rendimiento*

Es la recolección y análisis de información relativa al rendimiento del sistema existente.

Permite localizar embotellamientos con rapidez.

Puede ayudar a decidir la forma de mejorar el rendimiento.

Puede ser útil para determinar la distribución de trabajos de varios tipos; permitiría aconsejar el uso de compiladores optimizadores o compiladores rápidos y sucios.

El control del rendimiento puede hacerse por medio de técnicas de hardware o de software.

Los monitores de software:

- Generalmente son económicos.
- Pueden distorsionar las lecturas del rendimiento debido a que consumen recursos del sistema.

Los monitores de hardware:

- Generalmente son más costosos.
- Su influencia sobre la operación del sistema es mínima.

Los monitores:

- Producen grandes cantidades de datos que deben ser analizados manualmente o por sistema.
- Indican con precisión cómo está funcionando un sistema.
- Son de mucha ayuda para evaluar sistemas en desarrollo y tomar las decisiones de diseño adecuadas.

Los rastreos de ejecución de instrucciones (trace) o rastreos de ejecución de módulos pueden revelar embotellamientos.

Un rastreo de ejecución de módulos puede mostrar que se está ejecutando un pequeño subconjunto de módulos durante gran parte del tiempo:

- Los diseñadores deberán optimizarlos para mejorar en gran medida el rendimiento del sistema.
- Se podría eliminar el costo de optimización de los módulos poco usados.

### 2.1.1. Configuración y uso de los parámetros

La configuración de parámetros para el sistema operativo dependerá del sistema operativo con el cual se este trabajando.

Durante la instalación el sistema operativo nos pedirá parámetros de configuración básica como zona horaria o nombre del equipo, además de la contraseña de administrador/root.

Una vez instalado nuestro sistema operativo ya lo tenemos preparado para la instalación de utilidades y aplicaciones que necesitamos para su uso.

Concretamente, en la configuración de Windows, actualizar el sistema mediante Windows Update es la más básica medida para ayudar a proteger y mejorar las capacidades de tu PC. Y es muy fácil configurarlo para que ese proceso sea casi automático.

La mayor parte de los distintos elementos del sistema operativo se configuran desde el **Panel de Control**.

Para acceder al Panel de Control hay un ícono en la parte derecha del menú que aparece al pulsar el botón Inicio. También suele aparecer en muchos sistemas (según esté configurado) como una carpeta más al explorar el sistema (en MiPC, por ejemplo), al final de la estructura de carpetas.

Su manejo es muy sencillo ya que cada elemento del panel de control funciona como un programa independiente, por lo que basta explorar el Panel de Control, abrir los elementos correspondientes y seguir las instrucciones que indica el sistema.

Entre los elementos/parámetros configurables del Panel de Control tenemos, entre otras:

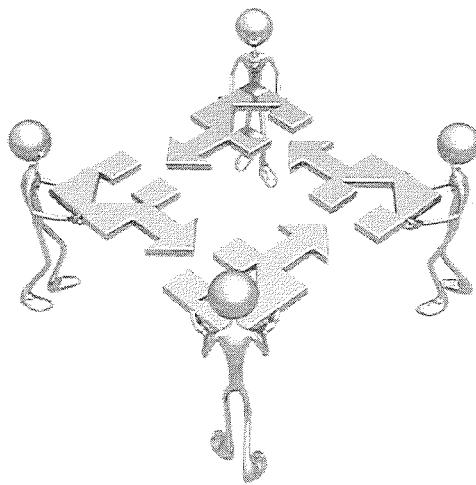
- Agregar o quitar hardware (periféricos), y configurar las distintas opciones de los mismos. Por ejemplo, instalar o mantener una impresora.
- Agregar o quitar software (programas), o reinstalarlos con distintas opciones.
- Configurar las conexiones a redes (locales o Internet).

- Administrar las cuentas de usuario, añadiendo o suprimiendo cuentas al equipo.
- Fijar las opciones de idioma, fecha, hora, etc.
- Configurar las opciones de seguridad antivirus.
- Controlar opciones avanzadas para el manejo del hardware.

Algunas de las opciones para configurar el sistema no están accesibles desde el Panel de Control. Por ejemplo, para cambiar el fondo de escritorio o el salvapantallas hay que actuar directamente sobre el propio escritorio (botón derecho, personalizar), y para cambiar el programa que abrirá cierto tipo de archivo hay que acudir al menú Herramientas / Opciones del Explorador de Windows.

### 2.1.2. Los gestores de datos y el rendimiento

Los **gestores de datos** son un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información de una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.



Hay diferentes tipos de gestores de datos en función del manejo que se hacen de ellos, es decir según funcionen en ordenadores personales con poca memoria o grandes sistemas que funcionan con sistemas de almacenamiento especiales.

Es importante tener claro los diferentes tipos de gestores, por lo que a continuación se detallarán estas diferencias, así como:

- Modelos lógicos.
- Modelos conceptuales.

Y las principales diferencias entre ellos.

### **Definición diferentes tipos de gestores de datos**

Cada sistema gestor de datos puede utilizar un modelo diferente para los datos. Por lo que hay modelos conceptuales diferentes según el sistema gestor de bases de datos que se utilicen.

La diferencia entre los distintos sistemas de gestión de datos está en que proporcionan diferentes modelos lógicos.

Las principales diferencias entre los modelos lógicos y conceptuales son las siguientes:

- El modelo conceptual es independiente del sistema gestor de base de datos que se vaya a utilizar. El lógico depende de un tipo de sistema de gestión de datos en particular.
- El modelo lógico está más cerca del modelo físico, el que utiliza internamente el ordenador.
- El modelo conceptual es el más cercano al usuario, el lógico es el encargado de establecer el paso entre el modelo conceptual y el modelo físico del sistema.

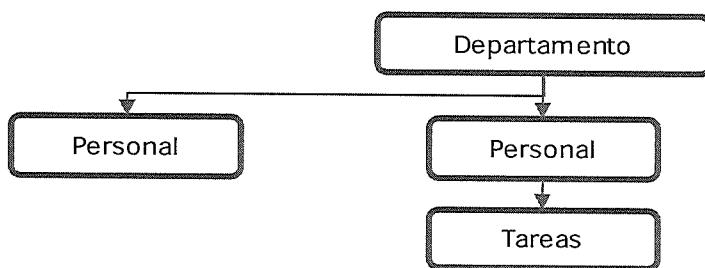
### **Modelos lógicos**

#### *Modelo jerárquico*

Es utilizado por los primeros gestores de datos. Es también conocido como modelo de árbol ya que se utiliza una estructura en árbol para organizar los datos.

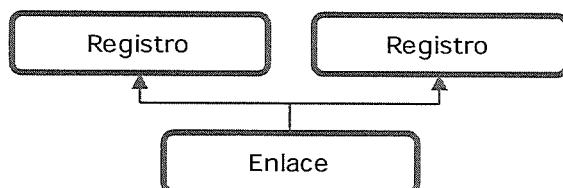
La información es organizada en orden jerárquico, en el que la relación entre las entidades siempre es del tipo padre/hijo. De este modo hay una serie de nodos que contienen atributos y que se relacionarán con nodos hijos de tal forma que puede haber más de un hijo para el mismo padre, pero solo un padre para un hijo.

En este modelo los datos se almacenan en estructuras lógicas conocidas como segmentos. Los segmentos se relacionan entre sí utilizando arcos.

*Modelo en red*

Es un modelo ampliamente aceptado. Organiza la información en registros, denominados nodos, y enlaces. En los registros se almacenan los datos, mientras que los enlaces permiten relacionarlos. Las bases de datos en red son parecidas a las jerárquicas, pero en este caso puede haber más de un padre.

En este modelo se pueden representar perfectamente cualquier tipo de relación entre los datos pero esto dificulta su manejo.

*Modelo relacional*

En este modelo los datos se organizan en tablas donde los datos se relacionan. Es el modelo más popular de todos.

*Modelo de bases de datos orientados a objetos*

Tras la aparición de la programación orientada a objetos, surge la necesidad de crear bases de datos adaptadas a este lenguaje. La programación orientada a objetos permite la cohesión de datos y procedimientos, haciendo que se diseñen estructuras que poseen datos en las que se definen los procedimientos que pueden realizar con los datos. En las bases de datos orientadas a objetos se utiliza esta misma idea.

A través de esta idea se pretenden que estas bases de datos consigan arreglar las limitaciones de las relationales.

Se supone por tanto que estas son las bases de datos de tercera generación, pero siguen sin reemplazar a las relacionales, aunque son el tipo de base de datos que más está creciendo en los últimos años.

#### *Bases de datos objeto-relacionales*

Pretenden ser un híbrido entre el modelo relacional y el orientado a objetos. El problema de la base de datos orientados a objetos es que requieren de una reinversión de capital y esfuerzo para convertir las bases de datos relacionales en bases de datos orientadas a objetos.

En las bases de datos objeto relacional se intenta conseguir una compatibilidad relacional dando la posibilidad de integrar mejoras de la orientación a objetos.

Estas bases de datos se basan en el estándar SQL 99. En este estándar se añade a las bases relacionales la posibilidad de almacenar procedimientos de usuarios, tipos definidos por el usuario, recursos para consulta, etc.

#### **Modelos conceptuales**

Estos modelos representan la información de forma absolutamente independiente al sistema gestor de base de datos. Los esquemas internos de las diferentes bases de datos no captan suficientemente bien la semántica del mundo real, de ahí que primero haya que pasar por uno o dos esquemas previos más cercanos al mundo real.

Algunos de los ejemplos de modelos conceptuales son:

- Modelo entidad-relación.
- Modelo RM/T.
- Modelos semánticos.

#### **Diferentes tipos de parámetros**

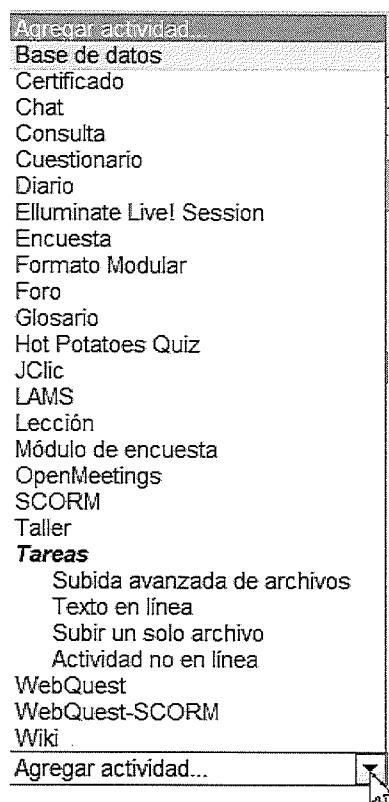
Algunos de los parámetros más comunes en los sistemas de gestión de bases de datos son:

- Redireccionamiento.
- Ruta de programas.

- Ruta de menús.
- Ruta de mensajes.
- Ruta de archivo de trabajo.
- Ruta de bases de datos.
- Idiomas implícitos.
- Separador de campos repetibles.
- Lista inicial de autodigitado.
- Modo de edición inicial.
- Caracteres gráficos para recuadros.
- Manejo de memoria expandida.
- Manejo de redes.
- Parámetros para meno de teclas de función.

### **2.1.3. Configuración de una base de datos**

Para configurar una base de datos, hemos de activar la edición y entrar en el bloque de actividades y seleccionar Base de datos:



Se abre un cuadro de diálogo con los siguientes campos:

- Ajustes generales.
- Ajustes comunes al módulo.

Vamos a ver los campos a llenar en el primer punto ya que el segundo es común a todas las actividades moodle.

### **Ajustes generales**

- Nombre: se pone un nombre a la base de datos.
- Introducción: se describe lo que se pretende. En el ejemplo hemos escrito:
  - Busca un recurso educativo que te pueda ser útil para la materia que impartes.
  - Rellena la base de datos cumplimentando:
    - Tu nombre y apellidos.
    - Nombre del recurso aportado.
    - Dirección web del recurso.
    - Descripción: utilidad del recurso.

## Actualizando Base de datos en tema 1

### Ajustes generales

Nombre\*

Introducción\*

Trebuchet  Idioma   Deshabilitar

**1.** Busca un recurso educativo que te pueda ser útil para la materia que impartes.

**2.** Rellena la base de datos cumplimentando:

Ruta:

Disponible en  junio   Deshabilitar

Disponible a  junio   Deshabilitar

Visible en  junio   Deshabilitar

Visible a  junio   Deshabilitar

Entradas requeridas

Entradas requeridas antes de verse

No. máximo de entradas

Comentarios

¿Se requiere aprobación?

¿Permitir calificar mensajes?  Usar calificaciones

Calificación

### Configuración de la base de datos

Disponible en  enero   Deshabilitar

Disponible a  abril   Deshabilitar

Visible en  abril   Deshabilitar

Visible a  abril   Deshabilitar

Entradas requeridas

Entradas requeridas antes de verse

No. máximo de entradas

Comentarios

¿Se requiere aprobación?

¿Permitir calificar mensajes?  Usar calificaciones

Calificación

- Disponible en y Disponible:** fechas en las que estará disponible la base de datos. Se puede deshabilitar.

- **Visible en y Visible:** fechas en las que estará visible la base de datos. Se puede deshabilitar.
- **Entradas requeridas:** número de entradas que se exige enviar a cada participante. Se puede seleccionar desde 1 hasta 50 entradas. Los usuarios verán un mensaje recordatorio que les informa si no han enviado el número de entradas exigido. La actividad no se considerará completada en tanto el usuario no haya enviado el número de entradas exigido.
- **Entradas requeridas antes de verse:** número de entradas que un participante debe enviar antes de que se le permita ver cualquier entrada en esta actividad de la base de datos.
- **Nº máximo de entradas:** número máximo de entradas que un participante puede enviar en esta actividad.
- **Comentarios:** permite que los usuarios hagan comentarios a cada entrada seleccionando Sí.
- **¿Se requiere aprobación?** La selección Sí, permite a un profesor/a aprobar las entradas antes de que los estudiantes puedan verlas.
- **¿Permite calificar mensajes?** Si activamos esta opción podrán calificarse las entradas.
- **Calificación:** permite elegir la escala de calificación de las entradas.

Una vez configurada la base de datos Guardamos los cambios.

## 2.2. Optimización de recursos de software y tiempos de ejecución

Al usuario final de un programa sólo le interesan dos cosas: que el programa sea fácil de manejar y que su ejecución sea lo más rápida posible. Para mejorar este último aspecto cada día se tienen más en cuenta las optimizaciones en el código del programa.

El objetivo de las técnicas de optimización es mejorar el programa objeto para que nos dé un rendimiento mayor. La mayoría de estas técnicas vienen a compensar ciertas ineficiencias que aparecen en el lenguaje fuente, ineficiencias

que son inherentes al concepto de lenguaje de alto nivel, el cual suprime detalles de la máquina objeto para facilitar la tarea de implementar un algoritmo.

Existen diversas técnicas de optimización que se aplican al código generado para un programa sencillo. Por programa sencillo entendemos aquel que se reduce a un solo procedimiento o subrutina. Las técnicas de optimización a través de varios procedimientos se reducen a aplicar las vistas aquí a cada uno de los procedimientos y después realizar un análisis interprocedural. Este último tipo de análisis no lo vamos a desarrollar en este epígrafe.

Partiendo de un programa sencillo, obtenemos código intermedio de tres direcciones [AH077], pues es una representación adecuada del programa sobre la que emplear las diversas técnicas de optimización. A partir de aquí dividimos el programa en bloques básicos [AHO86], o secuencia de sentencias en las cuales el flujo de control empieza al principio y acaba al final, sin posibilidad de parar o de tener saltos. Las sentencias de un bloque básico constituyen una unidad sobre la cual se aplican las optimizaciones locales.

Estas optimizaciones se pueden dividir en:

- **Optimizaciones que no modifican la estructura.** Son:
  - Eliminación de sub-expresiones comunes.
  - Eliminación de código muerto.
  - Renombrar variables temporales.
  - Intercambio de sentencias independientes adyacentes.
- **Transformaciones algebraicas.** Son aquellas transformaciones que simplifican expresiones y/o reemplazan operaciones costosas de la máquina por otras menos costosas.

Además de este tipo de optimizaciones locales a un bloque básico, existe otro tipo de optimizaciones aún más locales, pues su ámbito se reduce a una breve secuencia de instrucciones. A este tipo de optimización local se le llama **optimización peephole**, e intenta mejorar el rendimiento del programa por medio de reemplazar esa breve secuencia de instrucciones objeto por otra secuencia más corta y/o más rápida. Hay varios tipos de optimización peephole, siendo los más usuales los siguientes:

- Eliminación de instrucciones redundantes.

- Optimizaciones en el flujo de control.
- Simplificaciones algebraicas.
- Uso de instrucciones máquina específicas.

Debido a la naturaleza de este tipo de optimización, su salida es susceptible de ser optimizada de nuevo, con lo que serán necesarias varias pasadas para lograr la máxima optimización. Una aplicación de este tipo de optimización se puede encontrar en el kit ACK desarrollado por Tanembaum [Tane82], donde se describen 123 combinaciones distintas de sentencias de código intermedio con sus correspondientes equivalencias.

Las técnicas de optimización global se basan todos ellos en el análisis global de flujo de datos. Este análisis se realiza para el código de todo el programa, es decir, a lo largo de los distintos bloques básicos que forman el código del programa. Suponiendo que tenemos la información que nos proporciona este análisis, hay dos tipos de optimizaciones importantes que se realizan: la localización y la asignación global de registros para las variables, y las optimizaciones que se realizan en los bucles.

### **2.2.1. Localización y asignación de registros**

Para una máquina con registros, lo común en los procesadores actuales, las instrucciones cuyos operandos están en los registros de la máquina son más cortas y más rápidas que aquellas que tratan con operandos que están en la memoria. Es por tanto importante decidir qué variables se deben almacenar en los registros (localización) y en qué registro se debe almacenar cada variable (asignación).

Existen diversas estrategias para la localización y asignación de los registros. Es frecuente asignar algún número fijo de registros que contengan las variables más usadas en un bucle interno, sirviendo los registros restantes para las variables locales a cada bloque básico.

### **2.2.2. Optimizaciones en bucles**

Habitualmente, un programa pasa la mayor parte del tiempo de la ejecución en un trozo de código pequeño. A este fenómeno se le conoce como la regla 90-10, queriendo decir que el 90% del tiempo es pasado en el 10% del código. Este 10%

del código suele estar constituido por bucles, y de ahí la importancia de una correcta optimización del código que forma parte de los bucles.

Las principales optimizaciones que se realizan en los bucles son las siguientes:

- Movimiento de código.
- Eliminación de variables inducidas.
- Sustitución de variables costosas por otras menos costosas.

Y también se suelen aplicar (aunque con menor importancia):

- Expansión de código (loop unrolling).
- Unión de bucles (loop jamming).

### **2.2.3. Tiempo de ejecución de un logaritmo**

Para poder tener una medida del tiempo de ejecución de un programa, se debe pensar en los factores que tienen influencia en dicho valor. Inicialmente, se pueden citar los siguientes:

- La velocidad de operación del ordenador en el que se ejecuta. Es diferente ejecutar el programa en un micro 80386 que en un Pentium de 150 Mhz.
- El compilador utilizado (calidad del código generado). Cada compilador utiliza diferentes estrategias de optimización, siendo algunas más efectivas que otras.
- La estructura del algoritmo para resolver el problema.

Además de la estructura del algoritmo, se debe tener en cuenta que el número de datos con los cuales trabaja un programa también influye en su tiempo de ejecución. Por ejemplo, un programa para ordenar los elementos de un vector, se demora menos ordenando un vector de 100 posiciones que uno de 500. Eso significa que el tiempo de ejecución de un algoritmo debe medirse en función del tamaño de los datos de entrada que debe procesar. Esta medida se interpreta según el tipo de programa sobre el cual se esté trabajando.



## RECUERDA

- El **almacén de datos** o Data Warehouse es una herramienta tecnológica que permite mejorar la gestión de la información almacenada en las grandes bases de datos empresariales intentando resolver los problemas clásicos con los que se enfrentan los sistemas de información tradicionales como costes, redundancia de datos, dispersión de la información, escasa visión global corporativa y falta de fiabilidad.
- Una **estructura de datos**, o un tipo de datos estructurado, es un tipo de dato construido a partir de otros. Un dato de tipo estructurado está compuesto por una serie de datos de tipos elementales y alguna relación existente entre ellos. Normalmente, la relación suele ser de orden aunque puede ser de cualquier otro tipo.
- Una estructura de datos que siempre ocupa el mismo espacio en memoria, se dice que es **estática**. Por el contrario, si la memoria asignada a una determinada estructura de datos va variando durante la ejecución del programa, es decir, se realiza una asignación dinámica de memoria, se dice que es una estructura de datos **dinámica**.
- Dentro de los tipos de estructuras de datos que ocupan siempre el mismo espacio en memoria tenemos los **punteros**, las **cadenas** y los **arrays**. Las estructuras de dinámicas son: **colas**, **pilas**, **listas encadenadas** y **árboles**.
- Un proceso es básicamente un programa en ejecución. Un programa ejecutable es un conjunto de instrucciones y datos almacenados en un fichero. Cuando lo que tiene ese programa se carga en la memoria y se pone en ejecución, se convierte en un **proceso**.
- El objetivo de los sistemas multitarea es mantener múltiples programas en ejecución simultáneamente, pero como la CPU sólo puede ejecutar un programa de cada vez, hay que decidir quién se ejecuta en cada momento.
- En la mayoría de las ocasiones los procesos son entidades totalmente aisladas: llevan a cabo su trabajo sin tener que comunicarse con otros procesos o programas. Sin embargo las cosas en la realidad no son tan sencillas. En muchas ocasiones, los programas o procesos necesitan

intercambiar información entre sí.

- Un **parámetro** es "un dato o factor que se toma como necesario para analizar o valorar la situación", "una variable que, en una familia de elementos, sirve para identificar cada uno de ellos mediante su valor numérico" según la definición de la Real Academia Española.
- Un **sistema operativo** es, en primer lugar, un administrador de recursos, por ello es importante poder determinar con qué efectividad administra sus recursos un sistema determinado.
- La configuración de parámetros para el sistema operativo dependerá del sistema operativo con el cual se este trabajando.
- Los **gestores de datos** son un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información de una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.
- Al usuario final de un programa sólo le interesan dos cosas: que el programa sea fácil de manejar y que su ejecución sea lo más rápida posible. Para mejorar este último aspecto cada día se tienen más en cuenta las optimizaciones en el código del programa.

## Preguntas de Autoevaluación

### 1. Confirma si la siguiente afirmación es verdadera o falsa:

"El almacén de datos o Data Warehouse es una herramienta tecnológica que permite mejorar la gestión de la información almacenada en las grandes bases de datos empresariales intentando resolver los problemas clásicos con los que se enfrentan los sistemas de información tradicionales como costes, redundancia de datos, dispersión de la información, escasa visión global corporativa y falta de fiabilidad".

- a) Verdadero.
- b) Falso.

### 2. Una estructura de datos que siempre ocupa el mismo espacio en memoria, se dice que es:

- a) Dinámica.
- b) Estática.
- c) Rotativa.

### 3. Un programa ejecutable es un conjunto de instrucciones y datos almacenados en un fichero. Cuando lo que tiene ese programa se carga en la memoria y se pone en ejecución, se convierte en un:

- a) Array.
- b) Proceso.
- c) Gestor de datos.

**4. Completa la siguiente afirmación con la opción correcta:**

"Un \_\_\_\_\_ es un dato o factor que se toma como necesario para analizar o valorar la situación", "una variable que, en una familia de elementos, sirve para identificar cada uno de ellos mediante su valor numérico según la definición de la Real Academia Española".

- a) Proceso.
- b) Software.
- c) Parámetro.

**5. La mayor parte de los distintos elementos del sistema operativo se configuran desde el:**

- a) Panel de Control.
- b) Panel de Ajustes.
- c) Panel de Administración.

# UD2 Evaluación del rendimiento



**UF1886 Administración  
del Gestor de Datos en  
Sistemas ERP-CRM**



# 1. Herramientas de monitorización y de evaluación del rendimiento: características y funcionalidades

Para evaluar el rendimiento se necesita ejecutar un determinado software (distintas aplicaciones) y ver cuál es la utilización del hardware. Esta actuación nos aportará la información necesaria para la realización de modificaciones que mejoren las prestaciones, como es la selección del hardware, aplicaciones etc.



Existen una serie de pruebas que nos van a permitir estimar el rendimiento del sistema, ya sea antes o durante su explotación, si tenemos en cuenta que las necesidades software o hardware pueden variar a lo largo del tiempo.

Entre las distintas formas de evaluación tenemos:

- **Monitorización:** existen muchas herramientas que nos permiten visualizar el empleo de los recursos del sistema con la particularidad de que los datos obtenidos varían en el tiempo según el uso que se haga en función de su actividad o programas que se estén ejecutando.

- **Pruebas benchmarks:** recogen información comparativa entre distintos sistemas.
- **Modelado:** es una medida de evaluación que simula a un sistema donde existe un elemento que aún no se ha instalado.

Nosotros vamos a centrarnos en ver con más detalle en este punto las herramientas de monitorización.

La monitorización es una técnica de uso generalizado para supervisar, analizar y evaluar el comportamiento y rendimiento de los sistemas informáticos que están en funcionamiento.

En general, la monitorización de sistemas informáticos hace referencia a todo lo relativo a la extracción de información que permita conocer qué está sucediendo en ellos. Esto conlleva una serie de problemas que a menudo no son fáciles de resolver. Entre otros, hay que decidir qué datos hace falta recoger, hay que saber dónde se encuentran estos datos, y por último, es necesario acceder a ellos de manera que perturben lo mínimo en funcionamiento del sistema, para grabarlos con vistas a su posterior análisis.

La toma de medidas en un sistema informático se puede hacer mediante dos posibles técnicas:

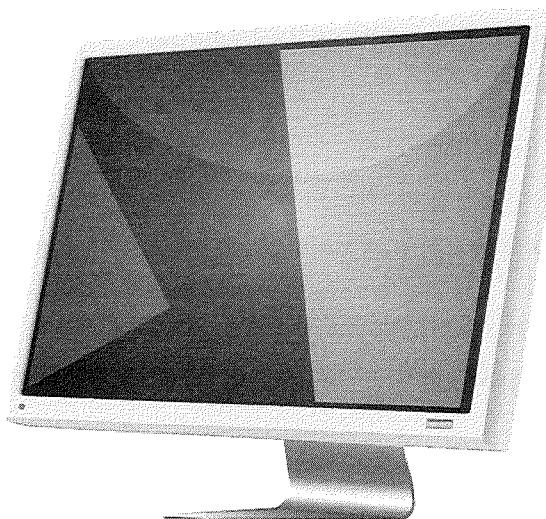
- La primera de ellas consiste en detectar todas las ocurrencias de aquello que queremos conocer.
- La segunda, a diferencia de la anterior, implica la toma de muestras a intervalos regulares de tiempo.

La mayoría de las herramientas que monitorizan la actividad de los sistemas informáticos emplean una mezcla de las dos: se utilizan contadores de sucesos que muestren de forma periódica. Este muestreo periódico permite un análisis estadístico sencillo, toda vez que el volumen de información recogida y su precisión dependerán de la cadencia con que se tomen las muestras.

A continuación vamos a ver el monitor como herramienta de medida y después pasaremos a ver las herramientas de monitorización más comunes en el sistema operativo Windows y Linux.

