

## UNIDAD I

### Características de archivos

#### Características de los archivos

Las principales características que diferencian a esta estructura de datos de las restantes son las siguientes:

- **Residencia en soporte de información externa:** también denominados memorias secundarias o auxiliares por ejemplo, *cintas y discos*.
- **Independencia respecto de los programas:** significa que la vida del archivo no está limitada por la vida del programa que lo creó y también que en diferentes momentos pueden utilizarse el mismo archivo en distintos programas.
- **Permanencia de las informaciones almacenadas:** es decir toda la información almacenada en la memoria central desaparece cuando se ejecuta el programa que lo maneja, pero para hacer desaparecer un archivo será necesario realizar explícitamente una operación de borrado.
- **Gran capacidad de almacenamiento:** teóricamente esta capacidad es ilimitada, por el contrario las estructuras de datos que residen en la memoria central tiene limitado su tamaño por la capacidad de este.

#### Clasificación de los archivos según su uso

Los archivos se clasifican según la utilización que se hace de ellos en tres grupos:

**Archivos permanentes:** contienen información que varía poco a lo largo del tiempo.

Pueden ser de tres clases:

Archivos constantes: su información permanece prácticamente inamovible utilizándose principalmente como archivos de consulta. Un archivo de este tipo puede ser la red del metro de una ciudad, que contiene la descripción, la característica, número de estaciones, número de trenes etc., de cada día.

Archivos de situación o maestros: son colecciones de registros que se refieren a eventos que afectan a una organización. Los datos pueden constituir una revisión histórica de eventos, como el caso de un archivo del historial de ventas o del historial de proveedores, o pueden ser archivos orientados a eventos presentes, tales como los archivos sobre el estado del inventario, o bien archivos de cuentas por cobrar. En cada uno de los casos, los archivos contienen los datos que son primordiales para la operación continua de la organización. En consecuencia, deben ser mantenidos constantemente (es decir, cambios con objeto de incluir datos referentes a nuevos eventos o para corregir errores) a fin de asegurarse que sean a la vez precisos y actualizados. Un archivo maestro que no contenga datos actualizados, a menos de que se conserve estrictamente para fines históricos o para conservación usual, es de muy pequeño o de ningún valor para una organización. Los archivos con el historial de las ventas o de las cuentas por cobrar pueden, por ejemplo, actualizarse diariamente para incluir los nuevos datos sobre las mercancías que se han retirado del inventario, para cargar o acreditar ventas que se han efectuado durante el día.

Estos archivos se utilizan periódicamente para adaptarlos a cada nueva situación.

Archivos históricos: se obtienen de los anteriores cuando se dejan fuera de uso para futuros estudios estadísticos o consulta. Será un archivo histórico el que contiene por ejemplo la información de libros adquiridos por una biblioteca en la década del 80.

**Archivos de movimiento o transacciones:** estos son relativamente temporales, ya que contienen datos sobre las transacciones de las actividades de la empresa. Estos pueden establecerse con objeto de capturar datos sobre transacciones de ventas (tales como cantidades de artículos vendidos a un solo cliente), recibos de ventas al contado, y así sucesivamente. Estos archivos son utilizados principalmente para registrar los datos en el archivo maestro, de la misma manera que un contador registra los datos en las cuentas respectivas. En otras palabras estos, que también se llaman archivos de detalle debido a que contienen los detalles de las transacciones, se procesan contra (utilizado como datos para actualización) el archivo maestro.

Una tienda de menudeo, por ejemplo, normalmente realiza ventas a crédito. El monto de estas ventas, así como ciertos datos descriptivos referentes a las transacciones, deben ser recopilados y utilizados para actualizar los registros de cada cliente que ha hecho una compra. Supongamos que los siguientes datos se han recolectado en un volante de ventas para cada una de las transacciones: (1) nombre del cliente, (2) número de cuenta del cliente, (3) fecha de la transacción, (4) descripción y precio de cada artículo comprado, y (5) importe total de la venta. Al final de cada semana, la tienda reúne todo los datos en el archivo de transacciones de la siguiente manera. Primeramente los volantes de las ventas son agrupados y puestos en un lote o expediente. Luego se introducen los datos a la computadora y los volantes se separan y guardan en un gabinete metálico para el caso de que sean necesarios más tarde. La computadora registra un archivo de transacciones de todas las ventas semanales en una cinta magnética.

Este proceso genera un archivo de transacciones que contiene todos los datos relevantes de cada venta a crédito. Este archivo es posteriormente procesado contra el archivo maestro de las cuentas por cobrar; esto es, todas las transacciones de las nuevas ventas son registradas o introducidas en el archivo maestro. Un programa de cómputo lee cada registro de las transacciones, encuentra el correspondiente registro del cliente en el archivo maestro y el importe de la compra se agrega al saldo existente. Este proceso, mostrado en la forma de un diagrama de flujo en la siguiente figura se repite para cada transacción en el archivo de transacciones.

## GRÁFICO

Los archivos de transacciones también pueden contener datos que han de ser procesados contra el archivo maestro para cambiar datos que han sido registrados previamente. Los datos pueden ser modificados porque se hayan detectado errores o debido a que haya que agregar nuevos registros o eliminar algunos ya existentes.