### Herramienta de medida: el monitor

Un **monitor** es una herramienta diseñada para observar el comportamiento de un sistema informático. Desde el punto de vista de su implementación, los monitores se clasifican en dos grupos principales: hardware y software.



Un monitor hardware es un dispositivo físico, independiente del sistema a examinar, que se conecta al mismo mediante un conjunto de sondas electrónicas. Este tipo de monitores no usan recursos del sistema monitorizado y, por tanto, su grado de intrusión es nulo. Sin embargo, presentan un difícil proceso de instalación, y se enfrentan a dos graves problemas. El primero de ellos es que gran parte de la información relevante del sistema es difícil de conseguir mediante sondas, ya que éstas sólo acceden a aquella información que se refleje en posiciones fijas de memoria. El segundo, quizás más importante incluso, es que el diseño actual de la mayoría de los equipos informáticos, que emplea una escala de integración muy elevada, no facilita la conexión de este tipo de dispositivos de medida.

El uso de programas aporta una gran flexibilidad en el proceso de monitorización, aunque se paga el precio de un grado de intrusión más elevado que en el caso de los monitores hardware. Dado que la toma de medidas por un monitor software implicará la ejecución de un programa, y para ello se usará el procesador de la PC, el simple hecho de monitorizar implicará la perturbación del sistema sobre el que se realiza la medida. El grado de distorsión en la medida suele

estimarse mediante una variable denominada sobrecarga del monitor (overhead), que viene definida de la siguiente manera:

$$\text{Sobrecarga} = \frac{\text{Tiempo de ejecución del monitor}}{\text{intervalo de medida}}$$

De la expresión anterior se deduce fácilmente que la sobrecarga del monitor se puede minimizar haciendo grande el intervalo entre activaciones o disminuyendo su tiempo de ejecución.

### Herramientas de monitorización más comunes en Windows

Las herramientas de monitorización más usadas en Windows para la evaluación del rendimiento son el **Administrador de tareas** y **Performance monitor**.

#### *Administrador de tareas de Windows*

El administrador de tareas de Windows es una de las herramientas de monitorización más populares entre los usuarios de Windows. Viene incluido por defecto en las distintas versiones del sistema operativo. Esta herramienta, aparte de mostrar información del sistema, permite al usuario interactuar con el mismo.

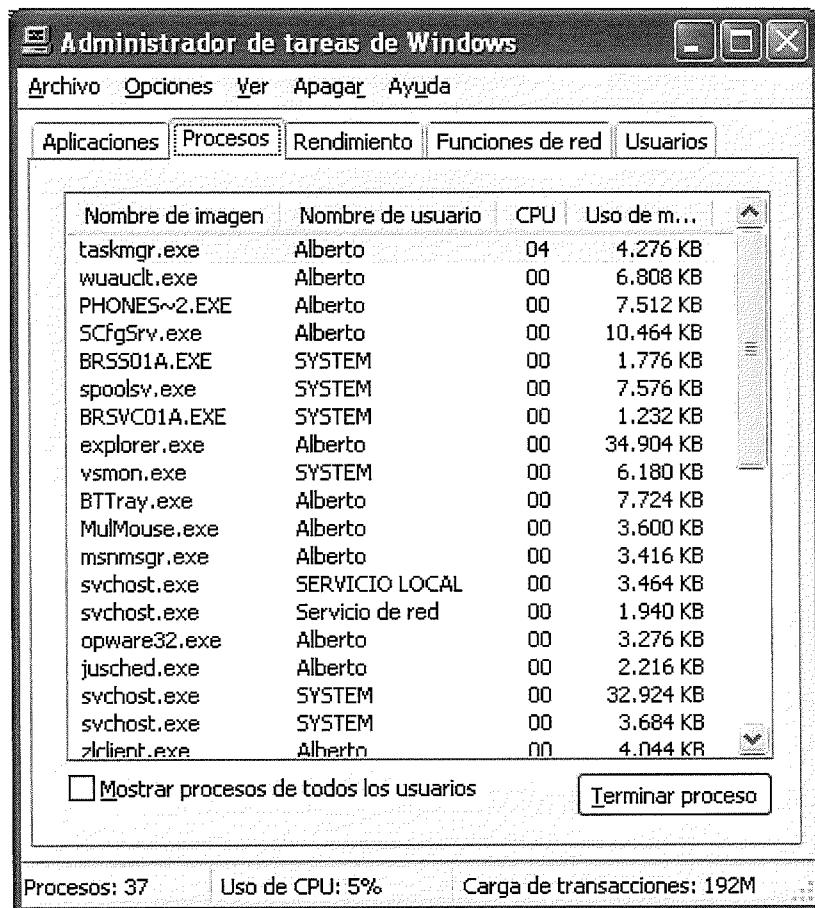
Entre la información que permite monitorizar la herramienta está:

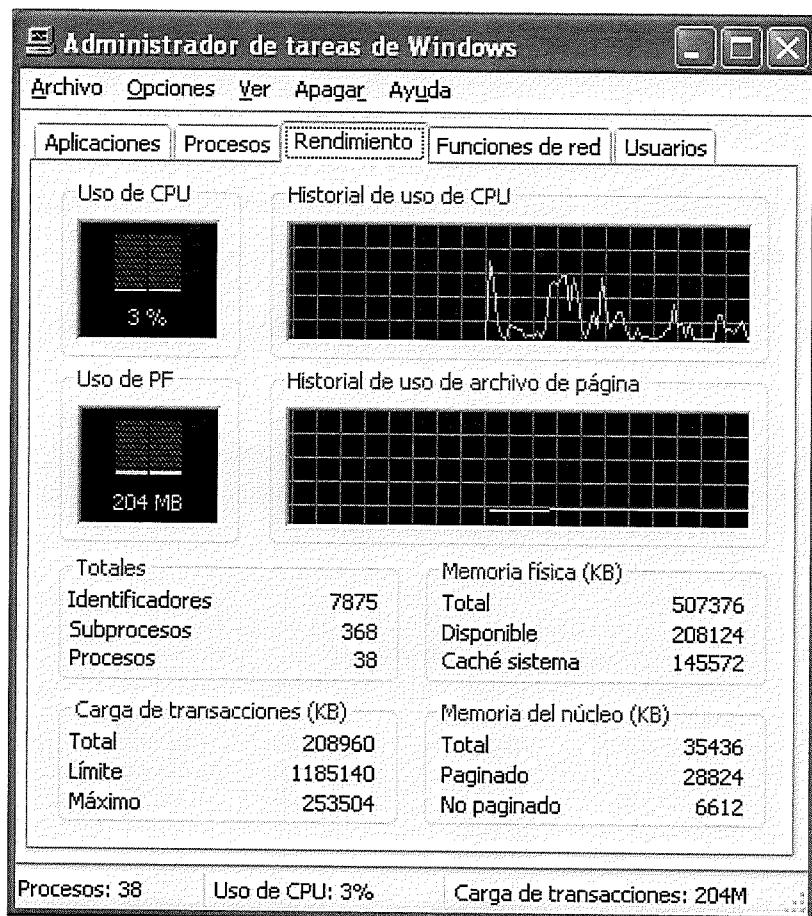
- **Aplicaciones:** muestra las distintas aplicaciones que están ejecutando en el sistema con la posibilidad de terminar dichas aplicaciones o ejecutar otras nuevas.
- **Procesos:** muestra todos los procesos que hay en el sistema con información sobre los mismos: uso de CPU, uso de memoria, identificador de proceso, operaciones de E/S, etc. Además permite terminar cualquier proceso, algo muy útil cuando algún proceso se queda "colgado".
- **Recursos:** permite visualizar información sobre recursos tales como: la CPU, la memoria (tanto física como virtual), el número de procesos totales del sistema, estadísticas sobre las diferentes interfaces de red, etc.

- **Usuarios:** muestra los usuarios que están en el sistema con la posibilidad de desconectar usuarios, cerrar sesiones, enviar mensajes a los demás usuarios, etc.

Otras funcionalidades que permite la herramienta son: apagar, reiniciar, suspender o hibernar el sistema.

En las siguientes figuras se pueden ver dos capturas de pantalla de la herramienta:





### *Performance Monitor*

El Performance Monitor de Windows es la herramienta por excelencia de monitorización en Windows. Aunque no es muy conocida por los "usuarios medios" de Windows, la herramienta es muy potente y permite monitorizar una infinidad de parámetros.

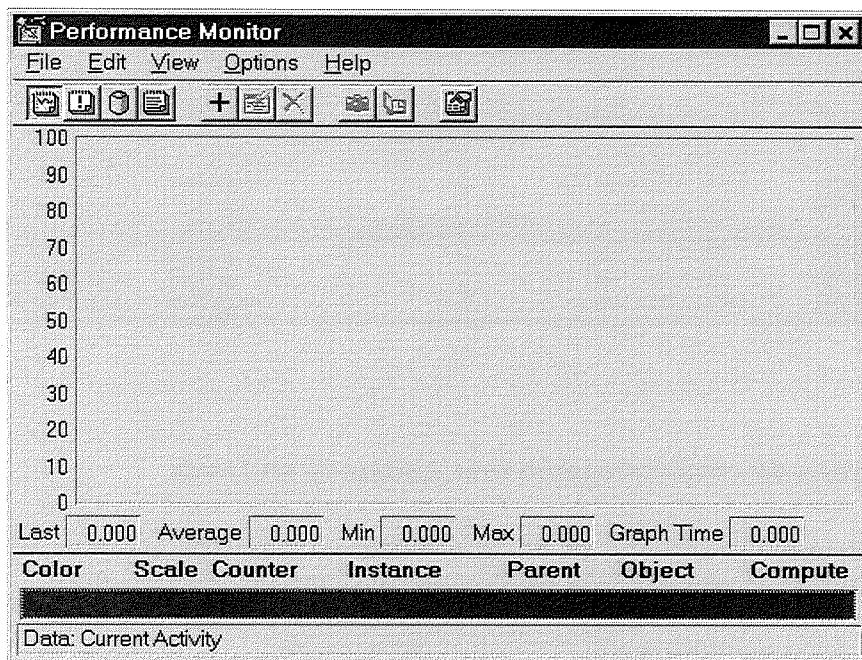
El Performance Monitor organiza la información que se puede monitorizar en una jerarquía cuyos elementos son:

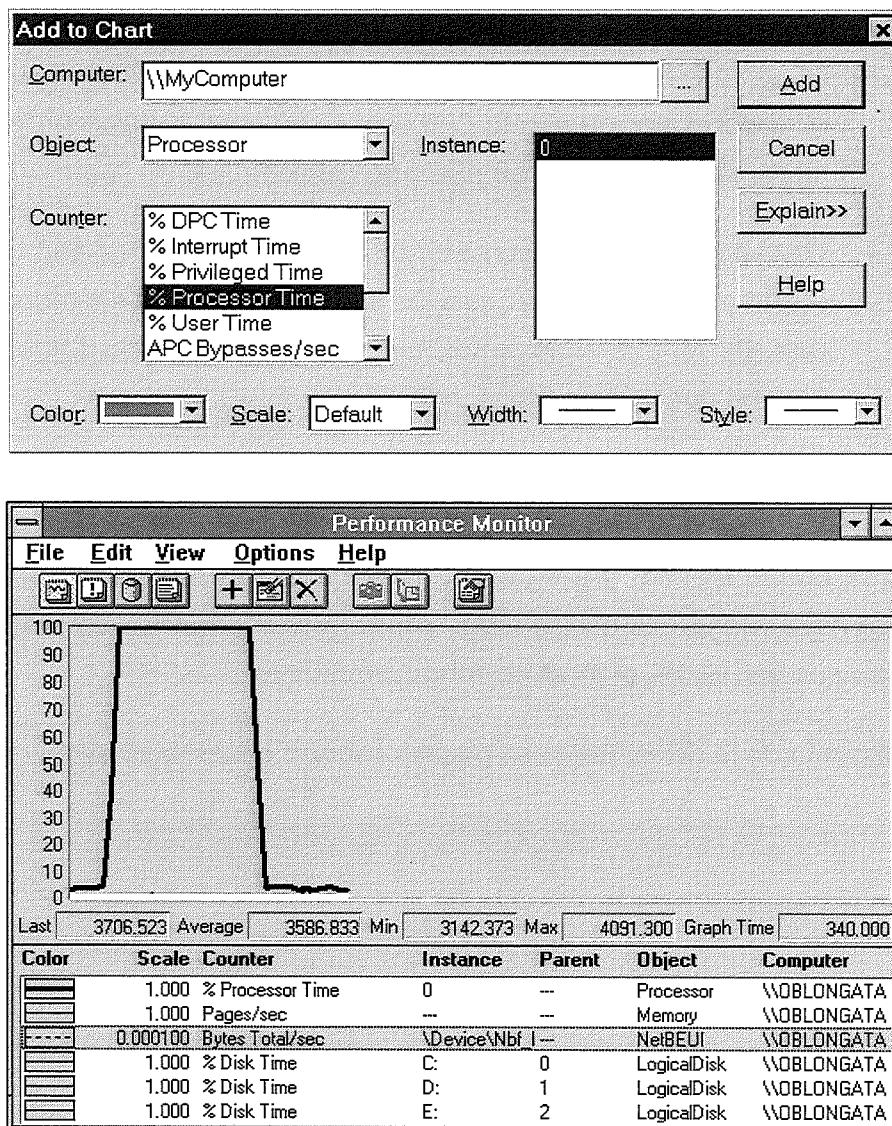
- **Computer:** representa el sistema a monitorizar. Puede ser el propio sistema donde ejecuta la herramienta u otro sistema visible por la red.
- **Object:** es el objeto del cual se desea monitorizar información. En este nivel se encuentra la CPU, la memoria, el disco, etc.

- **Counter:** este nivel contiene los distintos parámetros (contadores) de un determinado objeto que se puede monitorizar. Ejemplos de contadores para el objeto CPU son: tiempo del procesador, tiempo en modo usuario, tiempo en modo privilegiado, etc.
- **Instance:** este nivel contiene las distintas instancias que hay de un objeto determinado del cual se desea mostrar un contador. Para el caso de la CPU tendríamos tantas instancias como CPUs tenga el sistema que se está monitorizando.

La forma de usar el Performance Monitor es: seleccionar el sistema a monitorizar, elegir un objeto dentro del sistema, escoger un parámetro (o contador) del objeto anterior y por último seleccionar la instancia del objeto (en el caso de que haya varias). La herramienta mostrará la información de forma gráfica y permitirá visualizar varias gráficas de forma simultánea.

El aspecto de la herramienta se puede observar en las siguientes figuras:





### Herramientas de monitorización más comunes en Linux

La primera orden que vamos a considerar es `uname` (system information), que aporta información básica sobre el sistema operativo y la máquina. La opción `-a` imprime toda la información disponible (**`uname-a`**).

El sistema operativo Linux define la carga del sistema (system load average) como el número medio de procesos en la cola del núcleo. El valor de la carga se puede ver mediante la orden **`uptime`**.

La medición del tiempo de ejecución de un programa se lleva a cabo con la orden `time`. Esta variable es de especial importancia porque refleja, de forma directa, la percepción que el usuario tiene de las prestaciones del sistema.

informático. Por ejemplo, para saber cuánto tiempo necesitará la ejecución del programa quicksort, ejecutamos la orden: **time quicksort**.

Una de las herramientas más utilizadas por los administradores de sistemas para saber qué procesos hay en ejecución y cuánta memoria consumen es **top** (top CPU processes). Este programa muestra, de manera dinámica, los procesos que están consumiendo tiempo de procesador ordenados de acuerdo con este consumo. La información se actualiza cada 5 segundos, aunque este valor es configurable.

El monitor **vmstat (virtual memory statistics)** muestra información relativa al sistema de memoria, incluyendo datos sobre la memoria física y virtual. Además, ofrece datos relativos a la actividad de intercambio entre memoria y disco (swapping), transferencias con el disco, interrupciones, cambios de contexto y utilización del procesador. La sintaxis de esta orden es vmstat t n, donde t indica el tiempo transcurrido (en segundos, normalmente) entre dos muestras consecutivas, y n es el número de muestras.

El uso del sistema de ficheros (almacenamiento en disco) se puede examinar mediante la orden **df** (filesystem disk usage).

## **1.1. Definición de las funcionalidades que se desea evaluar**

De entre todas las magnitudes medibles de un sistema informático susceptibles de ser utilizadas como índices de prestaciones, el tiempo en llevar a cabo una actividad determinada representa la más intuitiva tanto para el analista como para el profano y, desde el punto de vista de la manipulación matemática, la menos susceptible de incorporar subterfugios.

De hecho, ha sido ampliamente aceptado que el tiempo de ejecución de un programa representa la medida exacta del rendimiento de una PC: aquel que ejecute la misma cantidad de trabajo en el menor tiempo posible será el más rápido.

Incluso teniendo en cuenta lo que acabamos de decir, existen otros índices de rendimiento que, aun presentando inconvenientes, se siguen utilizando, con mayor o menor acierto, en determinadas áreas de la informática. En particular, nos referimos a los índices MIPS, MFLOPS, MHz y CPI, todos ellos empleados con profusión por los diseñadores de procesadores.

La mayoría de los índices clásicos de prestaciones tienen como objetivo la medida del rendimiento del procesador. Por ejemplo, el índice **MIPS (million instructions per second)** para un programa determinado se define como:

$$\text{Número de instrucciones} / \text{Tiempo de ejecución} \times 10^6$$

Otro de los índices clásicos empleado para medir el rendimiento del procesador en el tratamiento de números en coma flotante, especialmente en entornos de grandes computadoras, es el **MFLOPS (million floating point operations per second)**, que computa el rendimiento en aritmética de coma flotante y se define como:

$$\text{Número de operaciones de coma flotante} / \text{Tiempo de ejecución} \times 10^6$$

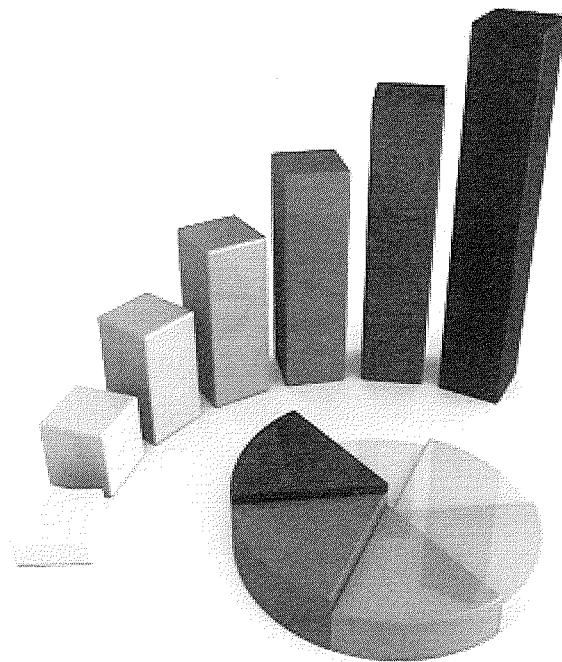
La mayoría de los textos académicos actuales que tratan sobre diseño del procesador tienen en cuenta varios factores que intervienen en el tiempo de ejecución de un programa. En particular, este tiempo se expresa en función del número  $I$  de instrucciones ejecutadas, del número medio de ciclos de reloj necesarios para ejecutar una instrucción (CPI, clock cycles per instruction), y del tiempo de ciclo de reloj **tc**:

$$\text{Tiempo de ejecución} = I \times CPI \times tc$$

En la fórmula anterior intervienen tres factores con implicaciones muy diferentes. La variable  $I$  depende tanto del repertorio de instrucciones como del compilador empleado en traducir el programa a lenguaje máquina. La variable CPI es un valor promedio y depende de la implementación que se haga del juego de instrucciones. Por último, el tiempo de ciclo o periodo de reloj **tc** depende de la frecuencia de reloj a la que funciona el procesador.

## 1.2. Definición de herramientas para evaluar el rendimiento del sistema ERP

La implementación de un sistema de ERP, por lo general, es larga y compleja, ya que implica rediseñar los esquemas de trabajo. Su implementación es de alto riesgo, ya que implica complejidad, tamaño, altos costos, un equipo considerable de desarrollo, además de inversión de tiempo.



En la mayoría de las empresas, se requiere remplazar la infraestructura existente, lo que implica inversión de capital adicional, especialización y hasta la posibilidad de parar el negocio temporalmente para la implementación, por otra parte es importante señalar que el grado de experiencia de los proveedores es un factor importante para el buen funcionamiento del sistema.

Después de la implementación es importante centrarse en el aseguramiento de la calidad y en la mejora del desempeño, para que así el sistema funcione correctamente a largo plazo. También se debe analizar constantemente el retorno de inversión y aspectos clave como la optimización, la cual proporciona ideas que no fueron consideradas durante la implementación como por ejemplo la expansión del software implementado; es importante ver a la optimización como un proceso de mejora continua.

### **1.2.1. Componentes de la implantación de un ERP**

Debido a la complejidad asociada a la implantación de un ERP, es importante considerar todos aquellos aspectos que pueden afectar durante dicha implantación:

#### **1. El ERP**

Existen multitud de ERPs, cada uno de ellos con unas características determinadas. Algunos ERPs sirven para cualquier tipo de organización y para cualquier sector, es lo que se denomina, solución horizontal. También existen ERPs específicos para atender a las necesidades concretas de un sector, es lo que se denomina, solución vertical.

De hecho, hay ERPs que disponen de ambas soluciones: horizontal y vertical. Es importante tener en cuenta que cada organización tienen unas necesidades concretas y que el ERP y su personalización dependerán de estas necesidades, por ello, no existe una implementación "tipo", lo que en una organización funciona puede no ser válido para otra organización.

#### **2. Las personas y la gestión del cambio**

En función de cómo se enfoque, la gestión del cambio permitirá u obstaculizará el proceso de implementación del ERP.

Por ello, el correcto análisis de los requerimientos de los usuarios e integrarlos desde el primer momento de la implantación es clave para conseguir buenos resultados con el proyecto.

Además, se deben definir exactamente las mejoras que va a obtener cada una de las personas de la organización con la implantación y definir un plan de comunicación para "vender" el proyecto a todas las personas de la organización.

Además, es poco habitual que las organizaciones cuenten con personal con una visión tanto de negocio como de tecnología que consiga liderar el proyecto por lo que el trabajo de consultores externos, y en concreto del director de proyecto, es muy importante.

### **3. La estrategia**

El proceso “ideal” sería que el plan tecnológico, incluyendo el ERP y su hardware asociado, soporte la estrategia corporativa y no al contrario.

Básicamente, la idea es que teniendo perfectamente definida la estrategia de la organización, se asocie a ella los recursos tecnológicos necesarios para que sea posible ejecutarla.

### **4. El hardware**

Aunque en principio el hardware no es la parte más compleja de la implantación, en algunos casos puede ocurrir que la mala elección del hardware o diseño del sistema haga disminuir el rendimiento global de la implantación.

En este sentido es básico definir exactamente los requerimientos del sistema y así diseñar la solución de manera que no se invierta ni más ni menos de lo necesario.

### **5. Los procesos**

Se ha de considerar que además de las personas, los procesos son los que definen la eficiencia y eficacia de la organización.

Por ello en el proyecto de implantación del ERP se deben redefinir los procesos para mejorar su eficiencia y eficacia.

El enfoque correcto es redefinir los procesos como un paso previo a la implantación y que los nuevos procesos sean soportados por el ERP.

Sin embargo, lo habitual es encontrar implantaciones de ERPs en los que, tras la implantación, se ejecutan los procesos exactamente igual que antes del ERP. Este es un gran problema ya que no se consigue ninguna mejora en los costes o tiempos de los procesos.

Aunque tengamos el mejor ERP del mundo, si los procesos no se remodelan seguirán siendo igual de eficientes o ineficientes como lo eran hasta el momento de la implantación y entonces, la implantación del ERP tendrá bajo o nulo impacto en la eficacia y eficiencia.

## **6. El resto de aplicaciones de gestión existentes en la organización**

Cada vez es más usual que las organizaciones tengan distintas aplicaciones para la gestión. Entre las aplicaciones más habituales están las herramientas propias o sectoriales, las de Gestión de Relaciones con los Clientes (CRM), Business Intelligence (BI), Gestión de la Cadena de Suministro (SCM), etc.

En la mayoría de las ocasiones, todas las aplicaciones han de estar conectadas con el ERP para conseguir una gestión de la información eficiente. Por ello, la integración entre las distintas aplicaciones (EAI) es una tarea cada vez más compleja y que condiciona los resultados finales de la implantación.

### **1.2.2. Pre-implantación**

La "pre-implantación", es decir, el análisis previo para definir los objetivos del proyecto, alcance funcional, coste total, recursos necesarios, necesidades concretas de la organización, calendarios, etc. Para conseguir evaluar la rentabilidad que supondrá la implantación del ERP.

El proyecto propio de implantación incluyendo desarrollos, personalizaciones, formación etc. Habitualmente la fase de pre-implantación es infravalorada y en muchas ocasiones ni se realiza este análisis llevando a implantaciones con objetivos poco definidos y con multitud de problemas.

Es habitual encontrar organizaciones que no han desarrollado correctamente el análisis pre-implantación y por tanto no han elegido bien la solución. Por todo ello, el análisis previo debe contener al menos los siguientes apartados:

- **Análisis inicial de la estrategia, tecnología, procesos, personas y organización:** en esta fase, se debe realizar un profundo análisis de la estrategia, personas, procesos y tecnología para así plantear la mejor solución tanto desde el punto de vista tecnológico como de gestión del cambio asociado. En esta etapa se crearán equipos de trabajo para hacer este análisis y para el trabajo posterior.

- **Definición de objetivos de la implantación del ERP:** claramente, habrán objetivos tangibles (reducción de costes, mejora de eficacia y eficiencia de procesos, reducción del plazo de entrega, reducción de los niveles de inventario, etc.) y otros intangibles como por ejemplo disponer de más cantidad de información y conocimiento para la toma de decisiones. Obviamente, todos estos objetivos deben estar integrados dentro de la estrategia de la organización.
- **Definición de las mejoras en los procesos y organización que aportará la implantación del ERP:** esto no debe ser una declaración de intenciones sino que se deben haber modelado los procesos de la organización y reconocer el impacto sobre ellos de la implantación del ERP. En esta fase se deben definir objetivos cuantificados de mejora para cada uno de los procesos y deben estar integrados en el calendario del proyecto.
- **Definición del plan de gestión del cambio para conseguir el cambio de manera no traumática:** dentro de este plan, la comunicación interna es muy importante para "vender" los beneficios del proyecto a los integrantes de la organización, y conseguir que todo el mundo perciba una mejora con el proyecto ERP.
- **Elección de la solución tecnológica:** así como el implantador más adecuado en función del análisis realizado en la primera fase así como los módulos y personalizaciones necesarias.
- **Definición de un calendario aproximado y presupuesto asociado:** obviamente esta fase estará directamente relacionada con la fase anterior ya que en función de la elección tecnológica y de los desarrollos anexos, el calendario y el presupuesto variarán.
- **Definición del retorno de la inversión (ROI):** y los parámetros clave KPI, para definir el seguimiento de la implantación así como un análisis de sensibilidad ante la variación de determinados parámetros.
- **Implantación del ERP:** seguimiento y control estricto de los objetivos previamente definidos así como de los elementos críticos para la rentabilidad del proyecto. Es muy importante que haya un estricto control del proyecto para que se cumplan los objetivos definidos en las primeras etapas.

### **1.2.3. Elección del ERP: Parámetros**

Para la elección del ERP se emplearán al menos los siguientes parámetros:

- Capacidad de adaptación a las necesidades del cliente.
- Cantidad de módulos adaptables a las necesidades.
- Facilidad de uso.
- Coste de la solución.
- Experiencias y casos de éxito en el sector.
- Solidez financiera del vendedor.
- Tecnologías empleadas.
- Estabilidad en las tecnologías empleadas.
- Cantidad y perfil de clientes.
- Robustez tecnológica de la solución.
- Inversión en I+D.
- Metodología de implantación.
- Independencia de sistema operativo y de motor de base de datos.
- Usabilidad.
- Escalabilidad.
- Flexibilidad para la gestión de nuevas líneas de negocio.

Para la evaluación del implantador, se emplearán al menos los siguientes parámetros:

- Experiencia en el producto y en el sector de la compañía.
- Coste.
- Conocimientos y experiencia del personal, sobre todo del líder de proyecto en implantaciones del producto y en el sector de la compañía.
- Metodología de implantación.
- Metodología de formación.
- Proximidad geográfica.
- Presencia global.

- Compromiso en la implantación.
- Conocimientos y experiencia en integración de sistemas.
- Capacidad de disposición de personal.
- Estabilidad financiera del implantador.

Después de la implementación del ERP, si el rendimiento de la empresa no aumenta o alcanza el punto de referencia, uno tiene que analizar el proceso de negocio para identificar los inconvenientes.



## RECUERDA

- Para evaluar el rendimiento se necesita ejecutar un determinado software (distintas aplicaciones) y ver cuál es la utilización del hardware. Esta actuación nos aportará la información necesaria para la realización de modificaciones que mejoren las prestaciones, como es la selección del hardware, aplicaciones etc.
- Existen una serie de **pruebas** que nos van a permitir estimar el rendimiento del sistema, ya sea antes o durante su explotación, si tenemos en cuenta que las necesidades software o hardware pueden variar a lo largo del tiempo.
- La **monitorización** es una técnica de uso generalizado para supervisar, analizar y evaluar el comportamiento y rendimiento de los sistemas informáticos que están en funcionamiento.
- Un **monitor** es una herramienta diseñada para observar el comportamiento de un sistema informático. Desde el punto de vista de su implementación, los monitores se clasifican en dos grupos principales: hardware y software.
- Las herramientas de monitorización más usadas en Windows para la evaluación del rendimiento son el **Administrador de tareas** y **Performance monitor**.
- El **Administrador de tareas** de Windows es una de las herramientas de monitorización más populares entre los usuarios de Windows. Viene incluido por defecto en las distintas versiones del sistema operativo. Esta herramienta, aparte de mostrar información del sistema, permite al usuario interactuar con el mismo.
- El **Performance Monitor** de Windows es la herramienta por excelencia de monitorización en Windows. Aunque no es muy conocida por los "usuarios medios" de Windows, la herramienta es muy potente y permite monitorizar una infinidad de parámetros.
- De entre todas las magnitudes medibles de un sistema informático susceptibles de ser utilizadas como índices de prestaciones, el tiempo en llevar a cabo un actividad determinada representa la más intuitiva tanto para el analista como para el profano y, desde el punto de vista de la manipulación matemática, la menos susceptible de incorporar

subterfugios.

- La **implementación** de un sistema de ERP, por lo general, es larga y compleja, ya que implica rediseñar los esquemas de trabajo. Su implementación es de alto riesgo, ya que implica complejidad, tamaño, altos costos, un equipo considerable de desarrollo, además de inversión de tiempo.

## Preguntas de Autoevaluación

### 1. Las pruebas benchmarks:

- a) Nos ayudan a visualizar el empleo de los recursos del sistema con la particularidad de que los datos obtenidos varían en el tiempo según el uso que se haga en función de su actividad o programas que se estén ejecutando.
- b) Recogen información comparativa entre distintos sistemas.
- c) Es una medida de evaluación que simula a un sistema donde existe un elemento que aún no se ha instalado.

### 2. Confirma si la siguiente afirmación es verdadera o falsa:

*"La monitorización es una técnica de uso generalizado para supervisar, analizar y evaluar el comportamiento y rendimiento de los sistemas informáticos que están en funcionamiento".*

- a) Verdadero.
- b) Falso.

### 3. Las herramientas de monitorización más usadas en Windows para la evaluación del rendimiento son: (respuesta múltiple)

- a) Administrador de tareas.
- b) Panel de control.
- c) Performance monitor.

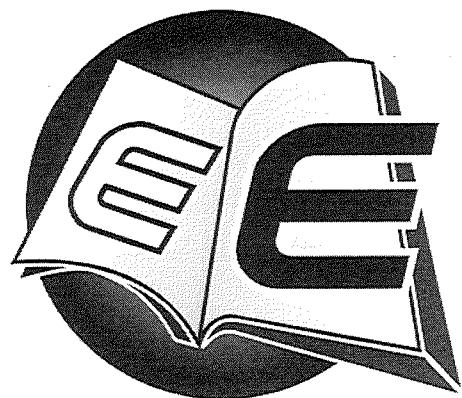
**4. El sistema operativo Linux define la carga del sistema (system load average) como el número medio de procesos en la cola del núcleo. El valor de la carga se puede ver mediante la orden:**

- a) top.
- b) uptime.
- c) uname-a.

**5. La mayoría de los índices clásicos de prestaciones tienen como objetivo la medida del rendimiento del procesador. Por ejemplo, el índice MIPS (million instructions per second) para un programa determinado se define como:**

- a) Número de instrucciones/tiempo de ejecución  $\times 10^6$ .
- b) Número de operaciones de coma flotante/tiempo de ejecución  $\times 10^6$ .
- c) Número de instrucciones – tiempo de ejecución  $\times 6^{10}$ .

# UD3 Resolución de problemas en el almacén de datos



**UF1886 Administración  
del Gestor de Datos en  
Sistemas ERP-CRM**



## 1. Trazas del sistema (logs)

La palabra **log** es un término anglosajón, equivalente a la palabra bitácora en español. Sin embargo, se utiliza en los países de habla hispana como un anglicismo derivado de las traducciones del inglés en la jerga informática. Del mismo término también proviene la palabra blog, que es la contracción de "web log".

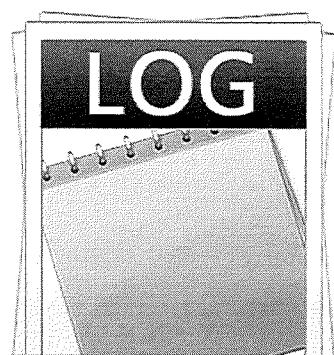
Un log es un registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué (who, what, when, where y why) un evento ocurre para un dispositivo en particular o aplicación.

La mayoría de los logs son almacenados o desplegados en el formato estándar, el cual es un conjunto de caracteres para dispositivos comunes y aplicaciones. De esta forma cada log generado por un dispositivo en particular puede ser leído y desplegado en otro diferente.

También se le considera como aquel mensaje que genera el programador de un sistema operativo, alguna aplicación o algún proceso, en virtud del cual se muestra un evento del sistema.

A su vez la palabra log se relaciona con el término evidencia digital. Un tipo de evidencia física construida de campos magnéticos y pulsos electrónicos que pueden ser recolectados y analizados con herramientas y técnicas especiales, lo que implica la lectura del log y deja al descubierto la actividad registrada en el mismo.

Para poder visualizar el contenido de estos ficheros, tan sólo necesitamos un editor de texto, como puede ser vi, nano, gedit, kwrite....



### **Documentación de procesos realizados**

Los LOG's, permiten detectar incidentes y comportamientos no habituales. Esta información es muy útil para localizar fallos en las configuraciones de los programas o cambios que se hayan hecho sobre dicha configuración. También detectan si se ha desconectado al dispositivo del sistema y controlar el uso de recursos por parte del usuario. Además de proporcionar información sobre el rendimiento del sistema, pueden resultar útiles para detectar cuándo un equipo está empezando a fallar.

La administración de usuarios, grupos y sus privilegios y también la de ficheros y directorios, se ha hecho correctamente. Hay que tener en cuenta las ACL (access control list) o listas de control de acceso. Mediante las ACL es posible la asignación de permisos a usuarios o grupos concretos. Esto puede ser útil en caso de que dos usuarios que pertenezcan a grupos diferentes necesiten los mismos permisos a la hora de acceder a unos determinados directorios. Por ejemplo, un proyecto interdisciplinario entre profesores del departamento de Informática y el de Filosofía (los dos grupos de usuarios con perfiles perfectamente definidos dentro de cada departamento), podría requerir que los componentes del proyecto tuvieran que acceder a los mismos directorios, necesidad que se podría satisfacer con la creación de la ACL correspondiente.

En informática, el concepto de historial o de logging designa la grabación secuencial en un archivo o en una base de datos de todos los acontecimientos que afectan un proceso particular (aplicación, actividad de una red informática...). El término (en inglés log file o simplemente log) designa al archivo que contiene estas grabaciones. Generalmente fechadas y clasificadas por orden cronológico, estos últimos permiten analizar paso a paso la actividad interna del proceso y sus interacciones con su medio.

### **Verificación de que los procesos se han realizado**

Los Sistemas Operativos incorporan LOG's donde registran información sobre que usuarios y en qué momento abren y cierran sesión, que procesos están en ejecución dentro del sistema, las aplicaciones que son ejecutadas por los usuarios, y así como los posibles problemas de seguridad.

La grabación de un historial permite poner en marcha funciones tales como los «últimos archivos abiertos», los «últimos comandos tipeados» o las «últimas páginas web consultadas».

La **periodización aplicativa** designa la grabación cronológica de operaciones durante el funcionamiento de la aplicación. Una periodización aplicativa es en sí misma, una exigencia del trabajo. Está definida como una función que forma parte de la lógica aplicativa. Por lo tanto, no debería estar detenida durante el funcionamiento de la aplicación.

La **periodización sistema** designa la grabación cronológica de acontecimientos que sobrevienen a nivel de componentes del sistema. El nivel de esta periodización puede ser medido, con el fin de filtrar los diferentes acontecimientos, según su categoría de gravedad. Las categorías generalmente utilizadas son: información, depuración, advertencia, error. Por ejemplo los sistemas Unix ponen en marcha esta periodización sistema con la ayuda del protocolo Syslog.

### Características y tipos

No existe sólo un fichero .log, porque sino sería interminable e inmanejable, para ello están divididos en distintos ficheros y cada uno con su función específica. Por ejemplo:

- **syslog**: se encarga de registrar los mensajes de seguridad del sistema.
- **kern**: se encarga de los mensajes del núcleo (kernel).
- **messages**: archiva los distintos mensajes generales que nos manda el sistema.
- **debug**: mensajes de depuración de los programas.
- **user.log**: información sobre el usuario.
- **Xorg.0.log**: guarda información sobre el entorno gráfico.
- **auth.log**: contiene los accesos al sistema, incluidos los intentos fallidos.

## 1.1. Definición de sistemas para trazar los procesos entre los sistemas ERP, CRM y almacén de datos

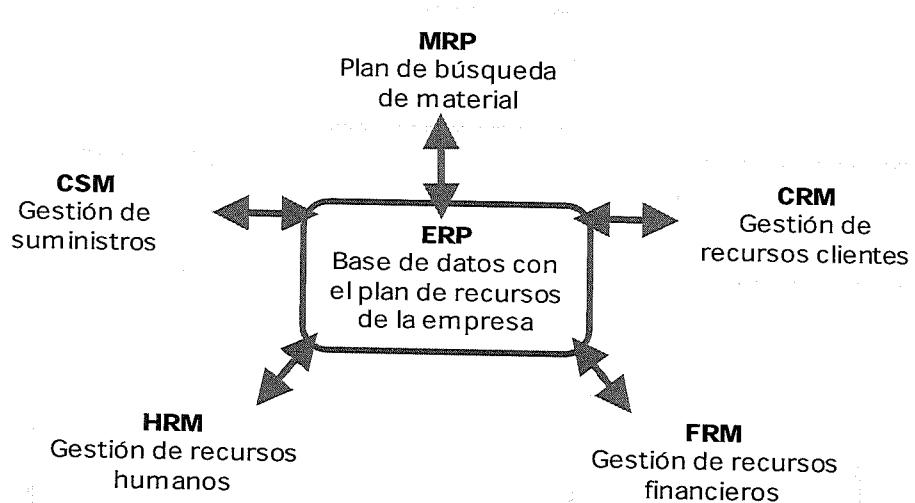
### 1.1.1. Sistemas ERP

Cuando hablamos de **sistema de gestión empresarial ERP** (*Enterprise Resource Planning*) nos referimos a un sistema de gestión integral, que se ha diseñado para facilitar las tareas rutinarias de la empresa. Mediante el ERP conseguimos automatizar estos procesos y al mismo tiempo facilitamos la utilización de todos los recursos disponibles en la empresa.

En el sistema de gestión ERP se utiliza una arquitectura modular en la cual cada modulo representa un área diferente dentro de la empresa (área de finanzas, comercial, contabilidad, logística, stock, producción, etc.), mediante este sistema interconectado que trabaja en dos direcciones podemos unificar y ordenar todos los movimientos de la empresa en una sola base de datos.

Por lo tanto el sistema ERP nos permite eliminar posibles barreras entre los distintos departamentos, permitiendo que la información fluya dentro de la empresa y facilitando la toma de decisiones, gracias a que tenemos acceso a todos los datos que necesitamos en poco tiempo.

Los ERP evolucionaron a partir de los MRP que estaban diseñados únicamente para la gestión de materiales y el control de la productividad. Como se puede ver en el esquema son el núcleo de otras aplicaciones útiles para la empresa como el CRM (Gestión de las relaciones con los clientes).



La competencia en un mundo cada vez más globalizado hace que las empresas quieran ser cada vez mejores y los sistemas de gestión ERP ayudan a conseguir esos objetivos, pero antes de lanzarse a cambiar el funcionamiento de una empresa es bueno conocer tanto las ventajas como las desventajas de estos planes de gestión.

La **principal ventaja** es aumentar la productividad ya que mediante este sistema se tiene acceso a la información de un modo sencillo y el tiempo se invierte en analizar las situaciones y buscar las soluciones.

Entre las ventajas del sistema ERP encontramos:

- Reducción de inventarios.
- Incremento en ventas por tiempote respuesta a clientes.
- Disminución de compras.
- Almacenar información de todos los procesos, para analizarla posteriormente.
- Seguimiento del cliente desde la aceptación hasta la satisfacción completa.
- Incrementar la producción manteniendo los costes fijos.

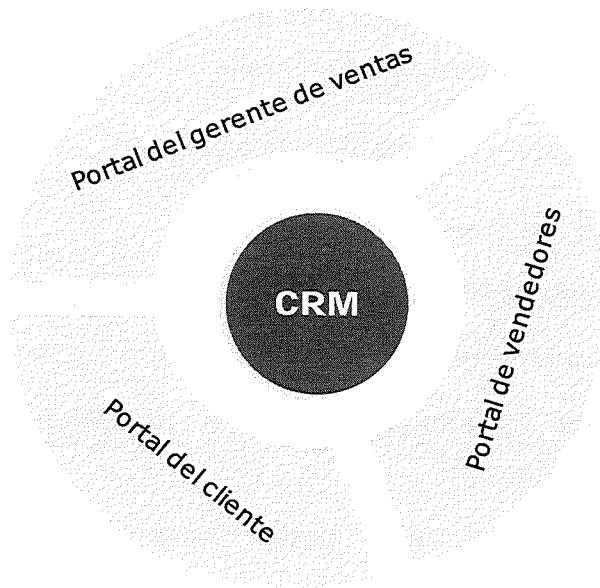
Así, también cuenta con una serie de desventajas como las siguientes:

- Son sistemas complejos y algunas compañías no saben ajustarse a ellos.
- Hay pocos expertos en y su implantación en la empresa resulta cara ERPs.
- Es necesario tener empleados capaces de controlar el sistema ERP.
- Cuando se establece el sistema, realizar cambios resulta muy costoso en tiempo y dinero.

### 1.1.2. Sistema de Gestión de clientes CRM

Con el **sistema de gestión de clientes CRM** (Customer Relationship Management), vamos a lograr tener en un solo sitio todas las relaciones que tenemos con nuestros clientes y que se refieren a sus necesidades, productos y servicios que demandan de nuestra empresa.

Mediante esta gestión avanzada de los clientes seremos capaces de recopilar la máxima información sobre ellos, adelantándonos de esta forma a sus necesidades y aumentando el valor de nuestros servicios. El sistema de gestión de clientes CRM se incluye también como un módulo que forma parte del sistema ERP, que como ya explicamos nos permite tener bajo control todos los departamentos de la empresa y tomar decisiones en poco tiempo.



El módulo CRM dentro del sistema de gestión ERP es de vital importancia si queremos dar el mejor servicio a nuestros clientes, hemos de tener en cuenta que la buena opinión de un cliente puede conseguirnos más ventas. Por esa razón es tan importante tener controladas las necesidades de nuestros clientes mediante módulos o aplicaciones informáticas que nos permitan dar un mejor servicio.

Con el fin de controlar este aspecto tan importante de la empresa han aparecido servicios en la red como el módulo CRM de Buzón, que nos permite controlar (cuántos enviamos, los que ha leído el receptor, cuántos lo han borrado, etc.) y revisar todos los documentos que hemos enviado a nuestros clientes.

En el módulo de gestión de la relación con el cliente o CRM se incluyen herramientas para llevar la gestión y el análisis de las campañas de marketing, para estrechar vínculos con los clientes y para garantizar un servicio posventa óptimo.

Mediante la interconexión al correo electrónico conseguimos informar de forma automática a nuestros clientes cuando hay cambios en los precios, promociones o nuevos productos. Otra fórmula que da resultado es la Web, mediante la cual podemos recibir las distintas peticiones de los clientes y almacenarlas mediante el sistema de gestión ERP, mediante el cual se buscará la respuesta adecuada a cada petición.

### 1.1.3. Almacén de datos o Data Warehouse

Un **almacén de datos** o Data Warehouse es básicamente una base de datos que integra datos procedentes de uno o varios sistemas de información de una compañía, y que está orientado a la toma de decisión.

La definición más generalizada y aceptada de un Data Warehouse es la que propuso Inmon (1996):

*"Un DW, es una colección de datos orientados al tema, integrados, no volátiles e históricos, cuyo objetivo es el de servir de apoyo en el proceso de toma de decisiones gerenciales".*

Un Data Warehouse representa el motor de los sistemas de información de la década de los noventa, ya que permite integrar datos, organizados y almacenarlos para su posterior tratamiento analítico, y siempre bajo una amplia perspectiva temporal. Se trata pues, de una tecnología que transforma los datos operacionales de una organización en una herramienta competitiva, ya que los usuarios finales pueden examinar los datos de un modo más estratégico, así como lograr alcanzar el conocimiento oculto en los enormes volúmenes de datos almacenados en el DW. Es tal la capacidad de almacenamiento en un DW, que hablar de gigabytes, o de terabytes, resulta de lo más habitual en los DW actuales. No obstante, existen ya algunas grandes compañías comerciales que empiezan a trabajar con volúmenes de datos del orden de los petabytes.

En la siguiente tabla que se expone a continuación vemos las diferencias que existen entre un DW y una base de datos tradicional.

Base de datos tradicional	Data Warehouse
Predomina la actualización	Predomina la consulta
La actividad más importante es de tipo operativo (día a día)	La actividad más importantes es el análisis y la decisión estratégica
Predomina el proceso puntual	Predomina el proceso masivo
Mayor importancia a la estabilidad	Mayor importancia al dinamismo
Datos en general desagregados	Datos en distintos niveles de detalle y agregación
Importancia del dato actual	Importancia del dato histórico
Importancia del tiempo de respuesta de la transacción instantánea	Importancia de la respuesta masiva
Estructura relacional	Visión multidimensional
Usuarios de perfiles medios o bajos	Usuarios de perfiles altos
Eplotación de la información relacionada con la operativa de cada aplicación	Eplotación de toda la información interna y externa de todo el negocio

Las bases de datos relacionales no representan una adecuada solución como herramienta para el soporte directo de sistemas de información para la toma de decisiones. El análisis de la información en un sistema DW parte de la visión de la información como dimensiones del negocio (zonas geográficas, productos, tiempo). A su vez, cada dimensión se organiza en diferentes jerarquías.

### Características del Data Warehouse

Inmon (1996) definió las siguientes características:

- **Temáticas:** los datos se organizan por temas, en contraposición con los tradicionales sistemas que se organizaban generalmente por procesos funcionales, nóminas, recursos humanos, etc. Es decir, para una universidad, los temas o sujetos pueden ser estudiantes, clases y profesores. Para una empresa, los clientes, proveedores, empleados, etc. Esta forma de agrupar los datos, permite a los responsables de las distintas áreas, disponer de los datos que consideren más oportunos para definir sus estrategias.

- **Integrados:** se trata de una base de datos única, de uso global para todos los departamentos de la empresa. Esta visión única y transversal se encuentra en contraposición con los sistemas transaccionales, que satisfacen necesidades de información en sentido vertical.
- **No volátiles:** es una característica derivada de la anterior, de forma que una vez cargado un dato en la DW, debe mantenerse invariable. En un sistema operacional, los datos se actualizan regularmente, mediante procesos que consisten en insertar, borrar o modificar registros. En el DW, sólo se realizan dos tipos de operaciones: la carga inicial de datos y el acceso a los mismos. No existe el proceso de actualización de datos, sólo de carga.
- **Históricas:** los datos no se actualizan, sino que se almacena el historial de los datos, es decir, el conjunto de valores que el dato ha tenido a lo largo de su historia, asociando a cada dato una referencia de tiempo, con el fin de poder identificar los distintos valores que dicho dato ha ido tomando a lo largo de su ciclo de vida. En los sistemas transaccionales, los datos representan la situación actual a la que se refieren, por ejemplo la dirección actual de un cliente, su estado civil, número de hijos, profesión, etc. En un DW, los datos guardan información de un período que oscila entre los 5 y 10 años, (en contraposición con el horizonte temporal de las bases de datos operacionales, que oscila entre 60 y 90 días). Durante ese período, los clientes pueden cambiar de dirección, casarse, tener hijos, cambiar de trabajo, etc. Si se desea, la empresa puede analizar la evolución, cambios y tendencias de sus clientes.

### **Etapas en la elaboración del Data Warehouse**

Martín y Ureña (2000), definen tres etapas en el proceso de elaboración de un DW.

#### *Primera etapa*

Carga de datos desde fuentes internas de la organización o bien de fuentes externas, estos últimos datos permiten validar o completar los datos existentes en el interior de la empresa. La carga de datos puede ser manual, a través de programas diseñados para este fin o por medio de herramientas específicas. En

esta etapa se llevan a cabo complejos procesos de transformación, limpieza, depuración, filtrado e integración de datos.

Es importante tener en cuenta que los datos del DW no son los mismos que los de las bases de datos operacionales. En el proceso de carga de datos al DW, éstos sufren una profunda transformación, y el resultado es un DW que contiene un resumen de la información que no se encuentra en el ambiente operacional, con un grado de redundancia en los datos entre los dos depósitos de datos (el operacional y el DW), inferior al 1%.

#### *Segunda etapa: almacenamiento de datos y creación de los meta datos*

Éstos últimos forman parte del DW, ya que contienen la semántica de los datos, es decir, su significado actual y sus distintas definiciones a lo largo del ciclo de vida del DW. También son necesarios los metadatos porque explican las interrelaciones entre los datos, necesarias para conocer las reglas del negocio.

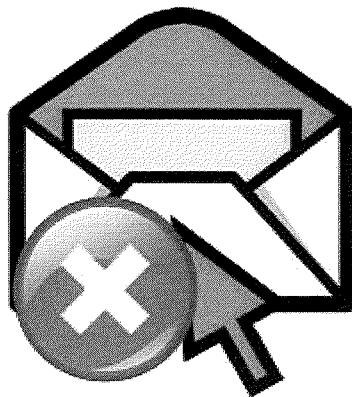
#### *Tercera etapa*

Explotación de los datos por parte del usuario, con el fin de obtener información útil y necesaria para la toma de decisiones. La explotación de los datos puede realizarse con distintas herramientas, que van desde no muy complejas hojas de cálculo, herramientas de consulta, de visualización, etc., hasta la utilización de procesos analíticos en línea, utilizando herramientas de tipo OLAP (On Line Analytical Processing). Las herramientas OLAP más potentes incluyen sistemas llamados de minería de datos o Data Mining, los cuales representan procesos que se basan en la aplicación de técnicas (semi-) automáticas para la obtención de información no representada explícitamente en los datos y de mayor utilidad estratégica que éstos.

## **2. Incidencias: identificación y resolución**

Generalmente los errores de bases de datos, también llamados errores de motor, ocurren cuando se dan errores en tiempo de ejecución en los registros del código de eventos. Por ejemplo, un error de base de datos ocurre cuando el usuario intenta almacenar un valor nulo en un campo que no admite valores nulos.

Cuando ocurre un error de base de datos, el motor de base de datos subyacente que detecta el error envía normalmente un mensaje de error. Sin embargo, la naturaleza exacta del mensaje de error depende de la base de datos a la que se tenga acceso, por ejemplo, los mensajes de error producidos por un servidor de base de datos remoto (como Microsoft SQL Server) probablemente serán distintos de los que se producen si ocurre un error en una tabla local de Visual FoxPro.



Además, los errores del motor son a veces muy genéricos, porque el motor de base de datos no tiene información sobre el contexto en el que se actualiza un registro. Como consecuencia, los mensajes de error producidos por un motor de base de datos suelen ser menos útiles para el usuario de una aplicación de Visual FoxPro.

Para controlar errores de base de datos de forma más específica para la aplicación, puede crear desencadenadores con el comando CREATE TRIGGER. El desencadenador se llama cuando se intenta la actualización de un registro (eliminar, insertar o actualizar). El código de desencadenador personalizado puede buscar entonces condiciones de error específicas de la aplicación e informar de ellas.

Si controla errores de base de datos mediante desencadenadores, debería activar el búfer. De esta forma, cuando se actualiza un registro, se llama al desencadenador pero el registro no se envía automáticamente a la base de datos subyacente. Así evita la posibilidad de producir dos mensajes de error: uno del desencadenador y otro del motor de base de datos subyacente.

## 2.1. Definición de procesos para identificar incidencias, control y resolución

El uso de bases de datos podría crear algunos problemas, especialmente desde el punto de vista del consumidor, como por ejemplo:

- **Falta de seguridad:** es muy difícil garantizar en la actualidad, completa seguridad en el manejo de la información que recopilan las empresas y es difícil estar 100% seguro de que los datos entregados por el consumidor serán utilizados únicamente para los fines en que se entregaron dichos datos.
- **Confiabilidad:** no siempre los datos recopilados son totalmente confiables, muchas veces los usuarios por no comprometer su integridad, utilizan datos inexactos y no son del todo sinceros. Éste es un problema típico en los Bancos, en donde en algunas ocasiones se brinda información inexacta acerca de niveles de ingresos, deudas etc.
- **Ética:** un problema serio es el manejo de la información por parte de los administradores de la base de datos, a veces, las preferencias por productos individuales y la información inexacta lleva al consumidor a elegir productos de mala calidad y con especificaciones que no son claras.

Con respecto a lo contenido en una base de datos, no existe un número de elementos mínimamente necesarios. Cada empresa debe realizar una auditoria rigurosa de sus necesidades de información en función de sus objetivos.

La mejor base de datos es aquella que por sus contenidos puede aparentar más complejidad o sofisticación en la información que proporciona. Frecuentemente, una base de datos de aparente simplicidad, pero con contenidos perfectamente adaptados los objetivos relacionales, cumplirá su papel de apoyar de forma discreta el conjunto de decisiones a tomar en la estrategia relacional.

No obstante, para lograr lo anterior, personalizar el marketing y construir relaciones estrechas con los clientes, es necesario apoyarse en tecnologías de información, en sistemas que están diseñadas para manejar grandes volúmenes de datos y administrar la información a través del proceso de negocios.

Una de las tecnologías que satisface esta necesidad es el CRM, un software que provee aplicaciones que integran marketing, ventas, e-commerce (comercio electrónico) y servicios de soporte al cliente para la empresa. El CRM es en sí una estrategia de negocios que está plenamente orientado al cliente.

Ahora bien, el CRM no es un nuevo concepto de marketing, aunque está basado en tres aspectos de su administración:

- Orientado al cliente.
- Marketing relacional.
- Base de datos de marketing.

A continuación se ofrecen las soluciones propuestas a los problemas que podría encontrar al trabajar con servidores de base de datos o geodatabases de servidores de base de datos, concretamente con ArcGIS Resource Center, Desktop 10.

**Error: no se puede conectar a este servidor de datos...SQL Server no existe o acceso denegado**

Las posibles razones por las que recibe el mensaje de error:

- Si no se ha iniciado la instancia de SQL Server Express, el cliente remoto no podrá localizarla.
- Si su cuenta de usuario no se ha configurado nunca en el servidor de base de datos o la cuenta de usuario se ha eliminado del servidor de base de datos, no podrá realizar consultas o conectarse al mismo.
- Los clientes remotos deben tener una manera de comunicarse con la instancia de SQL Server Express. Si el protocolo de red no está establecido, dicho método de comunicación no está establecido y no hay ninguna manera de realizar una conexión a la instancia a través de una red.
- El servicio SQL Server Browser proporciona información acerca de las instancias de SQL Server instaladas en el equipo cuando se recibe una solicitud de un cliente de SQL Server. Si el SQL Browser está desactivado, el cliente no recibirá los nombres y puertos utilizados para cada instancia de SQL Server instalada y, por consiguiente, no puede buscar la instancia.

*Solución:* en primer lugar, asegúrese de que se ha iniciado el servidor de base de datos al que está intentando conectarse. Si lo ha hecho, confirme que dispone de permisos para acceder remotamente a este servidor de base de datos. Si el servidor de base de datos se ha iniciado y tiene derechos para conectarse al mismo y realizar consultas, los protocolos de red probablemente no se han habilitado en el servidor de base de datos. Para volver a habilitar los protocolos de red, haga lo siguiente:

- Abra el administrador de configuración de Microsoft SQL Server. Se puede acceder al mismo desde Inicio > Todos los programas > Microsoft SQL Server > Herramientas de configuración.
- Expanda Configuración de red de SQL Server en el árbol y haga clic en los protocolos de su instancia de SQL Server.
- Haga clic con el botón derecho en el protocolo TCP/IP y haga clic en Habilitar.

Detenga y reinicie el servicio. Si los protocolos de red están habilitados y sigue sin poder conectarse al servidor de base de datos desde el cliente remoto, el servicio SQL Server Browser probablemente esté detenido. Para iniciar el servicio SQL Server Browser:

- Abra el administrador de configuración de Microsoft SQL Server.
- Haga clic en el ícono SQL Server para ver el servicio SQL Server Browser.
- Si el estado del SQL Server Browser está en pausa, haga clic con el botón derecho y haga clic en Reanudar. Si el estado de SQL Server Browser está detenido, haga clic con el botón derecho en el mismo y haga clic en Iniciar.

Si están deshabilitadas todas las opciones del menú contextual excepto Propiedades y Ayuda al hacer clic con el botón derecho en SQL Server Browser, el modo de inicio de SQL Server Browser se ha establecido en Desactivado. Para cambiarlo:

- Hacer clic con el botón derecho en SQL Server Browser y haga clic en Propiedades.
- Haga clic en la ficha Servicio.
- Haga clic en Modo de inicio.

- En el menú desplegable, seleccione Automático o Manual. Al establecer el modo de inicio en Automático el servicio SQL Server Browser se inicia automáticamente al reiniciar el equipo.
- Haga clic en Aceptar.

Ahora al hacer clic con el botón derecho en el servicio SQL Server Browser, debería estar disponible la opción Iniciar.

#### **Error: no se puede generar el contexto SSPI**

La razón posible por la que ha recibido el mensaje de error:

- Este error es más común si está trabajando en un equipo portátil y se ha registrado con las credenciales de dominio, pero no está conectado al dominio realmente. Esto podría ocurrir si está conectado en un sitio del cliente (en contraposición a su propia red) o utilizando su equipo portátil mientras no está conectado a ninguna red, por ejemplo trabajando en la obra. Están relacionados con el protocolo de red TCP/IP y la autenticación de Windows. También podría consultar el sitio web de Microsoft para acceder a un artículo de la base de conocimiento sobre cómo solucionar el mensaje de error "No se puede generar el contexto SSPI".

*Solución/solución alternativa:* realice su conexión al servidor de base de datos en el equipo local (Aregar servidor de base de datos) utilizando el bucle invertido de TCP/IP de 127.0.0.1 en lugar del nombre de dominio; por ejemplo: 127.0.0.1\sqlexpress.

Esto funcionará siempre que el servidor de base de datos (la instancia de SQL Server Express) esté instalada en su equipo local.

**Error de DBMS subyacente Error al conectar a la base de datos (Proveedor de base de datos de OLE de Microsoft para SQL Server). Se produjo un error en el inicio de sesión para el usuario '<nombre\_de\_usuario>'. El usuario no está asociado a una conexión de SQL Server de confianza**

La razón posible por la que ha recibido el mensaje de error:

- Cuando las variables de entorno SDEUSER o SDEPASSWORD se han establecido en las variables de entorno del sistema o en un archivo dbinit.sde, tendrán prioridad sobre el inicio de sesión de Windows al conectarse a una geodatabase de ArcSDE. No establezca estas variables si va a conectarse desde ese equipo a una geodatabase de ArcSDE en un servidor de base de datos, o si va a conectarse a una geodatabase de ArcSDE con licencia de ArcGIS Server Enterprise utilizando la autenticación del sistema operativo.

*Solución:* asegúrese de no tener establecida una variable de entorno SDEUSER o SDEPASSWORD en el equipo cliente que se conecta.

**Error al agregar la clase de entidad. <nombre\_de\_clase\_de\_entidad> está en modo de solo carga y no se puede agregar como capa en este estado**

La razón posible por la que ha recibido el mensaje de error:

- Si el tamaño de la cuadrícula espacial de un dataset se establece en 0 al intentar importarlo en la geodatabase de ArcSDE, se produce un error en la importación y el dataset puede terminar bloqueado en modo de entrada/salida (E/S) de solo carga. Al volver a calcular el tamaño de cuadrícula del índice espacial, la clase de entidad debería volver a ponerse en modo de E/S normal y, a continuación, debería poder trabajar con ella.

*Solución/solución alternativa:* vuelva a calcular los tamaños de cuadrícula del índice espacial en la clase de entidad.

- En la jerarquía de catálogo, haga clic con el botón derecho en la clase de entidad que está en modo de solo carga, haga clic en Propiedades, a continuación, haga clic en la ficha Índices en el cuadro de diálogo Propiedades de Clase de entidad.
- En la sección Índice espacial, haga clic en Recalcular.
- Haga clic en Aceptar en la ficha Índices.

**Problema: el usuario que se conecta no tiene acceso a las tareas administrativas del servidor de base de datos, como Nuevo geodatabase, Restaurar o Permisos, aunque el usuario ha agregado al grupo de administrador de Windows en el equipo servidor**

Motivo por el que el Administrador de Windows no ha podido realizar las tareas de administrador del servidor:

- Los equipos con Windows Vista, 7 y 2008 Server tienen habilitado UAC de forma predeterminada. Esto fuerza un modelo de seguridad más estricto, lo que significa que los Administradores de Windows no son automáticamente sysadmin en la base de datos de SQL Server. Los usuarios acostumbrados a tener acceso a servidores de base de datos de ArcSDE u otras instancias de SQL Server en equipos con Windows XP o 2003 Server podrían estar habituados al comportamiento predeterminado en estos sistemas operativos y es posible que se olviden de realizar los pasos necesarios para agregar a otros usuarios administrativos del servidor al servidor de base de datos.

*Solución:* deshabilite el control de acceso de usuarios (UAC) en el equipo servidor o inicie de sesión como usuario administrativo que se agregó al servidor de base de datos de ArcSDE cuando se configuró la instancia de SQL Server Express para almacenar geodatabases, agregue al usuario deseado al servidor de base de datos y concédale privilegios de administrador del servidor.

#### **Error: versión no encontrada [sde.DEFAULT]**

Razón por la que ha recibido el mensaje de error:

- Si realiza una conexión de base de datos espacial y se olvida de cambiar la versión de sde.DEFAULT, recibirá este mensaje de error.

*Solución:* el usuario dbo siempre es el propietario de las tablas del sistema y la versión DEFAULT de las geodatabases creadas en servidores de base de datos. Por tanto, debe cambiar la versión de la conexión de sde.DEFAULT a dbo.DEFAULT u otra versión histórica o transaccional disponible.

#### **Error: usuario de inicio de sesión no válido**

La razón posible por la que ha recibido el mensaje de error:

- Probablemente, intentó conectarse a una geodatabase en un servidor de base de datos utilizando un nombre de usuario y contraseña autenticado de base de datos.

*Solución:* las conexiones a las geodatabases en servidores de base de datos solo utilizan inicios de sesión autenticados de Windows. Por consiguiente, elija la opción de autenticación de sistema operativo en el cuadro de diálogo Conexión de base de datos espaciales una vez conectado con un inicio de sesión de Windows válido.

**Error: se produjo un error en el inicio de sesión. El inicio de sesión procede de un dominio que no es de confianza y no se puede utilizar con la autenticación de Windows**

Las posibles razones por las que recibe el mensaje de error:

- Intentó conectarse a una geodatabase en un servidor de base de datos remoto mientras estaba conectado con una cuenta de Windows local.
- Ha intentado conectarse a una geodatabase en un servidor de base de datos con una cuenta de dominio que no se ha agregado al servidor de base de datos.

*Solución:* inicie sesión en el equipo cliente con una cuenta de dominio que un administrador del servidor haya agregado al servidor de base de datos.

**Error: se intentó la conexión con una versión del software de comunicaciones de cliente de SQL Server antigua que no es compatible con el servidor de base de datos de SQL Server**

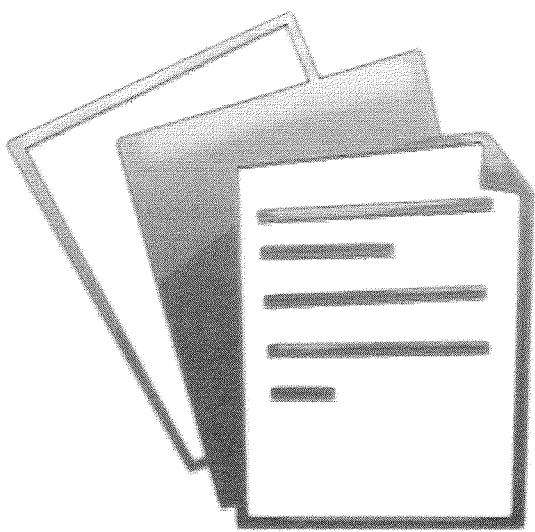
Razón por la que ha recibido el mensaje de error:

- El equipo cliente desde el que se está conectando a la geodatabase no tiene instalada ninguna versión compatible del Microsoft SQL Native Client.

*Solución:* instale la aplicación SQL Native Client en el equipo cliente. Si está conectándose a una base de datos SQL Server 2005 Express, necesita al menos SQL 2005 Native Client. Para conectarse a una base de datos SQL Server 2008 Express, debe instalar SQL 2008 Native Client.

## 2.2. Documentación de las tareas e incidencias realizadas

Los procesos realizados, así como las incidencias producidas en la administración, se reflejan en la documentación para mantener registro de seguimiento de los trabajos realizados, siguiendo el procedimiento establecido por la organización.



La documentación técnica específica asociada se interpreta, en su caso, en la lengua extranjera de uso más frecuente en el sector.

El log vendría a ser un historial que informa qué fue cambiando y cómo fue cambiando (información). Las bases de datos, por ejemplo, utilizan el log para asegurar lo que se llaman las transacciones. Las transacciones son unidades atómicas de cambios dentro de una base de datos; toda esa serie de cambios se encuadra dentro de una transacción, y todo lo que va haciendo la Aplicación (grabar, modificar, borrar) dentro de esa transacción, queda grabado en el log. La transacción tiene un principio y un fin, cuando la transacción llega a su fin, se vuelca todo a la base de datos. Si en el medio de la transacción se cortó por x razón, lo que se hace es volver para atrás. El log te permite analizar cronológicamente qué es lo que sucedió con la información que está en el Sistema o que existe dentro de la base de datos.

La documentación de todo el proyecto de implantación también juega un papel importante, por lo que no se debería descuidar la preparación de diversos materiales, entre los que podríamos citar:

- Documentación técnica del sistema.
- Manuales de procedimientos: pasos a seguir, codificación, etc.
- Manuales de usuario final.
- Material de formación (ejemplos, casos prácticos, etc.).

Los detalles relevantes de la instalación y de la prueba del sistema de CRM, así como las incidencias producidas durante el proceso se reflejan en la documentación para mantener registro y seguimiento de los trabajos realizados siguiendo el procedimiento establecido por la organización.

## RECUERDA

- La palabra **log** es un término anglosajón, equivalente a la palabra bitácora en español. Sin embargo, se utiliza en los países de habla hispana como un anglicismo derivado de las traducciones del inglés en la jerga informática. Del mismo término también proviene la palabra blog, que es la contracción de "web log".
- Los **LOG's**, permiten detectar incidentes y comportamientos no habituales. Esta información es muy útil para localizar fallos en las configuraciones de los programas o cambios que se hayan hecho sobre dicha configuración. También detectan si se ha desconectado al dispositivo del sistema y controlar el uso de recursos por parte del usuario. Además de proporcionar información sobre el rendimiento del sistema, pueden resultar útiles para detectar cuándo un equipo está empezando a fallar.
- Cuando hablamos de **sistema de gestión empresarial ERP** (*Enterprise Resource Planning*) nos referimos a un sistema de gestión integral, que se ha diseñado para facilitar las tareas rutinarias de la empresa. Mediante el ERP conseguimos automatizar estos procesos y al mismo tiempo facilitamos la utilización de todos los recursos disponibles en la empresa.
- Con el **sistema de gestión de clientes CRM** (Customer Relationship Management), vamos a lograr tener en un solo sitio todas las relaciones que tenemos con nuestros clientes y que se refieren a sus necesidades, productos y servicios que demandan de nuestra empresa.
- Un **almacén de datos** o Data Warehouse es básicamente una base de datos que integra datos procedentes de uno o varios sistemas de información de una compañía, y que está orientado a la toma de decisión.
- Generalmente los errores de bases de datos, también llamados errores de motor, ocurren cuando se dan errores en tiempo de ejecución en los registros del código de eventos. Por ejemplo, un error de base de datos ocurre cuando el usuario intenta almacenar un valor nulo en un campo que no admite valores nulos.

- Los procesos realizados, así como las incidencias producidas en la administración, se reflejan en la documentación para mantener registro de seguimiento de los trabajos realizados, siguiendo el procedimiento establecido por la organización.
- La documentación técnica específica asociada se interpreta, en su caso, en la lengua extranjera de uso más frecuente en el sector.

## Preguntas de Autoevaluación

### 1. ¿Qué es un log?

- a) Una base de datos que integra datos procedentes de uno o varios sistemas de información de una compañía, y que está orientado a la toma de decisión.
- b) Una técnica de uso generalizado para supervisar, analizar y evaluar el comportamiento y rendimiento de los sistemas informáticos que están en funcionamiento.
- c) Un registro oficial de eventos durante un rango de tiempo en particular.

### 2. Confirma si la siguiente afirmación es verdadera o falsa:

*"Cuando hablamos de sistema de gestión empresarial ERP (Enterprise Resource Planning) nos referimos a un sistema de gestión integral, que se ha diseñado para facilitar las tareas rutinarias de la empresa. Mediante el ERP conseguimos automatizar estos procesos y al mismo tiempo facilitamos la utilización de todos los recursos disponibles en la empresa".*

- a) Verdadero.
- b) Falso.

### 3. Confirma si la siguiente afirmación es verdadera o falsa:

*"Con el sistema de gestión de clientes CRM (Customer Relationship Management), vamos a lograr tener en un solo sitio todas las relaciones que tenemos con nuestros clientes y que se refieren a sus necesidades, productos y servicios que demandan de nuestra empresa".*

- a) Verdadero.
- b) Falso.

**4. ¿Qué característica corresponde con el almacén de datos o Data Warehouse?**

- a) Volátiles.
- b) No integrados.
- c) Temáticas.

**5. El uso de bases de datos podría crear algunos problemas, especialmente desde el punto de vista del consumidor, señala cuál podrá ser:**

- a) Falta de seguridad.
- b) Independencia.
- c) Abstracción de la información.

# UD4 Transporte de componentes entre entornos de desarrollo, prueba y explotación en almacén de datos

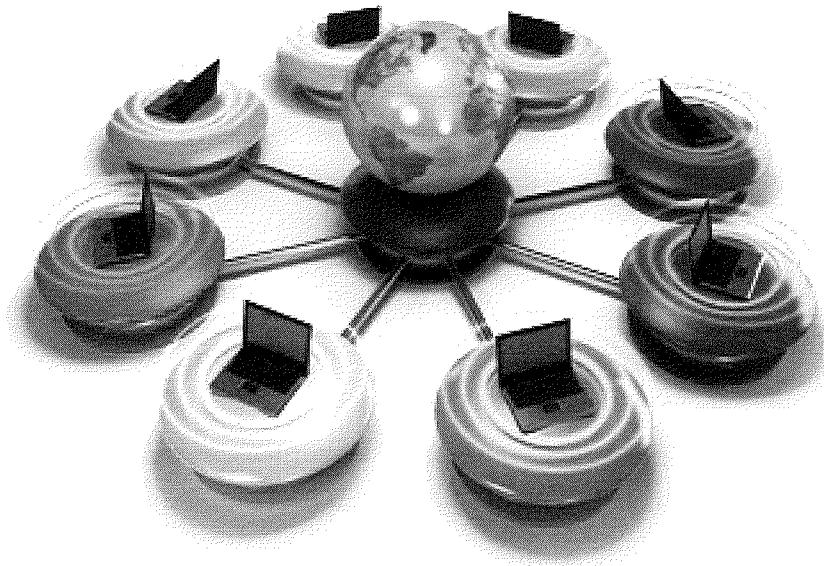


**UF1886 Administración  
del Gestor de Datos en  
Sistemas ERP-CRM**



# 1. El sistema de transmisión de información

Los sistemas de transmisión de datos constituyen el apoyo de los sistemas de cómputo para el transporte de la información que manejan. Sin estos sistemas no hubiera sido posible la creación de las redes avanzadas de cómputo de procesamiento distribuido, en las que compartir información y transferir datos entre computadoras con gran difusión geográfica, sumamente rápido y en grandes volúmenes, es vital para el funcionamiento eficiente de todo el engranaje económico, político y social del mundo.



Los sistemas de transmisión de datos son imprescindibles en redes cuyos enlaces exceden los 20 m. Las redes pueden ser sencillas, como una computadora enlazada a un dispositivo periférico (como una impresora), pasando por la conexión de punto a punto de larga distancia que se satisface con la utilización de módems, o redes ligeramente más complejas que conectan varias terminales de cómputo de edificios lejanos con la computadora principal (anfitriona) de un centro especializado de datos; o una red de área local que se emplea en una empresa para interconectar varios dispositivos de cómputo como: impresoras, dispositivos de almacenamiento de datos (por ejemplo servidores de archivo), etc. Las redes más complejas pueden interconectar las diferentes computadoras grandes de los principales centros financieros del mundo y suministrar a los comerciantes información del mercado de último momento, a través de miles de terminales remotas.

## 1.1. Características en la transmisión de datos

En la actualidad, la transmisión de datos utilizando señales eléctricas ha reemplazado, casi a la totalidad de cualquier otro tipo de transmisión de información, dado que es más fácil controlarla, amplificarla y digitalizarla, para su posterior manipulación.

Los sistemas de transmisión de información, son básicamente de dos tipos:

- Los denominados en banda base, donde no se altera la señal portadora, y la distancia a la que pueden estar emisor y receptor se ve reducida a escasos metros.
- Los sistemas donde se modifica alguna propiedad de la señal portadora (esta modificación se denomina, modulación) y en el otro extremo, se realiza el proceso inverso (denominado de modulación). La distancia entre emisor y receptor en estos sistemas es de cientos de kilómetros.

La evolución de los sistemas de transmisión de información, y a partir de las innovaciones tecnológicas, tanto las del desarrollo electrónicos como de materiales, ha permitido la coexistencia, de sistemas que antes estaban separados. Ahora pueden coexistir y compartir los recursos; por ejemplo, en la actualidad es posible navegar por Internet utilizando la red telefónica o utilizando el videocable, acceder a un canal de noticias por Internet, leer el diario, enviar música por nuestro teléfono móvil, etc.

### 1.1.1. Codificación

El proceso por el cual se asigna a un carácter de un determinado lenguaje o alfabeto, o sistema numérico, un determinado símbolo, se denomina **codificación**.

Existen varios métodos de codificación, como por ejemplo código Morse, códigos de banderas, códigos ASCII, etc.

Ejemplos de codificación:

- El número 7 decimal, se codifica en tres bits en el sistema binario como 111.

- La letra «a» del alfabeto se codifica en ASCII como 97.
- La letra «b» del alfabeto se codifica en código morse como \_...

### 1.1.2. Protocolos

La manera en que se realiza o establece la comunicación entre dos extremos de un sistema de transmisión de la información se basa en la implementación de un conjunto de normas en las que se establecen, las velocidades a la que se realizará la comunicación, cuáles mecanismos de control de errores se establecerán, direcciones de ambos extremos (al igual que en el sistema postal, el cartero necesariamente debe saber a quién entregarle las cartas). Ese conjunto de normas se denomina **protocolo**.

### 1.1.3. Trabajo de red

La posibilidad de «compartir», generalmente va a estar ligado íntimamente con las políticas de seguridad de cada red, esto significa que se tendrá acceso a tal o cual recurso, si estamos autorizados para tal fin. En este sentido es importante conocer el modo en que se estructura y distribuye el servicio de red de una organización y quiénes son los responsables y sus criterios al tomar las decisiones respecto a la seguridad y acceso.

Existe una gran variedad de dispositivos de interconexión, pudiendo conectar sitios distantes a miles de kilómetros o separados sólo por algunos metros. Pueden ser enlaces satelitales, módems, fibras ópticas y enlaces de radio entre otros.

De este modo, las redes de computadoras pueden llegar a infinidad de lugares brindando una cantidad de servicios, tendientes a facilitar la comunicación entre diferentes sitios. En este sentido el avance tecnológico marca enormes cambios en el estilo de vida de la población y fundamentalmente, nuevas posibilidades en el ámbito del trabajo.

Los servicios que brindan las redes, se ven limitados:

- Por la capacidad del dispositivo de interconexión, lo que llamaremos ancho de banda, que es la medida de la cantidad de información que puede transportar.
- Por el mismo ordenador, ya sea por sus recursos físicos, (memoria, procesador, etc.), como sus aplicaciones de software.

Las redes de ordenadores han pasado a formar parte de nuestra vida cotidiana, ya sea desde una terminal de cobro de impuestos, un cajero automático, la tarea diaria en una oficina, y claro está, la Red Internet con todos sus servicios, web, correo electrónico, mensajería instantánea (chat), etc.

#### 1.1.4. Redes

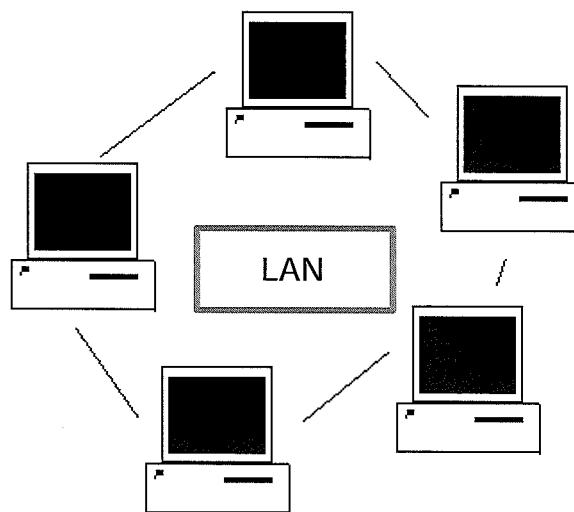
Una **red** es un conjunto de ordenadores conectados entre sí, que pueden comunicarse compartiendo datos y recursos sin importar la localización física de los distintos dispositivos. A través de una red se pueden ejecutar procesos en otro ordenador o acceder a sus ficheros, enviar mensajes, compartir programas...

Los ordenadores suelen estar conectados entre sí por cables. Pero si la red abarca una región extensa, las conexiones pueden realizarse a través de líneas telefónicas, microondas, líneas de fibra óptica e incluso satélites.

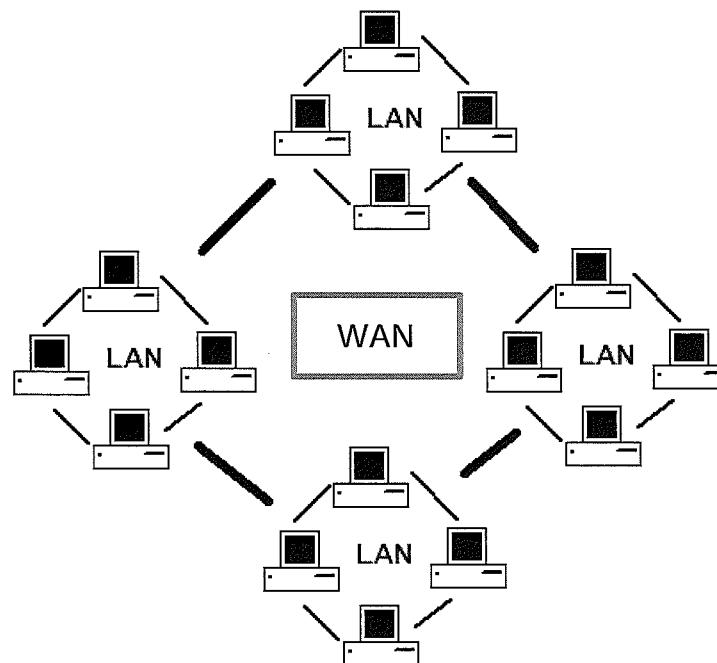
Cada dispositivo activo conectado a la red se denomina **nodo**. Un dispositivo activo es aquel que interviene en la comunicación de forma autónoma, sin estar controlado por otro dispositivo. Por ejemplo, determinadas impresoras son autónomas y pueden dar servicio en una red sin conectarse a un ordenador que las maneje; estas impresoras son nodos de la red.

Dependiendo del territorio que abarca una red se clasifican en:

- **LAN (Local Area Network):** está constituida por un conjunto de ordenadores independientes interconectados entre sí, pueden comunicarse y compartir recursos. Abarcan una zona no demasiado grande, un edificio o un campus.



- **WAN (Wide Area Network):** comprenden regiones más extensas que las LAN e incluso pueden abarcar varios países.



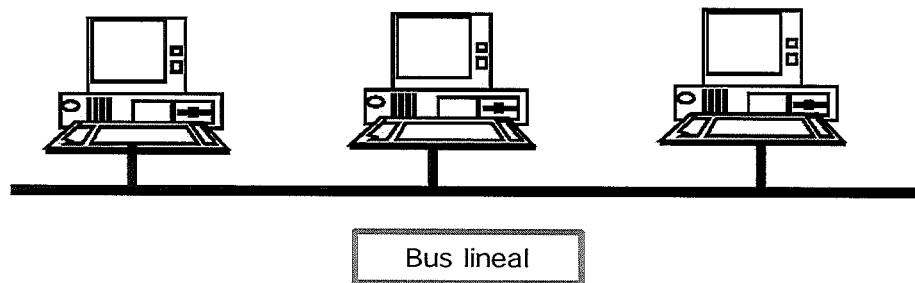
También un conjunto de redes puede conectarse entre sí dando lugar a una red mayor.

### 1.1.5. Topología de la red

Por topología de una red habitualmente se entiende la forma de la red, es decir, la forma en que se lleva a cabo la conexión. Las topologías más utilizadas son: en bus (lineal), en estrella, en árbol y en anillo.

### **Bus lineal**

La topología en bus es un diseño sencillo en el que un solo cable, que es conocido como "bus", es compartido por todos los dispositivos de la red. El cable va recorriendo cada uno de los ordenadores y se utiliza una terminación en cada uno de los dos extremos. Los dispositivos se conectan al bus utilizando generalmente un conector en T.



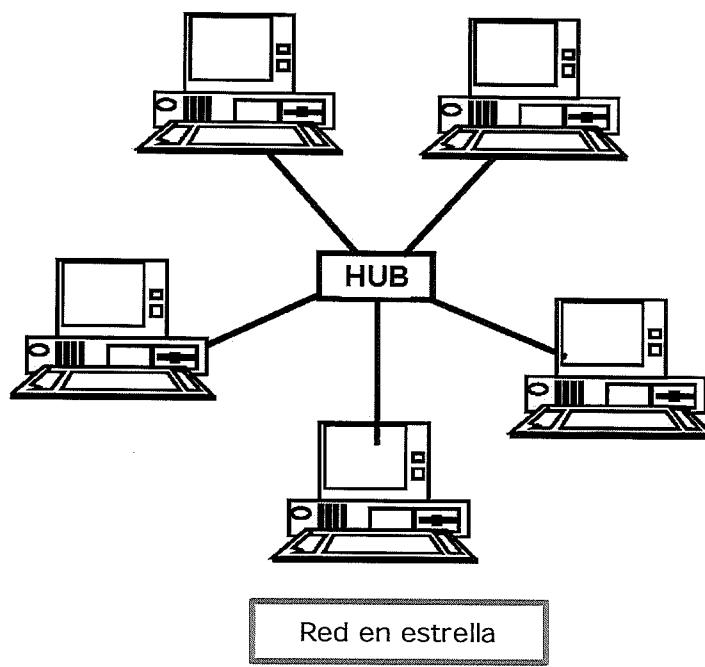
#### *Red en bus*

Las ventajas de las redes en bus lineal son su sencillez y economía. El cableado pasa de una estación a otra. Un inconveniente del bus lineal es que si el cable falla en cualquier punto, toda la red deja de funcionar. Aunque existen diversos procedimientos de diagnóstico para detectar y solventar tales problemas, en grandes redes puede ser sumamente difícil localizar estas averías.

### **Estrella**

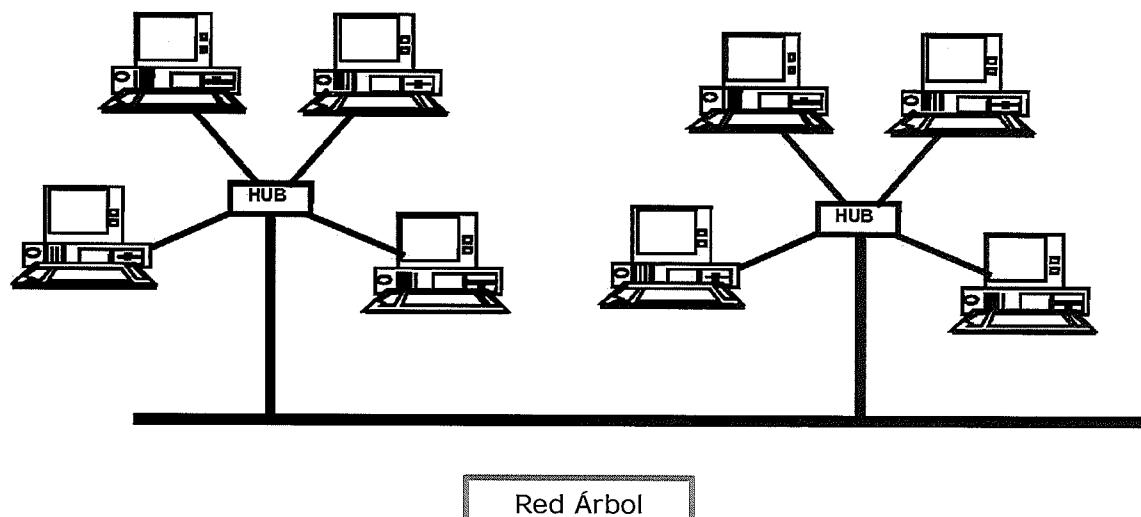
Los nodos de la red se conectan con cables dedicados a un punto que es una caja de conexiones, llamada HUB o concentradores. En una topología en estrella cada estación de trabajo tiene su propio cable dedicado, por lo que habitualmente se utilizan mayores longitudes de cable.

La detección de problemas de cableado en este sistema es muy simple al tener cada estación de trabajo su propio cable. Por la misma razón, la resistencia a fallos es muy alta ya que un problema en un cable afectará sólo a este usuario.



## Árbol

La topología en árbol se denomina también topología en estrella distribuida. Al igual que sucedía en la topología en estrella, los dispositivos de la red se conectan a un punto que es una caja de conexiones, llamado HUB.



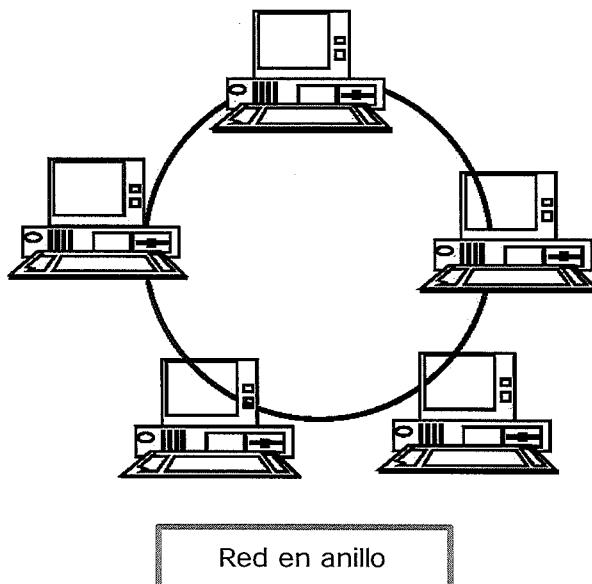
### *Red en árbol*

Estos suelen soportar entre cuatro y doce estaciones de trabajo. Los hubs se conectan a una red en bus, formando así un árbol o pirámide de hubs y

dispositivos. Esta topología reúne muchas de las ventajas de los sistemas en bus y en estrella.

### **Anillo**

En una red en anillo los nodos se conectan formando un círculo cerrado. El anillo es unidireccional, de tal manera que los paquetes que transportan datos circulan por el anillo en un solo sentido.



En una red local en anillo simple, un corte del cable afecta a todas las estaciones, por lo que se han desarrollado sistemas en anillo doble o combinando topologías de anillo y estrella. La red EtherNet cuando utiliza cable coaxial sigue una topología en bus lineal tanto físico como lógico. En cambio al instalar cable bifilar, la topología lógica sigue siendo en bus pero la topología física es en estrella o en estrella distribuida.

#### **1.1.6. Redes de datos**

Las redes de datos se implementan, generalmente, en base a modelos donde se utiliza la abstracción de capas o niveles, ejemplo de esto es el modelo OSI (Open Systems Interconnection), ampliamente utilizado para las redes que

conforman la Internet. Esta abstracción del modelo tiene por objeto reducir la complejidad del diseño de las redes.

El objetivo de cada capa es ofrecer información a las capas superiores, para así independizar a cada una del conocimiento de las otras. Cada una de las capas maneja un protocolo diferente.

Al conjunto de capas y protocolos se lo llama **arquitectura de red**.

### **1.1.7. Arquitectura de redes**

Se han realizado muchos desarrollos en cuanto a diseño y tipo de redes. Como es de esperar, alguno tiene mas uso que otros. Esto se debe, principalmente, al costo asociado que tiene cada uno de estos diseños en relación con su implementación y las prestaciones que pueda tener.

En una red es posible encontrar los siguientes componentes:

- Dispositivos responsables de las aplicaciones, como son los host (PCs, impresoras, terminales, et)
- Dispositivos que realizan la comunicación propiamente (líneas de transmisión, concentradores, comutadores, etc.).

Las líneas de transmisión, (cables, radio, satélite, etc.), son los vínculos entre los host y los dispositivos de conexión (hub, bridge, switch, router, etc.).

Los paquetes en una red LAN pueden ser de tres tipos:

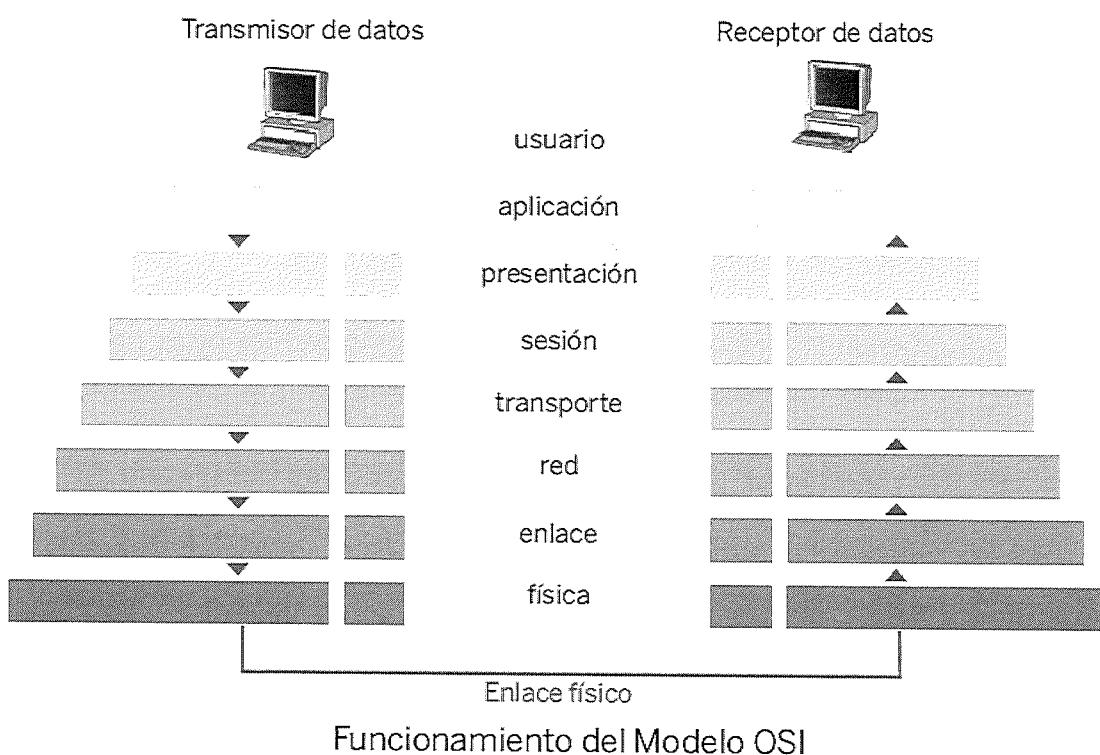
1. Tienen la dirección de destino, y de este modo las otras máquinas lo ignoran (Unicast); ejemplo de esto puede ser, la conexión entre un servidor y una PC.
2. Este tipo de redes también soporta paquetes destinados a todas las máquinas (Broadcast), ejemplo de esto, seria un mensaje del administrador de una red, a todos los usuarios conectados a un servidor, para por ejemplo notificar que el servidor en cuestión se apagara a una determinada hora.
3. Y paquetes destinados a un grupo de máquinas, (Multicast); un ejemplo, la actualización de un antivirus corporativo a un determinado grupo de PC's.

### 1.1.8. Modelo OSI

La Organización de Standares Internaciona (ISO), creó un modelo para el desarrollo de las comunicaciones entre computadoras llamado OSI (Open System Interconnection). Es un modelo de interconexión de sistemas abiertos. Conceptualmente consta de 7 capas, cada una de las cuales tiene una función específica.

Se trata de una representación teórica del modo en que dos PCs se comunican, pero no hace referencia a cuál tipo de software o hardware se debe utilizar en cada capa. De ahí su denominación de abierto ya que sólo se establecen parámetros o condiciones marco, dejando así librado a cada uno de los desarrolladores (ya sea de: cables, placas de red, hub, routers o switch) la manera en que lo hacen. Cada una de las capas esta relacionada con la capa que se encuentra por debajo y por encima de ella.

Las capas son: (1) Aplicación; (2) presentación; (3) Sesión; (4) Transporte; (5) Red; (6) Enlace; (7) Física.

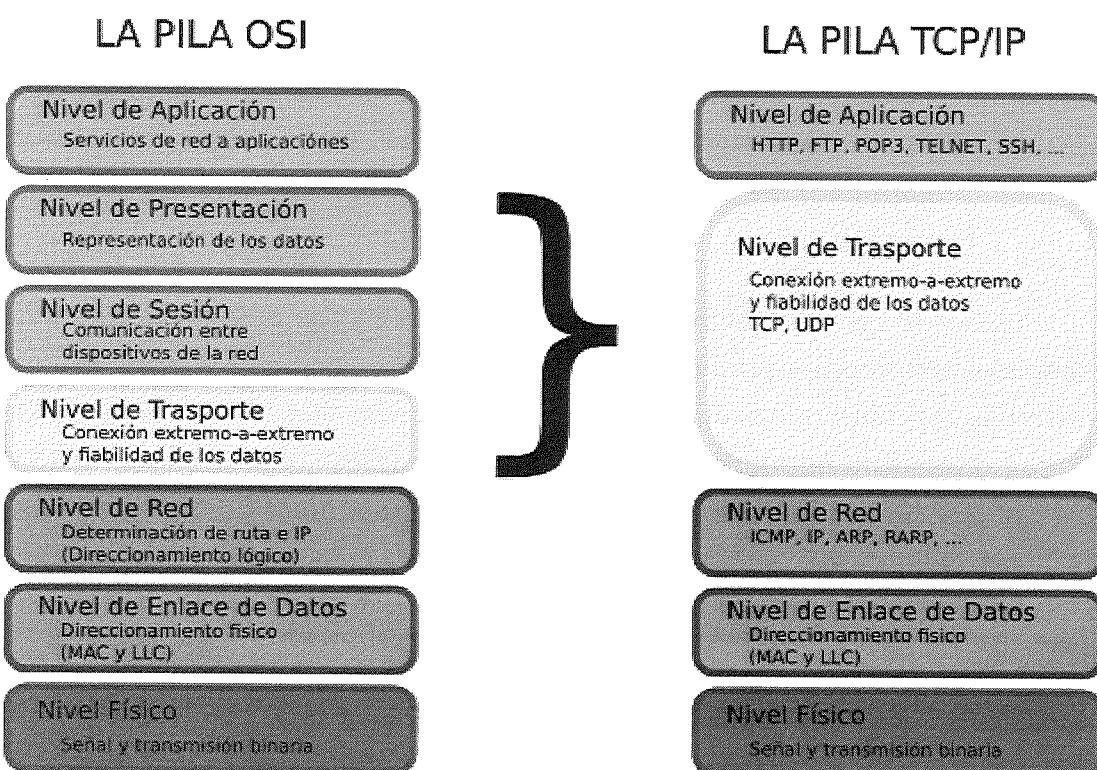


La conexión entre dos computadoras como se ve en la figura se basa en el modelo OSI, cada capa tiene un funcionamiento independiente de la otra, cada una maneja información diferente, y debe pasar información tanto hacia arriba como hacia abajo.

Las capas de la 1 a 4 se denominan **capas inferiores**, y de las capas de la 5 a 7 **capas superiores**.

### 1.1.9. Modelo TCP/IP

TCP/IP es el protocolo común utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre si. Hay que tener en cuenta que en Internet se encuentran conectados ordenadores de clases muy diferentes y con hardware y software incompatibles en muchos casos, además de todos los medios y formas posibles de conexión. Aquí se encuentra una de las grandes ventajas del TCP/IP, pues este protocolo se encargará de que la comunicación entre todos sea posible. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de hardware.



TCP/IP no es un único protocolo, sino que es en realidad lo que se conoce con este nombre es un conjunto de protocolos que cubren los distintos niveles del modelo OSI. Los dos protocolos más importantes son el TCP (Transmission Control Protocol) y el IP (Internet Protocol), que son los que dan nombre al conjunto. En Internet se diferencian cuatro niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI de la siguiente manera:

- **Aplicación:** se corresponde con los niveles OSI de aplicación, presentación y sesión. Aquí se incluyen protocolos destinados a proporcionar servicios, tales como correo electrónico (SMTP), transferencia de ficheros (FTP), conexión remota (TELNET) y otros más recientes como el protocolo HTTP (Hypertext Transfer Protocol).
- **Transporte:** coincide con el nivel de transporte del modelo OSI. Los protocolos de este nivel, tales como TCP y UDP, se encargan de manejar los datos y proporcionar la fiabilidad necesaria en el transporte de los mismos.
- **Internet:** es el nivel de red del modelo OSI. Incluye al protocolo IP, que se encarga de enviar los paquetes de información a sus destinos correspondientes. Es utilizado con esta finalidad por los protocolos del nivel de transporte.
- **Enlace:** los niveles OSI correspondientes son el de enlace y el nivel físico. Los protocolos que pertenecen a este nivel son los encargados de la transmisión a través del medio físico al que se encuentra conectado cada host, como puede ser una línea punto a punto o una red Ethernet.

El TCP/IP necesita funcionar sobre algún tipo de red o de medio físico que proporcione sus propios protocolos para el nivel de enlace de Internet. Por este motivo hay que tener en cuenta que los protocolos utilizados en este nivel pueden ser muy diversos y no forman parte del conjunto TCP/IP. Sin embargo, esto no debe ser problemático puesto que una de las funciones y ventajas principales del TCP/IP es proporcionar una abstracción del medio de forma que sea posible el intercambio de información entre medios diferentes y tecnologías que inicialmente son incompatibles.

Para transmitir información a través de TCP/IP, ésta debe ser dividida en unidades de menor tamaño. Esto proporciona grandes ventajas en el manejo de los datos que se transfieren y, por otro lado, esto es algo común en cualquier protocolo

de comunicaciones. En TCP/IP cada una de estas unidades de información recibe el nombre de "datagrama" (*datagram*), y son conjuntos de datos que se envían como mensajes independientes.

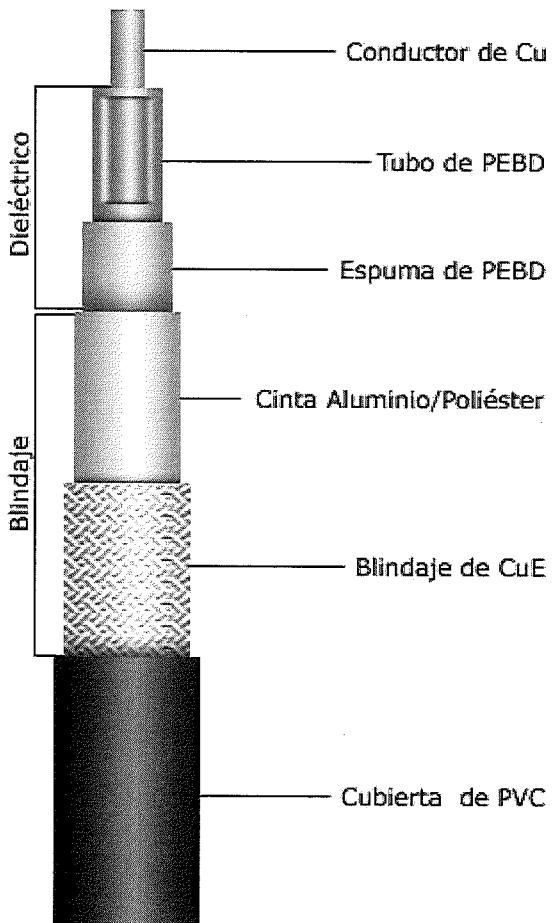
## **1.2. Herramientas que intervienen en la transmisión de datos, configuración de parámetros**

Por medio de la transmisión, la aceptación amplia de la palabra, se entiende el material físico cuyas propiedades de tipo electrónico, mecánico, óptico, o de cualquier otro tipo se emplea para facilitar el transporte de la información entre terminales distantes geográficamente.

El **medio de transmisión** consiste en el elemento que conecta físicamente las estaciones de trabajo al servidor y los recursos de la red. Entre los diferentes medios utilizados en las LANs se puede mencionar: el cable de par trenzado, el cable coaxial, la fibra óptica y el espectro electromagnético (en transmisiones inalámbricas).

### **1.2.1. Cable coaxial**

Hasta hace poco, era el medio de transmisión más común en las redes locales. El cable coaxial consiste en dos conductores concéntricos, separados por un dieléctrico y protegido del exterior por un aislante (similar al de las antenas de TV).



Existen distintos tipos de cable coaxial, según las redes o las necesidades de mayor protección o distancia. Este tipo de cable sólo lo utilizan las redes EtherNet.

Existen dos tipos de cable coaxial:

- **Cable Thick o cable grueso:** es más voluminoso, caro y difícil de instalar, pero permite conectar un mayor número de nodos y alcanzar mayores distancias.
- **Cable Thin o cable fino,** también conocido como *cheapernet* por ser más económico y fácil de instalar. Sólo se utiliza para redes con un número reducido de nodos.

Ambos tipos de cable pueden ser usados simultáneamente en una red. La velocidad de transmisión de la señal por ambos es de 10 Mb.

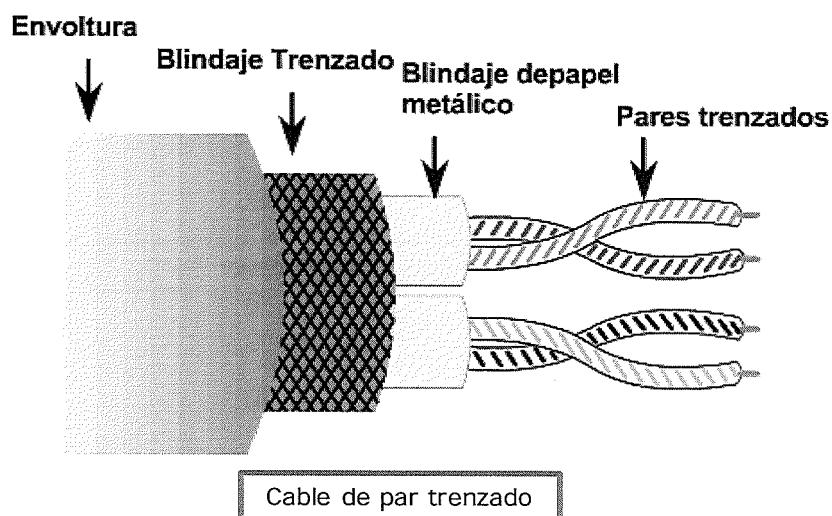
Ventajas del cable coaxial:

- La protección de las señales contra interferencias eléctricas debida a otros equipos, fotocopiadoras, motores, luces fluorescentes, etc.

- Puede cubrir distancias relativamente grandes, entre 185 y 1500 metros dependiendo del tipo de cable usado.

### 2.1.2. Cable bifilar o par trenzado

El par trenzado consta como mínimo de dos conductores aislados trenzados entre ellos y protegidos con una cubierta aislante. Un cable de este tipo habitualmente contiene 1, 2 ó 4 pares, es decir: 2, 4 u 8 hilos.



Los cables trenzados o bifilares constituyen el sistema de cableado usado en todo el mundo para telefonía. Es una tecnología bien conocida. El cable es bastante barato y fácil de instalar y las conexiones son fiables. Sus ventajas mayores son por tanto su disponibilidad y bajo coste.

En cuanto a las desventajas están la gran atenuación de la señal a medida que aumenta la distancia y que son muy susceptibles a interferencias eléctricas. Por este motivo en lugar de usar cable bifilar paralelo se utiliza trenzado y para evitar las interferencias, el conjunto de pares se apantalla con un conductor que hace de malla. Esto eleva el coste del cable en sí, pero su instalación y conexionado

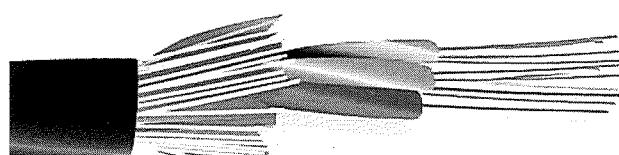
continúa siendo más barato que en el caso de cables coaxiales. Tanto la red EtherNet como la TokenRing pueden usar este tipo de cable.

### **2.1.3. Fibra óptica**

Es el medio de transmisión más moderno y avanzado. Utilizado cada vez más para formar la "espina dorsal" de grandes redes. Las señales de datos se transmiten a través de impulsos luminosos y pueden recorrer grandes distancias (del orden de kilómetros) sin que se tenga que amplificar la señal.



Cable de fibra óptica para interiores



Cable de fibra óptica para exterior ADDS

Por su naturaleza, este tipo de señal y cableado es inmune a las interferencias electromagnéticas y por su gran ancho de banda (velocidad de transferencia), permite transmitir grandes volúmenes de información a alta velocidad.

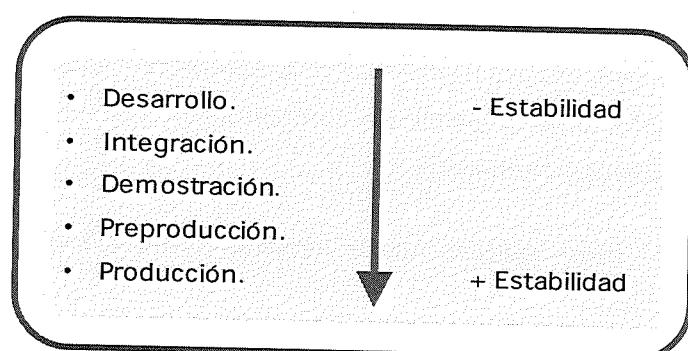
Estas ventajas hacen de la fibra óptica la elección idónea para redes de alta velocidad a grandes distancias, con flujos de datos considerables, así como en instalaciones en que la seguridad de la información sea un factor relevante.

Como inconveniente está, que es el soporte físico más caro. De nuevo, no debido al coste del cable en sí, sino por el precio de los conectores, el equipo requerido para enviar y detectar las ondas luminosas y la necesidad de disponer de técnicos cualificados para realizar la instalación y mantenimiento del sistema de cableado.

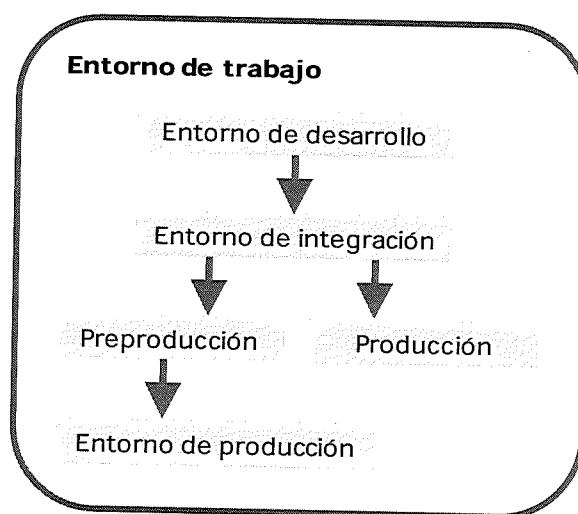
## 2. Entornos de desarrollo

Podemos definir un **entorno**, dentro de nuestro marco de trabajo, como la infraestructura necesaria para acometer las tareas específicas requeridas por el producto software, en función del estado en el que se encuentra.

Un producto software puede encontrarse en alguno de estos estados, en un momento dado, en su evolución:



Cada estado define un nivel de estabilidad del software, y la secuencia de estados representa la evolución del producto software desde su fase más temprana (toma de requerimientos) hasta su puesta en funcionamiento (deployment o despliegue en producción).



Estos cinco estados nos definirán los entornos de trabajo que vamos a describir para el desarrollo de nuestras aplicaciones, y cada entorno se centrará en un subconjunto específico de las tareas de creación y mantenimiento de la aplicación.

## 2.1. Definición de los entornos de desarrollo, pruebas y explotación

A continuación vamos a ver la definición de los entornos de desarrollo, pruebas y explotación.

### 2.1.1. Entornos de desarrollo

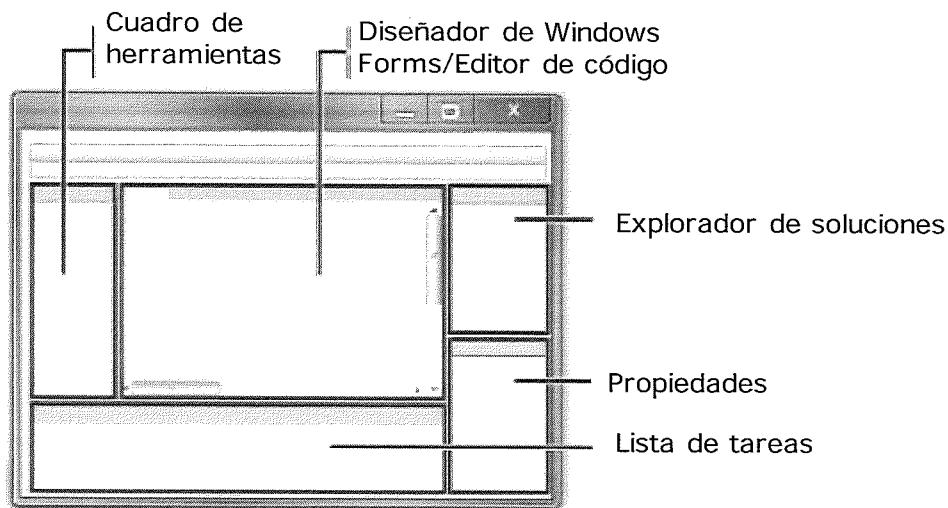
Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de integrated development environment), es una aplicación visual que sirve para la construcción de aplicaciones a partir de componentes. Por lo general todas ellas cuentan con los siguientes elementos:

- Una o más "paletas" para mostrar como iconos los componentes disponibles.
- Un "lienzo" o "contenedor" en el cual se colocan los componentes y se interconectan entre sí.
- Editores específicos para configurar y especializar los componentes.
- Ojeadores (bromen) para localizar componentes de acuerdo a ciertos criterios de búsqueda.
- Directorios de componentes.
- Acceso a editores, interpretes, compiladores y depuradores para desarrollar nuevos componentes.
- Acceso a algunas herramientas de control y gestión de proyectos y CSCW, esenciales para grandes proyectos software.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Ejemplos de IDEs son Visual Studio de Microsoft, VisualAge de IBM o VisualCafe de Symantec, complementados con lenguajes de configuración como VBScript y JavaScript.

Como ejemplo, en la siguiente imagen vemos la representación del entorno integrado (IDE) de visual C#:



### 2.1.1. Entorno de pruebas

El entorno de pruebas se divide a su vez en el entorno de integración y entorno de preproducción. Vemos cada uno de ellos.

#### Entorno de integración

En el entorno de integración se lleva a cabo las siguientes tareas:

- Integración de los distintos módulos que componen la aplicación.
- Pruebas de integración.

En el proceso de integración tiene especial importancia el sistema de control de versiones. Debemos obtener los fuentes etiquetados con la versión estable que deseamos integrar. Cuando los fuentes son obtenidos se compilan y se genera la aplicación con todos los módulos integrados.

Llegados a este punto, debemos poner en funcionamiento la aplicación para someterla a las pruebas de integración. Desplegamos la aplicación basándonos en los documentos generados durante el proyecto. Una vez integrada y activada la aplicación, llevaremos a cabo la secuencia de pruebas. El objetivo de estas pruebas es comprobar el funcionamiento de la aplicación como un todo. En ellas, se trata de probar las funcionalidades que debe cumplir el producto, aunque también debe ser probadas las anteriores funcionalidades para evitar que los cambios introducidos alteren inadvertidamente el comportamiento de las mismas.

Si la aplicación satisface las pruebas de integración, entonces está lista para su paso al entorno de preproducción (para realizar las pruebas de calidad), y al entorno de demostración (para permitir el acceso al cliente).

En el caso en el que se produzca un error en estas pruebas debemos generar una incidencia y emitirla al encargado del módulo en el cual se produjo.

### **Entorno de preproducción**

En el entorno de preproducción se lleva a cabo las pruebas finales de la aplicación antes de su paso final al entorno de producción, donde se pondrá en funcionamiento en un escenario real.

Para ello, en primer lugar se transfiere la aplicación una vez ha sido validada en el entorno de integración, a ser posible de una forma automatizada. El equipo de calidad, someterá la aplicación a un conjunto exhaustivo de pruebas, de diversos tipos:

- Funcionales y estructurales.
- De rendimiento.
- De tolerancia a fallos.
- De seguridad.

Si estas pruebas, conocidas como de aceptación, resultan satisfactorias, entonces la aplicación está ya lista para su paso al entorno de producción.

### **2.1.2. Entorno de explotación**

El entorno de explotación o también conocido con el nombre de entorno de producción contiene en todo momento la versión activa de la aplicación. Los usuarios finales tienen acceso a la aplicación implantada en este entorno, de modo que resulta indispensable planificar y adoptar las medidas de seguridad oportunas, en consonancia con la importancia de la información que maneja el sistema.

Por otra parte, este entorno también contiene los datos reales, información que es preciso salvaguardar frente a posibles pérdidas mediante la aplicación sistemática de una política de copias de seguridad. También debemos proteger los datos frente a exposición o usos fraudulentos de los mismos, restringiendo su acceso exclusivamente al personal de confianza que administra el sistema.

La aplicación se despliega en el entorno de producción procedente de la versión existente en el entorno de preproducción. La subida debería producirse de forma automática para evitar, en la medida de lo posible, la introducción de errores.

## **2.2. Gestión del transporte entre los entornos de desarrollo, pruebas y explotación**

Como hemos visto, las empresas disponen habitualmente de varios entornos donde se desarrolla, prueba y utiliza un sistema de información. Estos entornos suelen llamarse, entorno de desarrollo, entorno de pruebas o preproducción y entorno de producción (vistos anteriormente).

El entorno de desarrollo en ocasiones es local a cada desarrollador. Es decir, cada desarrollador descarga el código del sistema de información localmente, trabaja sobre él y finalmente lo sube a un repositorio común. En otras ocasiones, cuando el entorno de desarrollo requiere de muchos componentes complejos, se opta por disponer de una máquina virtual o un acceso remoto a una máquina que efectivamente se encuentra en un entorno de desarrollo donde están desplegados esos componentes.

El entorno de pruebas o preproducción suele ser uno o dos entornos. El entorno de pruebas sirve para probar el sistema de información, realizar pruebas funcionales y no-funcionales. El entorno de preproducción (que puede ser el mismo que el de pruebas) tiene la característica principal que es idéntico (en la medida de lo posible) al de producción. De esta manera estaremos probando el sistema de información en un entorno idéntico y podremos saber que efectivamente el código funcionará bien cuando se suba a producción.

El entorno de producción es el entorno real donde los usuarios están haciendo uso del sistema de información, donde además existen datos reales que pueden contener datos personales, y que por tanto su criticidad es mucho más alta. Si estropeamos algo en producción, los usuarios son los que mayormente se van a ver afectados.

Sin embargo, en algunas ocasiones, las organizaciones se niegan a realizar cualquier tipo de prueba en producción. Dependiendo del caso, puede ser necesario hacer las pruebas en producción, especialmente cuando no es viable disponer de un

entorno de preproducción idéntico. Por ejemplo, cuando se realizan pruebas de seguridad.

Cualquier organización dedicada al desarrollo de sistemas de información, que tenga la responsabilidad de subir a producción nuevos sistemas de información, debe gestionar adecuadamente el movimiento del código de un entorno a otro. Es necesario automatizar esta tarea en la medida de lo posible incluyendo en el procedimiento medidas de seguridad que permitan la vuelta atrás en caso de que la subida a producción falle. Por ejemplo, cuando hablamos de sistemas de información que son aplicaciones web, un recurso habitual, es que nuestra web apunte a un enlace directo en el sistema de ficheros que a su vez apunta a una carpeta que es la versión actual del código en producción. Cuando se hace una subida a producción se crea una carpeta adicional y lo único que se hace es cambiar en enlace directo desde la versión anterior a la actual. En el caso de que se descubra un problema en producción nuestro plan de vuelta atrás consiste en modificar el enlace directo para que apunte de nuevo a la versión anterior.

## RECUERDA

- En la actualidad, la transmisión de datos utilizando señales eléctricas ha reemplazado, casi a la totalidad de cualquier otro tipo de transmisión de información, dado que es más fácil controlarla, amplificarla y digitalizarla, para su posterior manipulación.
- El proceso por el cual se asigna a un carácter de un determinado lenguaje o alfabeto, o sistema numérico, un determinado símbolo, se denomina **codificación**.
- La manera en que se realiza o establece la comunicación entre dos extremos de un sistema de transmisión de la información se basa en la implementación de un conjunto de normas en las que se establecen, las velocidades a la que se realizará la comunicación, cuáles mecanismos de control de errores se establecerán, direcciones de ambos extremos (al igual que en el sistema postal, el cartero necesariamente debe saber a quién entregarle las cartas). Ese conjunto de normas se denomina **protocolo**.
- La posibilidad de «compartir», generalmente va a estar ligado íntimamente con las políticas de seguridad de cada red, esto significa que se tendrá acceso a tal o cual recurso, si estamos autorizados para tal fin. En este sentido es importante conocer el modo en que se estructura y distribuye el servicio de red de una organización y quiénes son los responsables y sus criterios al tomar las decisiones respecto a la seguridad y acceso.
- Una **red** es un conjunto de ordenadores conectados entre sí, que pueden comunicarse compartiendo datos y recursos sin importar la localización física de los distintos dispositivos. A través de una red se pueden ejecutar procesos en otro ordenador o acceder a sus ficheros, enviar mensajes, compartir programas...
- Las redes de datos se implementan, generalmente, en base a modelos donde se utiliza la abstracción de capas o niveles, ejemplo de esto es el **modelo OSI** (Open Systems Interconnection), ampliamente utilizado para las redes que conforman la Internet. Esta abstracción del modelo tiene por objeto reducir la complejidad del diseño de las redes.

- La Organización de Standares Internacional (ISO), creó un modelo para el desarrollo de las comunicaciones entre computadoras llamado OSI (Open System Interconnection). Es un modelo de interconexión de sistemas abiertos. Conceptualmente consta de 7 capas, cada una de las cuales tiene una función específica.
- **TCP/IP** es el protocolo común utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre si.
- Por **medio de la transmisión**, la aceptación amplia de la palabra, se entiende el material físico cuyas propiedades de tipo electrónico, mecánico, óptico, o de cualquier otro tipo se emplea para facilitar el transporte de la información entre terminales distantes geográficamente.
- Podemos definir un **entorno**, dentro de nuestro marco de trabajo, como la infraestructura necesaria para acometer las tareas específicas requeridas por el producto software, en función del estado en el que se encuentra.
- Un **entorno de desarrollo integrado**, llamado también IDE (sigla en inglés de integrated development environment), es una aplicación visual que sirve para la construcción de aplicaciones a partir de componentes.
- El **entorno de pruebas** se divide a su vez en el entorno de integración y entorno de preproducción.
- El **entorno de explotación** o también conocido con el nombre de **entorno de producción** contiene en todo momento la versión activa de la aplicación. Los usuarios finales tienen acceso a la aplicación implantada en este entorno, de modo que resulta indispensable planificar y adoptar las medidas de seguridad oportunas, en consonancia con la importancia de la información que maneja el sistema.
- Las empresas disponen habitualmente de varios entornos donde se desarrolla, prueba y utiliza un sistema de información. Estos entornos suelen llamarse, entorno de desarrollo, entorno de pruebas o preproducción y entorno de producción.

## Preguntas de Autoevaluación

**1. El proceso por el cual se asigna a un carácter de un determinado lenguaje o alfabeto, o sistema numérico, un determinado símbolo, se denomina:**

- a) Implementación.
- b) Codificación.
- c) Regulación.

**2. La manera en que se realiza o establece la comunicación entre dos extremos de un sistema de transmisión de la información se basa en la implementación de un conjunto de normas en las que se establecen, las velocidades a la que se realizará la comunicación, cuáles mecanismos de control de errores se establecerán, direcciones de ambos extremos (al igual que en el sistema postal, el cartero necesariamente debe saber a quién entregarle las cartas). ¿Cómo se le denomina a ese conjunto de normas?**

- a) Protocolo.
- b) Topología.
- c) Arquitectura de red.

**3. ¿De cuántas capas consta el modelo OSI (Open System Interconnection)?**

- a) De cuatro capas.
- b) De cinco capas.
- c) De siete capas.

**4. Confirma si la siguiente afirmación es verdadera o falsa:**

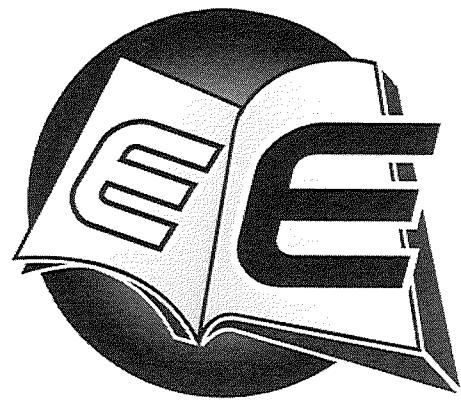
*"TCP/IP es el protocolo común utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre si".*

- a) Verdadero.
- b) Falso.

**5. El entorno de pruebas se divide a su vez en: (respuesta múltiple)**

- a) Entorno de integración.
- b) Entorno de preproducción.
- c) Entorno de producción.

# UD5 Errores en la ejecución del transporte



**UF1886 Administración  
del Gestor de Datos en  
Sistemas ERP-CRM**



# 1. Tipos de errores

Uno de los principales problemas de la programación de aplicaciones, es el tema de migrar de un entorno de desarrollo, a un entorno final y productivo. Esto ocurre, ya que en el ordenador donde estamos programando, normalmente disponemos de todos los permisos necesarios para instalar extensiones que vamos utilizando, sin embargo, en un servidor, la mayoría de las veces no tenemos acceso a este tipo de actualizaciones.

Por ello, es fundamental que desde el principio, es decir, desde que se comienza a trabajar en el entorno de desarrollo para desarrollar la aplicación se tenga en cuenta los siguientes aspectos:

- Verificación de la sintaxis.
- Identificación de incompatibilidades entre componentes.
- Documentación de errores encontrados.

## 1.1. Verificación de la sintaxis

Para que un ordenador (hardware) funcione es necesario utilizar programas (software), los cuales le indican cuál es la tarea que se tiene que hacer. Un lenguaje de programación es el que se utiliza para escribir dichos programas. Posteriormente estos se introducirán en la memoria del computador y éste último ejecutará todas las operaciones que se incluyen.

Los lenguajes de programación constan de:

- Un conjunto finito de símbolos, a partir del cual se define el léxico o vocabulario del lenguaje.
- Un conjunto finito de reglas, la gramática del lenguaje, para la construcción de las sentencias "correctas" del lenguaje. (Sintaxis).
- Semántica, que asocia un significado (la acción que debe llevarse a cabo) a cada posible construcción del lenguaje.

Por tanto, a la forma visible de un lenguaje de programación se le conoce como **sintaxis**. La mayoría de los lenguajes de programación son puramente textuales, es decir, utilizan secuencias de texto que incluyen palabras, números y

puntuación, de manera similar a los lenguajes naturales escritos. Por otra parte, hay algunos lenguajes de programación que son más gráficos en su naturaleza, utilizando relaciones visuales entre símbolos para especificar un programa.

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

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

```

expresión ::= átomo | lista
átomo ::= número | símbolo
número ::= [+-]? ['0'-'9']+
símbolo ::= ['A'-'Z'] ['a'-'z']*
lista ::= '(' expresión* ')'

```

Con esta gramática se especifica lo siguiente:

- Una expresión puede ser un átomo o una lista;
- un átomo puede ser un número o un símbolo;
- un número es una secuencia continua de uno o más dígitos decimales, precedido opcionalmente por un signo más o un signo menos;
- un símbolo es una letra seguida de cero o más caracteres (excluyendo espacios); y
- una lista es un par de paréntesis que abren y cierran, con cero o más expresiones en medio.

Algunos ejemplos de secuencias bien formadas de acuerdo a esta gramática:

'12345', '()', '(a b c232 (1))'

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

El siguiente fragmento en el lenguaje C es sintácticamente correcto, pero ejecuta una operación que no está definida semánticamente (dado que p es un apuntador nulo, las operaciones p->real y p->im no tienen ningún significado):

```
complex *p = NULL;
complex abs_p = sqrt (p->real * p->real + p->im * p->im);
```

Si la declaración de tipo de la primera línea fuera omitida, el programa dispararía un error de compilación, pues la variable "p" no estaría definida. Pero el programa sería sintácticamente correcto todavía, dado que las declaraciones de tipo proveen información semántica solamente.

La gramática necesaria para especificar un lenguaje de programación puede ser clasificada por su posición en la **Jerarquía de Chomsky**. La sintaxis de la mayoría de los lenguajes de programación puede ser especificada utilizando una gramática Tipo-2, es decir, son gramáticas libres de contexto. Algunos lenguajes, incluyendo a Perl y a Lisp, contienen construcciones que permiten la ejecución durante la fase de análisis. Los lenguajes que permiten construcciones que permiten al programador alterar el comportamiento de un analizador hacen del análisis de la sintaxis un problema sin decisión única, y generalmente oscurecen la separación entre análisis y ejecución. En contraste con el sistema de macros de Lisp y los bloques BEGIN de Perl, que pueden tener cálculos generales, las macros de C son meros reemplazos de cadenas, y no requieren ejecución de código.

### **1.1.1. Errores de sintaxis**

Un error de sintaxis ocurre cuando el código se escribe incorrectamente, por ejemplo se escribe mal una sentencia o se emite o se requiere cierta puntuación, o no se escriben estructuras completas como por ejemplo la falta de punto y coma, falta de llaves, no declarar variables, etc.

Este tipo de errores son los más comunes y los más fáciles de corregir ya que es básicamente el compilador quien los identifica y hasta muestra la línea donde se encuentra dicho error.

A continuación veremos los mensajes de error más comunes relacionados con la sintaxis en programación con C++.

- **Statement missing ";"**: quiere decir que al final le falta punto y coma.
- **Compound statement missing "}"**: quiere decir que le falta una llave. Generalmente todas las llaves que se abren deben cerrarse; es muy común no cerrar todas las llaves.
- **Declaration syntax error**: error de sintaxis en declaración; quiere decir que le falta alguna palabra.
- **Undefined symbol**: símbolo indefinido; cuando por ejemplo alguna sentencia está mal escrita.
- **Undefined symbol "cont"**: no se ha definido la variable "cont"; sucede cuando nos olvidamos declarar una variable.
- **Multiple declaration for "cont"**: múltiple declaración para "cont"; las variables sólo se deben declarar una vez, si declaramos dos veces nos mostrara este mensaje.

## 1.2. Identificación de incompatibilidades entre componentes

La compatibilidad es la condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente tanto directamente o indirectamente (mediante un algoritmo). A este algoritmo que hace que un programa logre ser comprendido por un sistema, arquitectura o aplicación se lo denomina emulador por el hecho de que es un intérprete entre el programa y el sistema, arquitectura o aplicación.

### 1.2.1. Problemas de compatibilidad

Un problema de compatibilidad (incompatibilidad) surge a partir de la falta o mala interpretación de un programa por un algoritmo, esto conlleva a una mala ejecución de dicho programa o a la imposibilidad de ser ejecutado.

Un ejemplo práctico:

- **Compatibilidad:** el programa le indica una orden al sistema y el sistema la interpreta y la ejecuta sin problemas.

```
programa_orden_decir=("5%5Dos")
sistema> WUACOhN
```

- **Incompatibilidad Caso A (Mala ejecución):** el programa le indica una orden al sistema y el sistema la interpreta pero de forma errónea, devolviendo un resultado no esperado.

```
programa_orden_decir=("Hola")
sistema> Chau
```

- **Incompatibilidad Caso B (Imposibilidad de ejecución):** el programa le indica una orden al sistema que para él es arbitraria y por ende no logra interpretarla.

```
programa_orden_da31s4s232sd2453ce
sistema> Error
```

### 1.2.2. Emulación

La emulación consiste en utilizar un algoritmo de por medio, denominado **emulador** que simula ser el sistema, arquitectura o aplicación para el cual el programa está preparado, el emulador modifica los comandos del programa en memoria para que el sistema pueda interpretarlo como si estuviera especialmente diseñado para él.

Un ejemplo:

- Ejecución de orden del programa:

```
programa_orden_decir=("hputa")
```

- Ejecución del emulador:

```
emul transformar programa_orden_* en sistema_realizar_*
```

- Ejecución de orden del programa emulada en memoria:

```
sistema_realizar_decir=("Hola")
```

- Resultado en el sistema:

```
sistema> Hola
```

### 1.2.3. Open Source

Hoy en día los programas Open Source (código abierto), generalmente en los sistemas basados en Unix (Linux) lograron solucionar bastante el tema de la compatibilidad, por el hecho de que el sistema que compilará el programa, podrá antes adaptar el código a su kernel modificando opciones de compilación, generalmente ingresando en la consola el siguiente comando: **./configure**.

Luego compila el código con el siguiente comando:

```
make
```

Y por último instala los ejecutables compilados con:

```
make-install
```

Logrando así obtener un programa genérico completamente adaptado al sistema operativo que lo ha compilado.

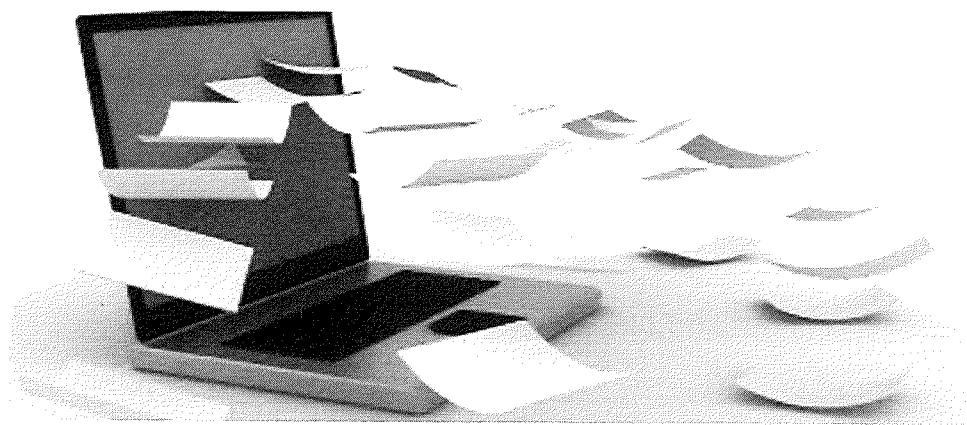
### **1.3. Documentación de errores encontrados**

Los procesos realizados, así como las incidencias producidas en la administración, se reflejan en la documentación para mantener registro de seguimiento de los trabajos realizados, siguiendo el procedimiento establecido por la organización.

La documentación técnica específica asociada se interpreta, en su caso, en la lengua extranjera de uso más frecuente en el sector.

Los informes de auditoría deberán ajustarse a los principios y normas de auditoría informática generalmente aceptados (NAIGA).

El informe es el instrumento que se utiliza para comunicar los objetivos de la auditoría, el alcance que va a tener, las debilidades que se detectan y las conclusiones a las que se lleguen. A la hora de preparar el informe, el auditor debe tener en cuenta las necesidades y características de los que se suponen serán sus destinatarios. El informe debe contener un párrafo en el que se indique los objetivos que se pretenden cumplir. Si, según la opinión del autor, alguno de estos objetivos no se pudiera alcanzar, se debe indicar en el informe.



### **2. Resolución de problemas**

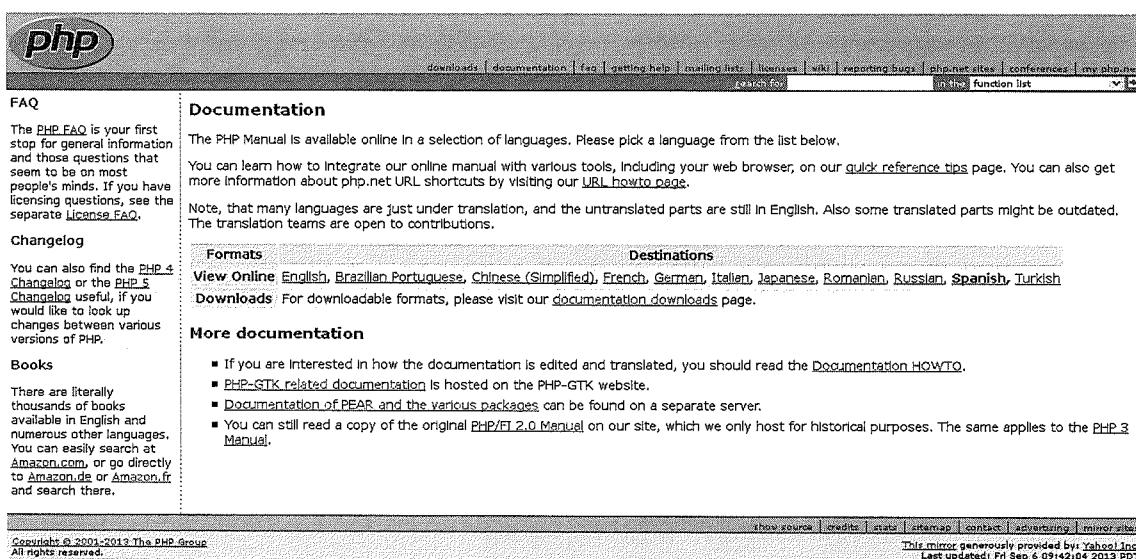
Para la resolución de problemas encontrados en la ejecución del transporte es recomendable la utilización de manuales, que son los que a continuación vamos a ver. Además también es necesario que se realice unos documentos donde se exponga las soluciones que se han ofrecido a cada problema encontrado.

## 2.1. Utilización de manuales para la resolución de problemas

Cuando hablamos de manuales para la resolución de problemas, nos estamos refiriendo, como bien su nombre indica, a un documento en el cual se establecen los procedimientos que se deben seguir ante un problema en concreto encontrado cuando no se ejecuta correctamente la aplicación.

Como hemos visto con anterioridad, los tipos de errores que podemos encontrarnos en el desarrollo de la programación son muy diversos, aunque entre ellos se destacan los errores de sintaxis y la incompatibilidad entre componentes. Generalmente cada lenguaje de programación (PHP, JavaScript, etc.) tiene unas características concretas y por tanto, cada uno unas reglas específicas que deben seguirse para que la aplicación se ejecute correctamente.

En la actualidad, las Web oficiales de cada uno de los lenguajes de programación tienen diferentes apartados en cuales se exponen los problemas que pueden surgir con dicho lenguaje y sus respectivas soluciones. Por ejemplo si vamos a la página oficial de PHP, podemos encontrar un apartado denominado documentación.



En este apartado podremos encontrar todo lo relacionado con el lenguaje PHP, además de muchas otras opciones, así como grupos o foros en los que la gente puede expresar sus opiniones y sus experiencias.

## **2.2. Documentación de las soluciones aportadas**

Es muy importante llevar un registro de las soluciones aportadas a todos los problemas encontrados en la ejecución del transporte entre entornos de desarrollo, prueba y explotación. De esta forma podremos llevar un control mucho más estricto en el desarrollo de nuestra aplicación.



## RECUERDA

- Uno de los principales problemas de la programación de aplicaciones, es el tema de migrar de un entorno de desarrollo, a un entorno final y productivo. Esto ocurre, ya que en el ordenador donde estamos programando, normalmente disponemos de todos los permisos necesarios para instalar extensiones que vamos utilizando, sin embargo, en un servidor, la mayoría de las veces no tenemos acceso a este tipo de actualizaciones.
- Para que un ordenador (hardware) funcione es necesario utilizar programas (software), los cuales le indican cuál es la tarea que se tiene que hacer. Un lenguaje de programación es el que se utiliza para escribir dichos programas. Posteriormente estos se introducirán en la memoria del computador y éste último ejecutará todas las operaciones que se incluyen.
- A la forma visible de un lenguaje de programación se le conoce como **sintaxis**. La mayoría de los lenguajes de programación son puramente textuales, es decir, utilizan secuencias de texto que incluyen palabras, números y puntuación, de manera similar a los lenguajes naturales escritos. Por otra parte, hay algunos lenguajes de programación que son más gráficos en su naturaleza, utilizando relaciones visuales entre símbolos para especificar un programa.
- Un **error de sintaxis** ocurre cuando el código se escribe incorrectamente, por ejemplo se escribe mal una sentencia o se emite o se requiere cierta puntuación, o no se escriben estructuras completas como por ejemplo la falta de punto y coma, falta de llaves, no declarar variables, etc.
- La **compatibilidad** es la condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente tanto directamente o indirectamente (mediante un algoritmo). A este algoritmo que hace que un programa logre ser comprendido por un sistema, arquitectura o aplicación se lo denomina emulador por el hecho de que es un intérprete entre el programa y el sistema, arquitectura o aplicación.
- La emulación consiste en utilizar un algoritmo de por medio, denominado **emulador** que simula ser el sistema, arquitectura o

aplicación para el cual el programa está preparado, el emulador modifica los comandos del programa en memoria para que el sistema pueda interpretarlo como si estuviera especialmente diseñado para él.

- Hoy en día los programas Open Source (código abierto), generalmente en los sistemas basados en Unix (Linux) lograron solucionar bastante el tema de la compatibilidad, por el hecho de que el sistema que compilará el programa, podrá antes adaptar el código a su kernel modificando opciones de compilación, generalmente ingresando en la consola el siguiente comando: **./configure**.
- Los procesos realizados, así como las incidencias producidas en la administración, se reflejan en la documentación para mantener registro de seguimiento de los trabajos realizados, siguiendo el procedimiento establecido por la organización.

## Preguntas de Autoevaluación

### 1. Se denomina sintaxis en programación:

- a) A la forma visible de un lenguaje de programación.
- b) Al conjunto finito de símbolos.
- c) A la construcción del lenguaje en PHP.

### 2. Confirma si la siguiente afirmación es verdadera o falsa:

"La sintaxis de los lenguajes de programación es definida generalmente utilizando una combinación de expresiones regulares (para la estructura léxica) y la Notación de Backus-Naur (para la estructura gramática)".

- a) Verdadero.
- b) Falso.

### 3. ¿Qué quiere decir el mensaje de error "Undefined symbol" relacionado con la sintaxis en programación con C++?

- a) Quiere decir que al final le falta punto y coma.
- b) Quiere decir que le falta alguna palabra.
- c) Significa símbolo indefinido y se da cuando por ejemplo alguna sentencia está mal.

**4. Confirma si la siguiente afirmación es verdadera o falsa:**

*"Un problema de compatibilidad (incompatibilidad) surge a partir de la falta o mala interpretación de un programa por un algoritmo, esto conlleva a una mala ejecución de dicho programa o a la imposibilidad de ser ejecutado".*

- a) Verdadero.
- b) Falso.

**5. Confirma si la siguiente afirmación es verdadera o falsa:**

*"Los informes de auditoría deberán ajustarse a los principios y normas de auditoría informática generalmente aceptados (NAIGA)".*

- a) Verdadero.
- b) Falso.

# UD6 Procesos de extracción de datos en sistemas de almacén de datos



**UF1886 Administración  
del Gestor de Datos en  
Sistemas ERP-CRM**

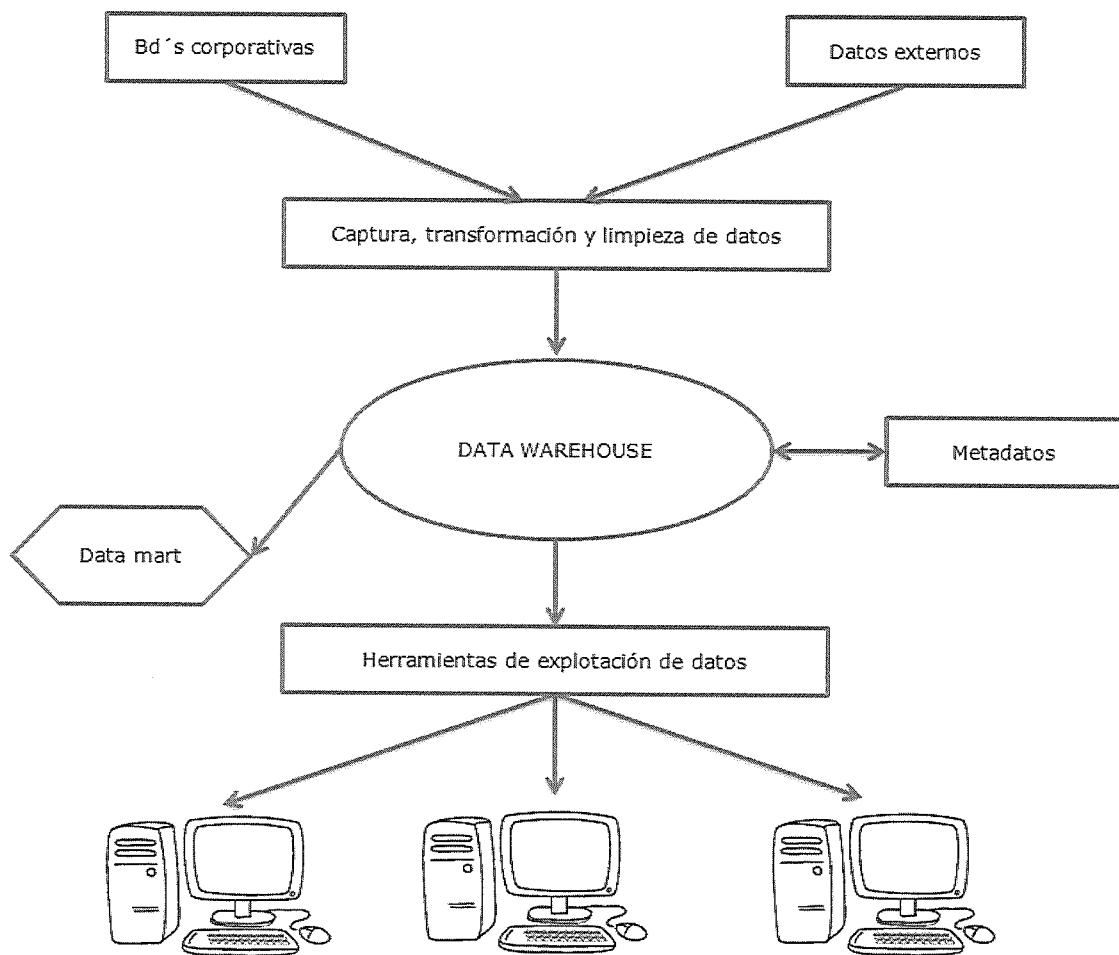


## 1. Características y funcionalidades

Antes de empezar a centrarnos en los procesos de extracción de datos es necesario conocer la arquitectura de un Data Warehouse; es decir cual es su estructura y que elementos la componen. Dichos elementos son:

- **Fuentes de datos:** pueden ser sistemas operacionales corporativos, con información relativa a la actividad rutinaria de la empresa, sistemas operacionales departamentales, fuentes externas, etc.
- **Extracción y transformación:** proceso que permite que la información pase desde las fuentes de datos, hasta la DW propiamente dicha.
- **Servidor o gestor de los datos:** su misión es la de mantener, distribuir, aportar seguridad y monitorización.
- **Herramientas de acceso y análisis:** son las llamadas business intelligence tool (BIT), cada usuario debe seleccionar qué herramientas se adaptan mejor a sus necesidades y a su DW.
- **Metadatos:** contienen información acerca de los datos contenidos en el DW. Entre sus funciones destacan: catalogar y describir la información, especificar el propósito de la misma, indicar las relaciones entre los datos, establecer quién es el propietario de la información, relacionar las estructuras técnicas de datos con la información de negocio, establecer las relaciones con los datos operacionales y las reglas de transformación y por último, limitar la validez de la información.
- **Data Marts:** son almacenes de datos especializados, diseñados para soportar necesidades de análisis específicas para un único departamento o área funcional de la empresa, por ejemplo marketing, finanzas, producción, etc. Estos almacenes soportan menos usuarios y menos cantidades de datos que un DW centralizado, y por lo tanto pueden ser optimizados para cargar y recuperar la información de forma más rápida y eficaz que un DW.

En el siguiente esquema vemos de forma más clarificada la arquitectura que presenta un Data Warehouse.



## Ambiente de Data Warehouse

Es básicamente el bloque donde se almacenan los datos informativos, utilizando principalmente para usos estratégicos. Sin embargo, existen herramientas que no hacen uso de este bloque, realizando las consultas multidimensionales directamente sobre la base operacional. En este caso se puede pensar en el Data Warehouse simplemente como una vista lógica o virtual de datos.

### Data mart

Un Data Mart es una implementación de un Data Warehouse con un determinado alcance de información y un soporte limitado para procesos analíticos, que sirve a un sólo departamento de una organización o para el análisis de problemas de un tema particular.

El Data Mart es un subconjunto de información corporativa con formato adicional a la medida de un usuario específico del negocio. Un Data Mart será siempre menor en complejidad y alcance de los datos.

Un Data Warehouse tiene más usuarios y más temas que un Data Mart, brindando una vista más amplia entre múltiples áreas.

Existen dos grandes filosofías con respecto a la relación entre los conceptos de Data Warehouse y Data Mart.

Bill Inmon, quien es considerado el padre del Data Warehouse, propuso la idea de que los Data Marts se sirven del Data Warehouse para extraer información. La misma está almacenada en tercera forma normal, en un modelo relacional.

Por su parte, Ralph Kimball, el principal impulsor del enfoque dimensional para el diseño del Data Warehouse, sostiene que el Data Warehouse es el resultado de la unión de los Data Marts de la empresa.

## **Metadatos**

Los metadatos son datos acerca de los datos. En una base de datos los metadatos son la representación de los diversos objetos que definen una base de datos, por ejemplo, ubicación y descripción de base de datos, tablas, nombres y resúmenes. También podemos mencionar las descripciones lógicas y físicas de tablas, columnas y atributos.

Uno de los problemas con el que pueden encontrarse los usuarios de un data warehouse es saber lo que hay en él y cómo pueden acceder a lo que quieren. A fin de proveer el acceso a los datos universales, es absolutamente necesario mantener los metadatos. Un componente llamado repositorio les ayuda a conseguirlo.

Los metadatos son sólo una de las utilidades del repositorio, pero éste tiene muchas funcionalidades: catalogar y describir la información disponible, especificar el propósito de la misma, indicar las relaciones entre los distintos datos, establecer quién es el propietario de la información, relacionar las estructuras técnicas de datos con la información de negocio, establecer las relaciones con los datos operacionales y las reglas de transformación, y limitar la validez de la información.

## ETL (Extraer, Trasformar y Cargar)

Para poblar el Data Warehouse se deben mover bloques de datos, muchas veces desde diferentes sistemas operativos, estructuras de archivos y bases de datos, mediante procesos programados que se ejecutan frecuentemente fuera del horario de trabajo para no insumir tiempo de procesamiento del hardware de la empresa, entorpeciendo la operatoria de la misma.

Los subsistemas para poblar el Data Warehouse se pueden construir utilizando herramientas y productos disponibles en el mercado, programas y procesos codificados desde cero, o combinaciones de estos elementos.

Al construir los sistemas para poblar el Data Warehouse, se debe considerar la posibilidad de que estos permitan regular el crecimiento evolutivo del Data Warehouse, brindando escalabilidad y soporte para grandes cantidades de datos y consultas complejas. Se pueden encontrar dificultades adicionales dependiendo de las fuentes de datos que se tengan disponibles, que implican el uso de diferentes herramientas y tecnologías para acceder a cada uno de ellos.

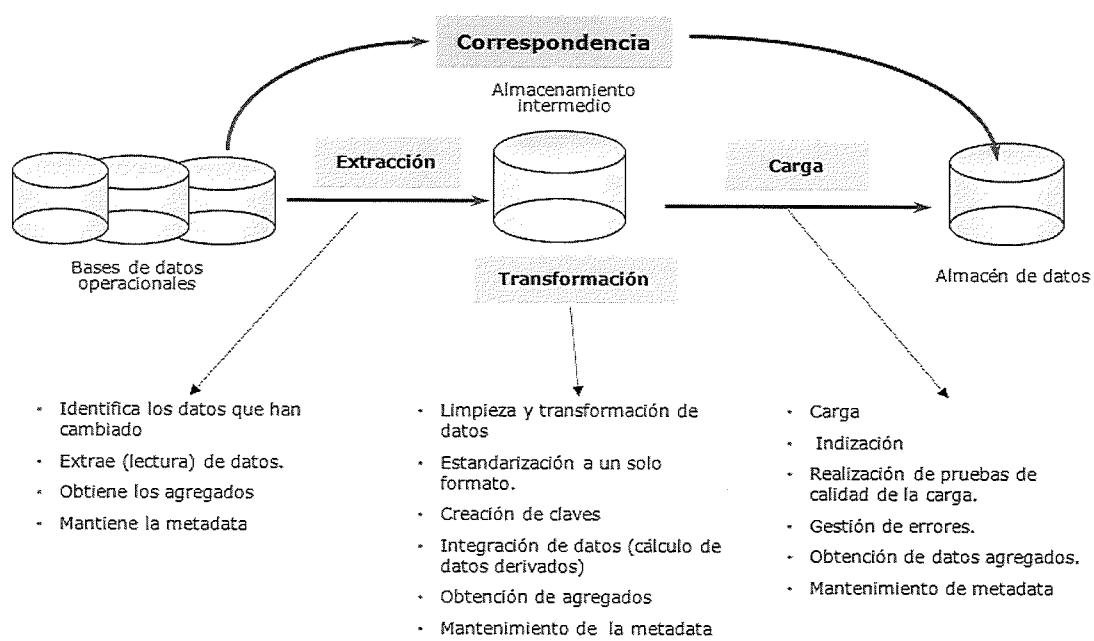
Los procesos que se conectan a los datos de los sistemas fuente, los recodifican, los integran y los cargan dentro del almacén de datos, reciben el nombre de procesos ETL, que son las siglas en inglés de **Extraer, Transformar y Cargar**.

Los procesos ETL son probablemente los componentes más importantes y de mayor valor añadido en una infraestructura que implique la integración de varias fuentes de datos. En consecuencia, representan un pilar fundamental tanto de simples proyectos de recopilación como de soluciones complejas.

Aunque suelen resultar transparentes a los usuarios, los procesos ETL son los encargados de recuperar información de todos los orígenes necesarios, formatearla, limpiarla e integrarla en un data mart, un datawarehouse, una base de conocimiento o cualquier otro tipo de repositorio digital. En resumen, los procesos ETL recopilan los datos y hacen posible que la información subyacente pueda ser presentada mediante las herramientas de análisis y reporting pertinentes.

Como su propio nombre indica, los procesos ETL se dividen en tres fases:

- **Extracción:** consiste en obtener los datos del sistema origen, realizando volcados completos o incrementales. En ocasiones esta etapa suele apoyarse en un almacén intermedio, llamado ODS (Operational Data Store), que actúa como pasarela entre los sistemas fuente y los sistemas destino, y cuyo principal objetivo consiste en evitar la saturación de los servidores funcionales de la organización.
- **Transformación:** los datos procedentes de repositorios digitales distintos no suelen coincidir en formato. Por tanto, para lograr integrarlos resulta imprescindible realizar operaciones de transformación. El objetivo no es otro que evitar duplicidades innecesarias e impedir la generación de islas de datos inconexas. Las transformaciones aplican una serie de reglas de negocio (o funciones) sobre los datos extraídos para convertirlos en datos destino.
- **Carga:** se trata de introducir los datos, ya adaptados al formato deseado, dentro del sistema destino. En algunos casos se sobreescribe la información antigua con la nueva, mientras que en otros se guarda un historial de cambios que permite consultas retrospectivas en el tiempo, así como revertir modificaciones. Para la carga masiva de datos suele ser necesario desactivar temporalmente la integridad referencial de la base de datos destino.

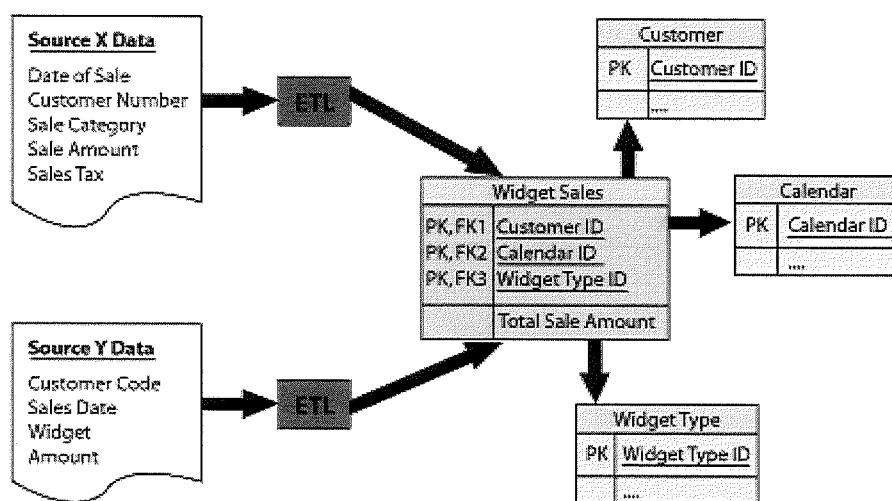


A continuación veremos de forma más detallada cada una de estas fases que componen los procesos ETL.

## 1.1. Definición de la estructura que interviene en los procesos de extracción de datos

La primera parte del proceso ETL consiste en extraer los datos desde los sistemas de origen. La mayoría de los proyectos de almacenamiento de datos fusionan datos provenientes de diferentes sistemas de origen. Cada sistema separado puede usar una organización diferente de los datos o formatos distintos. Los formatos de las fuentes normalmente se encuentran en bases de datos relacionales o ficheros planos, pero pueden incluir bases de datos no relacionales u otras estructuras diferentes. La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.

El propósito principal de la fase de extracción es capturar y copiar los datos requeridos de uno o más sistemas operacionales o fuentes de datos. Los datos que se extraen son colocados en un archivo intermedio con un formato definido, que luego será utilizado por la siguiente fase del proceso.



Los registros que sean rechazados en el proceso deben ser registrados en un archivo o log de rechazos para que puedan ser analizados posteriormente y así tener la posibilidad de cargarlos en el Data Warehouse correctamente. Además, esto permite descubrir los errores que han ocurrido en los procesos de creación de los datos operacionales.

Ejemplos de estos errores son violaciones de integridad, claves duplicadas, formatos de datos incorrectos y datos inválidos como campos vacíos, fechas futuras e importes negativos cuando estos no correspondan.

Además, esto permite descubrir los errores que han ocurrido en los procesos de creación de los datos operacionales. Ejemplos de estos errores son violaciones de integridad, claves duplicadas, formatos de datos incorrectos y datos inválidos como campos vacíos, fechas futuras e importes negativos cuando estos no correspondan.

Un requerimiento importante que se debe exigir a la tarea de extracción es que ésta cause un impacto mínimo en el sistema origen. Si los datos a extraer son muchos, el sistema de origen se podría ralentizar e incluso colapsar, provocando que éste no pueda utilizarse con normalidad para su uso cotidiano. Por esta razón, en sistemas grandes las operaciones de extracción suelen programarse en horarios o días donde este impacto sea nulo o mínimo.

## **1.2. Tratamiento de la información y transformación de estos datos para facilitar la toma de decisiones**

Las funciones básicas a ser realizadas en esta fase consisten en leer los archivos intermedios generados por la fase de extracción, realizar las transformaciones necesarias, construir los registros en el formato del Data Warehouse y crear un archivo de salida conteniendo todos los registros nuevos a ser cargados en el Data Warehouse.

La mayor parte del trabajo en esta fase involucra el efectuar las transformaciones necesarias. Estas transformaciones incluyen:

- Combinar campos múltiples de nombres y apellidos en un solo campo.
- Fusionar campos o datos homónimos.
- Separar un campo de fecha en campos de año, mes y día.
- Cambiar la representación de los datos, como TRUE (verdadero) a 1, y FALSE (falso) a 0, o códigos postales numéricos a alfanuméricos, respetando los estándares de la empresa.
- Cambiar un dato que tiene múltiples representaciones a una sola representación, como por ejemplo definir un formato común para números telefónicos, o establecer un término común para los nombres de los campos o los valores de los datos que sean sinónimos.

### 1.3. Interpretación de resultados

La fase de carga es el momento en el cual los datos de la fase anterior (transformación) son cargados en el sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes. En algunas bases de datos se sobrescribe la información antigua con nuevos datos. Los Data Warehouse mantienen un historial de los registros de manera que se pueda hacer una auditoría de los mismos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo.

Existen dos formas básicas de desarrollar el proceso de carga:

- **Acumulación simple:** la acumulación simple es la más sencilla y común, y consiste en realizar un resumen de todas las transacciones comprendidas en el período de tiempo seleccionado y transportar el resultado como una única transacción hacia el data warehouse, almacenando un valor calculado que consistirá típicamente en un sumatorio o un promedio de la magnitud considerada.
- **Rolling:** el proceso de Rolling por su parte, se aplica en los casos en que se opta por mantener varios niveles de granularidad. Para ello se almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones de la unidad de tiempo o diferentes niveles jerárquicos en alguna o varias de las dimensiones de la magnitud almacenada (por ejemplo, totales diarios, totales semanales, totales mensuales, etc.).

La fase de carga interactúa directamente con la base de datos de destino. Al realizar esta operación se aplicarán todas las restricciones y triggers (disparadores) que se hayan definido en ésta (por ejemplo, valores únicos, integridad referencial, campos obligatorios, rangos de valores). Estas restricciones y triggers (si están bien definidos) contribuyen a que se garantice la calidad de los datos en el proceso ETL, y deben ser tenidos en cuenta.

## 2. Procedimientos de ejecución

Recordemos que la extracción de datos en sistemas de almacén de datos pasa por tres fases que son:

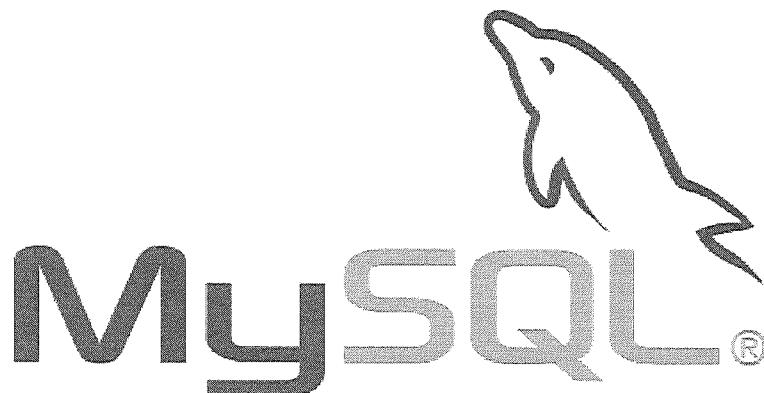
- **Extracción:** consiste en obtener los datos del sistema origen, realizando volcados completos o incrementales. En ocasiones esta etapa suele apoyarse en un almacén intermedio, llamado ODS (Operational Data Store), que actúa como pasarela entre los sistemas fuente y los sistemas destino, y cuyo principal objetivo consiste en evitar la saturación de los servidores funcionales de la organización.
- **Transformación:** los datos procedentes de repositorios digitales distintos no suelen coincidir en formato. Por tanto, para lograr integrarlos resulta imprescindible realizar operaciones de transformación. El objetivo no es otro que evitar duplicidades innecesarias e impedir la generación de islas de datos inconexas. Las transformaciones aplican una serie de reglas de negocio (o funciones) sobre los datos extraídos para convertirlos en datos destino.
- **Carga:** se trata de introducir los datos, ya adaptados al formato deseado, dentro del sistema destino. En algunos casos se sobreescribe la información antigua con la nueva, mientras que en otros se guarda un historial de cambios que permite consultas retrospectivas en el tiempo, así como revertir modificaciones. Para la carga masiva de datos suele ser necesario desactivar temporalmente la integridad referencial de la base de datos destino.

A continuación veremos en concreto como se extraen los datos entre los sistemas ERP, CRM y la base de datos con la ayuda del servidor de bases de datos MySQL.

### 2.1. Definición de procedimientos de extracción de datos entre los sistemas ERP, CRM y la base de datos

Para la extracción de datos entre los sistemas ERP, CRM y la base de datos es recomendable usar MySQL que se trata del servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. MySQL AB

es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL.



MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de datos. La información que puede almacenar una base de datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas, ó tan vasta como la de una tienda en línea, un sistema de noticias, un portal, o la información generada en una red corporativa. Para agregar, acceder, y procesar los datos almacenados en una base de datos, se necesita un sistema de administración de bases de datos, tal como MySQL.

MySQL es un sistema de administración de bases de datos relacionales. Una base de datos relacional almacena los datos en tablas separadas en lugar de poner todos los datos en un solo lugar. Esto agrega velocidad y flexibilidad. Las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando se necesitan consultar datos. La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta", y es el lenguaje más usado y estandarizado para acceder a bases de datos relacionales.

Open Source significa que la persona que quiera puede usar y modificar MySQL. Cualquiera puede descargar el software de MySQL de Internet y usarlo sin pagar por ello. Inclusive, cualquiera que lo necesite puede estudiar el código fuente y cambiarlo de acuerdo a sus necesidades. MySQL usa la licencia GPL (Licencia Pública General GNU), para definir qué es lo que se puede y no se puede hacer con el software para diferentes situaciones. Sin embargo, si uno está incómodo con la licencia GPL o tiene la necesidad de incorporar código de MySQL en una aplicación comercial es posible comprar una versión de MySQL con una licencia comercial. Para mayor información, ver la página oficial de MySQL en la cuál se proporciona mayor información acerca de los tipos de licencias.

El servidor de bases de datos MySQL es muy rápido, seguro, y fácil de usar. Si eso es lo que se está buscando, se le debe dar una oportunidad a MySQL. Se pueden encontrar comparaciones de desempeño con algunos otros manejadores de bases de datos en la página de MySQL.

El servidor MySQL fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, el servidor MySQL ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas Web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones...
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos

guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

## Sentencia SELECT

La sentencia SELECT de MySQL nos permite consultar los datos almacenados en una tabla de la base de datos.

El formato de la sentencia select es:

```

SELECT [ALL | DISTINCT ]
    <nombre_campo> [{,<nombre_campo>}]
FROM <nombre_tabla>|<nombre_vista>
    [{,<nombre_tabla>|<nombre_vista>}]
[WHERE <condicion> [{ AND|OR <condicion>}]]
[GROUP BY <nombre_campo> [{,<nombre_campo>}]]
[HAVING <condicion>[{ AND|OR <condicion>}]]
[ORDER BY <nombre_campo>|<indice_campo> [ASC | DESC]
    [{,<nombre_campo>|<indice_campo> [ASC | DESC ]}]]
```

Veamos por partes que quiere decir cada una de las partes que conforman la sentencia:

- **SELECT:** palabra clave que indica que la sentencia de SQL que queremos ejecutar es de selección.
- **ALL:** indica que queremos seleccionar todos los valores. Es el valor por defecto y no suele especificarse casi nunca.
- **DISTINCT:** Indica que queremos seleccionar sólo los valores distintos.
- **FROM:** indica la tabla (o tablas) desde la que queremos recuperar los datos. En el caso de que exista más de una tabla se denomina a la consulta "consulta combinada" o "join". En las consultas combinadas es necesario aplicar una condición de combinación a través de una cláusula WHERE.
- **WHERE:** especifica una condición que debe cumplirse para que los datos sean devueltos por la consulta. Admite los operadores lógicos AND y OR.
- **GROUP BY:** especifica la agrupación que se da a los datos. Se usa siempre en combinación con funciones agregadas.
- **HAVING:** especifica una condición que debe cumplirse para los datos. Especifica una condición que debe cumplirse para que los datos sean devueltos por la consulta. Su funcionamiento es similar al de WHERE pero aplicado al conjunto de resultados devueltos por la consulta. Debe aplicarse siempre junto a GROUP BY y la condición debe estar referida a los campos contenidos en ella.
- **ORDER BY:** presenta el resultado ordenado por las columnas indicadas. El orden puede expresarse con ASC (orden ascendente) y DESC (orden descendente). El valor predeterminado es ASC.

Para formular una consulta a la tabla tCoches (creada en el capítulo de tablas) y recuperar los campos matricula, marca, modelo, color, numero\_kilometros, num\_plazas debemos ejecutar la siguiente consulta. Los datos serán devueltos ordenados por marca y por modelo en orden ascendente, de menor a mayor.

```
SELECT matricula,  
       marca,  
       modelo,  
       color,  
       numero_kilometros,  
       num_plazas  
  FROM tCoches  
 ORDER BY marca,modelo;
```

La palabra clave **FROM** indica que los datos serán recuperados de la tabla **tCoches**. Podríamos haber especificado más de una tabla, pero esto se verá en el apartado de consultas combinadas.

También podríamos haber simplificado la consulta a través del uso del comodín de campos, el asterisco **"\*"**.

```
SELECT *  
  FROM tCoches  
 ORDER BY marca,modelo;
```

El uso del asterisco indica que queremos que la consulta devuelva todos los campos que hemos visto.

## RECUERDA

- Para poblar el Data Warehouse se deben mover bloques de datos, muchas veces desde diferentes sistemas operativos, estructuras de archivos y bases de datos, mediante procesos programados que se ejecutan frecuentemente fuera del horario de trabajo para no insumir tiempo de procesamiento del hardware de la empresa, entorpeciendo la operatoria de la misma.
- Los procesos que se conectan a los datos de los sistemas fuente, los recodifican, los integran y los cargan dentro del almacén de datos, reciben el nombre de procesos ETL, que son las siglas en inglés de **Extraer, Transformar y Cargar**.
- La primera parte del proceso **ETL** consiste en extraer los datos desde los sistemas de origen. La mayoría de los proyectos de almacenamiento de datos fusionan datos provenientes de diferentes sistemas de origen. Cada sistema separado puede usar una organización diferente de los datos o formatos distintos. Los formatos de las fuentes normalmente se encuentran en bases de datos relacionales o ficheros planos, pero pueden incluir bases de datos no relacionales u otras estructuras diferentes. La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.
- Las funciones básicas a ser realizadas en la fase de **tratamiento de la información** consisten en leer los archivos intermedios generados por la fase de extracción, realizar las transformaciones necesarias, construir los registros en el formato del Data Warehouse y crear un archivo de salida conteniendo todos los registros nuevos a ser cargados en el Data Warehouse.
- La **fase de carga** es el momento en el cual los datos de la fase anterior (transformación) son cargados en el sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes.
- Para la extracción de datos entre los sistemas ERP, CRM y la base de datos es recomendable usar MySQL que se trata del servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo negocio consiste en

proporcionar servicios en torno al servidor de bases de datos MySQL.

- **MySQL** es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de datos. La información que puede almacenar una base de datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas, ó tan vasta como la de una tienda en línea, un sistema de noticias, un portal, o la información generada en una red corporativa. Para agregar, acceder, y procesar los datos almacenados en una base de datos, se necesita un sistema de administración de bases de datos, tal como MySQL.
- La sentencia **SELECT** de MySQL nos permite consultar los datos almacenados en una tabla de la base de datos.

## Preguntas de Autoevaluación

### 1. Los Data Marts:

- a) Contienen información acerca de los datos contenidos en el DW.
- b) Tienen como misión la de mantener, distribuir, aportar seguridad y monitorización.
- c) Son almacenes de datos especializados, diseñados para soportar necesidades de análisis específicas para un único departamento o área funcional de la empresa, por ejemplo marketing, finanzas, producción, etc.

### 2. Los procesos que se conectan a los datos de los sistemas fuente, los recodifican, los integran y los cargan dentro del almacén de datos, reciben el nombre de:

- a) Procesos de desarrollo.
- b) Procesos ETL.
- c) Procesos de extracción.

### 3. Confirma si la siguiente afirmación es verdadera o falsa:

*"La primera parte del proceso ETL consiste en extraer los datos desde los sistemas de origen".*

- a) Verdadero.
- b) Falso.

**4. ¿Qué dos formas básicas existen de desarrollar el proceso de carga?**

- a) Acumulación simple y acumulación doble.
- b) Acumulación simple y rolling.
- c) Acumulación simple y carga.

**5. ¿Qué es MySQL?**

- a) Un sistema de administración de base de datos.
- b) Un sistema complementario al Data Warehouse.
- c) Un software específico para la conexión entre el Data Warehouse y el sistema ERP.

# UD7 Incidencias en el proceso de extracción de datos



**UF1886 Administración  
del Gestor de Datos en  
Sistemas ERP-CRM**



## 1. Trazas de ejecución

La detección de fallos es una de las áreas funcionales de la gestión de redes. Su objetivo fundamental es la localización y solución de los problemas que se presentan en una red, ofreciendo un soporte preventivo y correctivo a su funcionamiento. En muchos casos, permite incluso detectar y subsanar los problemas antes de que se percaten los usuarios. Cuando una falla o funcionamiento raro se detecta en la red se dedica un tiempo a identificar la causa que le dio origen y, en caso de que sea necesario, se toman medidas para evitar su repetición.



Las **trazas de ejecución** de un servicio son una fuente importante de la que se puede extraer gran cantidad de información acerca de su comportamiento. Por esta razón, es usual que los administradores de una red las examinen cuando se están buscando las causas de un error. Sin embargo, ocurre con frecuencia que el cúmulo de datos que se almacena en estos archivos es tan grande que termina por agotar al administrador sin haber encontrado nada útil.

La traza de un algoritmo se puede definir como la ejecución manual de las sentencias que lo componen. Las acciones de un algoritmo no son interpretables por un ordenador; deben traducirse a un lenguaje de programación para su ejecución real en una computadora. El proceso de una traza es similar a la ejecución de un programa en un ordenador. Sobre papel se llevan a cabo las sentencias indicadas en el algoritmo. Estas instrucciones modifican el contenido de

las variables (por ejemplo, una asignación, el cálculo de una expresión, etc.) o producen una interacción con el usuario (las operaciones de lectura/escritura).

Al igual que ocurre en un entorno de programación para expresar y realizar la traza de un algoritmo nos construimos un entorno compuesto por una tabla que analiza las variables, y otra que contiene propiamente la traza. La tabla de análisis de las variables contiene la siguiente información:

- **Variable:** nombre de la variable.
- **Función:** función de esta variable dentro del algoritmo.
- **Valor inicial:** valor de la variable al comienzo del algoritmo.
- **Valor final:** valor final de la variable cuando termina el algoritmo.

La tabla de la traza tiene los siguientes bloques de información:

- **Algoritmo:** en este bloque se visualiza el algoritmo y se indica mediante un puntero la próxima acción a realizar.
- **Bloque de datos:** en este bloque se muestran todas las variables y constantes accesibles desde el algoritmo. Se aprecia la evolución del contenido de las variables a medida que se ejecutan las acciones del algoritmo. Cuando se ejecuta el cuerpo del algoritmo en este bloque aparecen todas las variables y constantes declaradas en el algoritmo. Cuando se ejecuta el cuerpo de un subproblema aparecen las variables de entrada y las locales, así como las variables globales del algoritmo.
- **Terminal:** la ejecución de operaciones de entrada/salida provoca una interacción con el usuario. En este bloque se simula el resultado de la operación de E/S. mostrando los resultados por pantalla o esperando a que el usuario introduzca unos valores por teclado.

La traza de un algoritmo nos permite comprobar que el resultado de un algoritmo se corresponde con la salida esperada.

A modo de ejemplo, consideremos el siguiente fragmento de código en Pascal, el cual calcula la suma de los N primeros números naturales.

```

{1} Readln(N);
{2} Suma := 0;
{3} FOR Contador:=1 TO N DO Suma := Suma + Contador;
{4} Writeln(Suma);

```

La siguiente tabla muestra la traza de este código. Las celdas con sombra representan los estados que no son afectados por una sentencia.

Sentencia	N	Contador	Suma	Entrada	Salida
--	?	?	?	4	--
{1}	4	?	?	--	--
{2}	4	?	0	--	--
{3}	4	1	1	--	--
{3}	4	2	3	--	--
{3}	4	3	6	--	--
{3}	4	4	10	--	--
{4}	4	4	10		10

En síntesis podemos decir que una traza es el seguimiento del estado de la memoria y de la entrada y salida de un programa según se ejecuta el mismo. Generalmente, la traza de un programa se representa con una tabla cuyas columnas representan las variables del programa y cada fila representa el estado global de la memoria en cada paso de ejecución del programa. De este modo, podemos analizar y comprobar el estado anterior y posterior de cada sentencia ejecutada. Las trazas resultan muy útiles en la detección de errores lógicos de un programa ya que permiten localizar las sentencias erróneas.

## 1.1. Establecimiento de trazas para estudiar incidencias

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. Estas «Trazas» se utilizan para comprobar la ejecución de las validaciones de datos previstas.

Las mencionadas trazas no deben modificar en absoluto el Sistema. El auditor informático también emplea la amplia información que proporciona el propio Sistema, como son los ficheros de *Accounting* o de contabilidad, en donde se encuentra la producción completa de aquél, y los log de dicho Sistema, en donde se recogen las modificaciones de datos y se detalla la actividad general.

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 qué tipos carga. Estas actuaciones, en principio útiles, pueden resultar contraproducentes si se traspasan los límites.

No obstante la utilidad de las Trazas, ha de repetirse lo expuesto en la descripción de la auditoría informática de Sistemas: el auditor informático emplea preferentemente la amplia información que proporciona el propio Sistema: así, los ficheros de <*Accounting*> o de <*contabilidad*>, en donde se encuentra la producción completa de aquél, y los <*Log*> de dicho Sistema, en donde se recogen las modificaciones de datos y se pormenoriza la actividad general.

Del mismo modo, el Sistema genera automáticamente exacta información sobre el tratamiento de errores de máquina central, periféricos, etc.

**Log4j** es una biblioteca Open Source desarrollada en Java por la Apache Software Foundation que permite a los desarrolladores de software elegir la salida y el nivel de granularidad de los mensajes o "logs" (data logging) a tiempo de ejecución y no a tiempo de compilación como es comúnmente realizado. La configuración de salida y granularidad de los mensajes es realizada a tiempo de ejecución mediante el uso de archivos de configuración externos. Log4J ha sido implementado en otros lenguajes como: C, C++, C#, Perl, Python, Ruby y Eiffel.

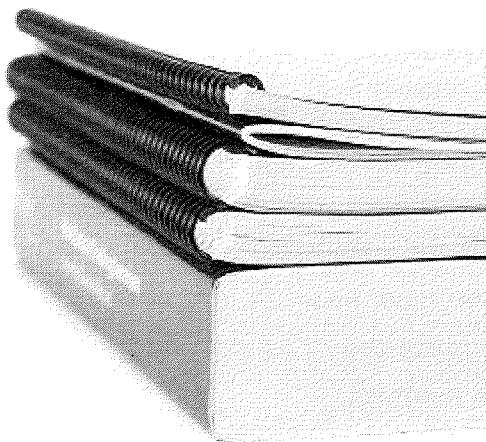
## 2. Resolución de incidencias

Como ya se ha dicho con anterioridad uno de los principales problemas de la programación de aplicaciones, es el tema de migrar de un entorno de desarrollo, a un entorno final y productivo. Es normal que en muchas ocasiones al programador se le pasen ciertos aspectos que se vean reflejados a la hora de ejecutar la aplicación, por ello es necesario seguir un procedimiento estructurado para la resolución de incidencias.

Una incidencia básicamente es un error o problema que aparece cuando no se ejecuta una aplicación determinada. A continuación vamos a ver posibles soluciones a incidencias ocurridas en las trazas, así como la documentación que se debe cumplimentar para dichas incidencias.

### 2.1. Solución y documentación de incidencias

Los procesos realizados, así como las incidencias producidas en la administración, se reflejan en la documentación para mantener registro de seguimiento de los trabajos realizados, siguiendo el procedimiento establecido por la organización.



La documentación técnica específica asociada se interpreta, en su caso, en la lengua extranjera de uso más frecuente en el sector.

Una tarea fundamental en todo proyecto informático es la generación de log o trazas, generalmente para detectar y solucionar los posibles errores que pudieran producirse.

Aunque normalmente estos logs suelen dejarse en archivos de texto, en determinadas situaciones puede ser necesario enterarse rápidamente de los errores sin tener que esperar a que el usuario nos lo notifique. Una posible solución podría ser que el sistema nos enviase esos errores a una cuenta de correo electrónico.

Muchas veces, una aplicación no te funciona correctamente, o la ejecutas y no se muestra en pantalla, o tal vez un dispositivo te da problemas. En el caso de las aplicaciones, una opción es intentar ejecutarla desde la línea de comandos, entonces, generalmente si no funciona, o no puede inicializarse, te imprime una traza de error, y ya te puede poner un poco sobre la pista de lo que está pasando.

En otras ocasiones lo que podemos hacer es intentar examinar los ficheros de trazas del sistema para detectar esos fallos o problemas. En estos ficheros de trazas se va almacenando toda la actividad y eventos que ocurren en nuestro equipo: quien entra, qué comandos ejecuta, qué errores muestran las aplicaciones, etc. Estos ficheros se guardan en: **/var/log**

Para consultarlos necesitamos tener permisos de root, o pertenecer a un grupo con permisos para ver esos ficheros. Alguno de los principales son:

- **Syslog**: el log principal del sistema que guarda mensajes de trazas de demonios y otros programas como cron, init, dhclient, y algunos mensajes relacionados con el kernel.
- **dmesg**: este log muestra todos los mensajes del kernel en tiempo de arranque. También incluye otros mensajes relacionados con la carga de dispositivos y Módulos. Debería ser el primer lugar para consultar si quieras ver qué tipos de dispositivos son detectados por el kernel cuando arranca (boot) así como los errores del kernel cuando carga los diferentes módulos.
- **kern.log y messages**: contiene los mensajes de log del kernel, los mensajes generales y los relacionados con el sistema.
- **auth.log y secure**: mensajes de trazas relacionados con la autentificación en el sistema. Contienen también las trazas de cuando ejecutamos comandos relacionados con root como por ejemplo un apt-get install.

- **cron.log**: los mensajes del dominio croad.
- boot.log: mensajes de trazas del arranque.
- **mysqld.log**: mensajes de trazas de la base de datos MySQL si es instalada.
- **mail.log**: si un servidor de correo es instalado, aquí se dejan sus mensajes de log.
- **apache**: mensajes de trazas del servidor web Apache, si es instalado.
- **Xorg.O.log**: mensajes relacionados con el sistema X y la interfaz gráfica.

Para consultar estos ficheros de manera grafica, podemos usar en Gnome, la utilidad: *Visor del registro del sistema*, que puedes iniciar desde: **Sistema > Administracion > Registro de actividad del Sistema**.

En el desarrollo de cualquier proyecto o en la gestión del soporte en cualquier ámbito de los sistemas de información (tanto si se trata de soporte interno o a clientes), se requiere el uso de herramientas apropiadas que nos permitan la gestión de dicho soporte, permitiéndonos hacer un seguimiento de los procesos, realizar tareas de control o reporting, así como documentar adecuadamente las acciones realizadas.

En el mundo OpenSource, existen multitud de herramientas orientadas a la gestión de incidencias, tickets o bugs. Herramientas que nos pueden servir para la gestión de un Help Desk o como soporte al desarrollo de nuevos proyectos o la gestión de los bugs y problemas detectados en un producto software. Eventum fue considerada la solución desarrollada internamente en el proyecto de MySql y posteriormente fue liberada al público para su uso. Es muy sencilla de utilizar y configurar (PHP+MySql) y puede ser válida para la gestión de soporte y documentación de incidencias en un departamento de Informática de una empresa pequeña o mediana. La herramienta es un poco limitada y su mayor inconveniente es que ha dejado de ser mantenida, por lo que se recomienda buscar otras alternativas.

**List of Issues - Eventum - Mozilla Firefox**

File Edit View Go Bookmarks Tools Help

Eventum (Logout) Administration | Create Issue | List Issues | Advanced Search | My Requirements | Profile | Reports | Help

User: Joseo Prado Mata (Preferences)

Issues: Id # Go Keywords Search

Keywords(s): Assigned: Status: Category: Priority: Search Clear

Advanced Search Saved Searches: Run Search

Search Results (12 issues found, 1 - 10 shown) Hide/Show: [quick search] [advanced search]

All	Priority	Issue ID	Assigned To	Time Spent	Category	Status	Status Change Date	Last Action Date	Summary	Export Data
<input type="checkbox"/>	Critical	2	Joséo Prado Mata	0dh 00m	Bug	Pending	Created: 8d 7h ago	Updated: 8d 0h ago	New email notification options	
<input type="checkbox"/>	High	5	Joséo Prado Mata	0dh 00m	Feature Request	QA	Last Updated: 8d 0h ago	Updated: 8d 0h ago	Move email attachments to separate table	
<input type="checkbox"/>	High	1	Joséo Prado Mata	0dh 00m	Bug	Implementation	Last Updated: 8d 0h ago	Updated: 8d 0h ago	Add more daily time	
<input type="checkbox"/>	High	10	Joséo Prado Mata	0dh 00m	Feature Request	Pending	Created: 8d 0h ago	Created: 8d 0h ago	New "silent" flag to be used when associating emails into issues	
<input type="checkbox"/>	High	8	Bryan Alsdorf	0dh 00m	Feature Request	Pending	Created: 8d 0h ago	Created: 8d 0h ago	Display list of active filters in issue listing page	
<input type="checkbox"/>	Medium	1	Joséo Prado Mata	0dh 00m	Feature Request	QA	Last Updated: 8d 0h ago	Updated: 8d 0h ago	Configuration setting to specify the format of outgoing emails	
<input type="checkbox"/>	Medium	2	Joséo Prado Mata	0dh 00m	Feature Request	Pending	Created: 8d 0h ago	Created: 8d 0h ago	Documentation improvement about advanced search	
<input type="checkbox"/>	Medium	3	Bryan Alsdorf	0dh 00m	Feature Request	Closed	Closed: 8d 0h ago	Closed: 8d 0h ago	Report ideal workload of issues per MySQL server version	
<input type="checkbox"/>	Medium	12	Bryan Alsdorf	0dh 00m	Feature Request	Pending	Created: 8d 0h ago	Created: 8d 0h ago	Ability to sort by the Last Action Date column in the issue listing page	
<input type="checkbox"/>	Medium	5	Bryan Alsdorf	0dh 00m	Feature Request	Implementation	Last Updated: 8d 0h ago	Updated: 8d 0h ago	Option to only display the date part of the created date column	

All Assigned: Go >> Set Rows per Page: 10 Set Hide Closed Issues

Done

Interfaz de usuario de Eventum

## RECUERDA

- Las trazas de ejecución de un servicio son una fuente importante de la que se puede extraer gran cantidad de información acerca de su comportamiento. Por esta razón, es usual que los administradores de una red las examinen cuando se están buscando las causas de un error. Sin embargo, ocurre con frecuencia que el cúmulo de datos que se almacena en estos archivos es tan grande que termina por agotar al administrador sin haber encontrado nada útil.
- La tabla de la traza tiene los siguientes bloques de información: **Algoritmo, Bloque de datos y Terminal.**
- La traza de un algoritmo nos permite comprobar que el resultado de un algoritmo se corresponde con la salida esperada.
- 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. Estas «Trazas» se utilizan para comprobar la ejecución de las validaciones de datos previstas.
- **Log4j** es una biblioteca Open Source desarrollada en Java por la Apache Software Foundation que permite a los desarrolladores de software elegir la salida y el nivel de granularidad de los mensajes o "logs" (data logging) a tiempo de ejecución y no a tiempo de compilación como es comúnmente realizado. La configuración de salida y granularidad de los mensajes es realizada a tiempo de ejecución mediante el uso de archivos de configuración externos. Log4J ha sido implementado en otros lenguajes como: C, C++, C#, Perl, Python, Ruby y Eiffel.
- Los procesos realizados, así como las incidencias producidas en la administración, se reflejan en la documentación para mantener registro de seguimiento de los trabajos realizados, siguiendo el procedimiento establecido por la organización.



## Preguntas de Autoevaluación

### 1. Confirma si la siguiente afirmación es verdadera o falsa:

"Las trazas de ejecución de un servicio son una fuente importante de la que se puede extraer gran cantidad de información acerca de su comportamiento".

- a) Verdadero.
- b) Falso.

### 2. ¿Qué significa el valor **nal**?

- a) Valor de la variable al comienzo del algoritmo.
- b) Valor final de la variable cuando termina el algoritmo.
- c) Función de esta variable dentro del algoritmo

### 3. ¿Qué bloques de información contiene la tabla de la traza?

- a) Algoritmo, Bloque de datos y Terminal.
- b) Datos, Variables y Valor.
- c) Algoritmo, Variable y Función.

### 4. Confirma si la siguiente afirmación es verdadera o falsa:

"La traza de un algoritmo nos permite comprobar que el resultado de un algoritmo se corresponde con la salida esperada".

- a) Verdadero.
- b) Falso.

**5. ¿Qué es Log4j?**

- a) Un sistema de administración de base de datos.
- b) Un sistema complementario al Data Warehouse.
- c) Una biblioteca Open Source desarrollada en Java por la Apache Software Foundation que permite a los desarrolladores de software elegir la salida y el nivel de granularidad de los mensajes o "logs" (data logging) a tiempo de ejecución y no a tiempo de compilación como es comúnmente realizado.

# Actividades Prácticas



**UF1886 Administración  
del Gestor de Datos en  
Sistemas ERP-CRM**



## Actividad Práctica RP3

Para la elaboración de esta actividad práctica se atenderá a lo indicado en el Real Decreto 1531/2011, de 31 de octubre, en lo correspondiente a la UC1214\_3 (Administrar sistemas de planificación de recursos empresariales y de gestión de relaciones con los clientes).

El ejercicio que se expone a continuación se corresponde con la realización profesional RP3 (Administrar el almacén de datos (data warehouse) para mantener y asegurar su funcionamiento, de acuerdo a los criterios establecidos por la organización y a las recomendaciones del fabricante del producto).

### Ejercicio

Se solicitará al alumnado que elabore una definición y una breve explicación del almacén de datos o Data Warehouse, indicando los aspectos que más le llaman la atención del mismo, cuáles considera más complicados, cuáles modificaría y cuáles fomentaría, así como la justificación de todas y cada una de las aportaciones anteriores.

Se va a proporcionar una tabla que puede servir a modo orientativo en la resolución del presente ejercicio.

Se valorará la capacidad de síntesis, concreción y claridad en las indicaciones aportadas. La capacidad de reflexión también se valorará en gran medida.

ALMACÉN DE DATOS	
<b>¿En qué consiste?</b>	
<b>¿Qué es lo que más me llama la atención?</b>	<b>¿Por qué?</b>
<b>¿Qué considero más complicado?</b>	<b>¿Por qué?</b>
<b>¿Qué modificaría?</b>	<b>¿Por qué?</b>
<b>¿Qué fomentaría?</b>	<b>¿Por qué?</b>

# Respuestas a las Preguntas de Autoevaluación

UNIDAD DIDÁCTICA 1		UNIDAD DIDÁCTICA 2		UNIDAD DIDÁCTICA 3	
1	A	1	B	1	C
2	B	2	A	2	A
2	B	3	A y C	3	A
4	C	4	B	4	C
5	A	5	A	5	A

UNIDAD DIDÁCTICA 4		UNIDAD DIDÁCTICA 5		UNIDAD DIDÁCTICA 6	
1	B	1	A	1	C
2	A	2	A	2	B
3	C	3	C	3	A
4	A	4	A	4	B
5	A y B	5	A	5	A

UNIDAD DIDÁCTICA 7	
<b>1</b>	A
<b>2</b>	B
<b>2</b>	A
<b>4</b>	A
<b>5</b>	C