Se dice que los archivos de transacciones son temporales comparados con los archivos maestros. Aunque frecuentemente, es necesario referirse a estos archivos para verificar o confirmar detalles de transacciones en particular o para corregir errores que no fueron detectados previamente. Por lo tanto, aunque un archivo de transacciones es temporal, se conserva a menudo (fuera de línea de los dispositivos de almacenamiento magnético) durante varios meses antes de que sea destruido o eliminado.

Sus registros denominados movimientos o transacciones son de tres clases:

- Altas
- Bajas
- Modificaciones

Una vez realizado el proceso de actualización de un archivo maestro por medio de un archivo de movimiento, este pierde su validez y podemos deshacernos de él.

**Archivo de trabajo o maniobra:** de la misma manera que el archivo de transacciones, estos (también llamados de *clasificación*) son temporales, el tiempo de retención es simplemente el tiempo destinado para correr un programa. Su único objeto es ayudar en la selección o clasificación de conjuntos de datos con un

orden particular con base en un registro clave. Una clave es un campo o elemento de datos en un registro que se utiliza para identificar al registro dentro de un archivo. Por ejemplo, puede desearse disponer un archivo de empleados por orden alfabético según los apellidos. En este caso, la clave es el apellido y el valor para el elemento de datos se utiliza para su clasificación. Si, por lo contrario, se desea clasificar el archivo por un orden basado en los números de identificación, entonces el número identificado será la clave del registro.

Estos archivos posteriormente se utilizarán para crear nuevos archivos con diferentes secuencias. Se procesan utilizando un programa de clasificación (un programa de computación que ejecuta la operación de clasificar) para ordenar los datos en una secuencia seleccionada, tal como por un orden alfabético o numérico. La clasificación se efectúa con base en el valor de la clave del registro que especifica el usuario.

Los archivos de clasificación pueden ser copias de los archivos maestros, conjuntos de transacciones, o datos originales de cualquiera de ellos. En el caso de los archivos de transacciones, un archivo de clasificación normalmente se crea para colocar las transacciones en el mismo orden en que se tienen los registros en el archivo maestro, de manera que la actualización pueda ser ejecutada en forma más eficiente. Los archivos de clasificación también pueden ser utilizados en la preparación de los reportes con base en los datos de archivo maestro o de archivo de transacciones. El archivo de transacciones normalmente no se conserva una vez que los reportes han sido terminados.

### Clasificación de los archivos según su organización

Al diseñar un archivo dependiendo del uso que se va a hacer del mismo y del soporte utilizado, se pueden elegir distintas maneras de organizar sus registros, siendo las siguientes las principales organizaciones:

- **Secuencial /seriada:**

En una organización **serial**, el tipo más básico de organización de archivos o conjunto de datos, los registros se almacenan en ubicaciones adyacentes físicamente en el medio de almacenamiento (por ejemplo cinta magnética, disco o tambor magnéticos) sin considerar algún orden o secuencia particular de los registros. Sólo se considera el almacenamiento de cada registro en la siguiente posición disponible. En estos casos no importa cuál es el contenido de un registro o los otros registros que lo preceden. Para leer un registro almacenado bajo la organización serial, el archivo debe ser procesado desde el principio hasta encontrar el registro que se busca. Debido a que esto es una manera incierta de organización de los archivos, el método serial raramente es utilizado para almacenar los registros en archivos permanentes. Sin embargo, a menudo se utiliza para los datos de transacciones que se registran tan pronto como ocurre un evento. Por ejemplo, si los datos de ventas se almacenan en una cinta magnética utilizando un dispositivo de tecla a cinta a medida que ocurren las transacciones, estos archivos están organizados en forma serial. Puesto que los registros se hallan escritos uno a continuación del otro, no importa que no estén organizados alfabéticamente por el nombre del cliente o por el nombre del artículo vendido.

En la organización **secuencial**, los registros se escriben en la misma secuencia en que se recolectan, pero están organizados con un cierto orden. Esto significa, y es la principal diferencia entre la organización serial y la secuencial, que si se requiere un orden en particular los registros de datos deben ser ordenados antes de que se cree el archivo de nómina en una cinta magnética a partir de documentos fuente, y se desea una clasificación alfabética por apellido, los registros deben ser puestos en un orden alfabético antes de que sean enviados a la cinta magnética. Después de que haya terminado la clasificación, los datos se registran y almacenan en el orden en que se graban. En otras palabras, el primer registro procesado es el primer registro escrito en la primera posición de la cinta y así sucesivamente.

Cuando se procesan archivos organizados en forma secuencial, los registros deben ser leídos en el mismo orden en que se encuentran almacenados, comenzando con el principio del archivo. El procesador lee el archivo desde el principio, verificando cada registro individual hasta encontrar el deseado. Por lo tanto, si el

registro particular que se desea procesar se halla en el lugar 101 del archivo le precederán 100 archivos que deben ser leídos antes que él. La organización secuencial puede ser utilizada en todo los medios de almacenamiento, pero es el método dominante para las cintas magnéticas.

En un archivo secuencial no se pueden hacer operaciones de escritura cuando se está leyendo, ni operaciones de lectura cuando se está escribiendo, por otro lado para actualizar es necesario crear nuevos archivos donde se copien los registros que vayan a permanecer modificado o no, junto con los nuevos

El método usual de poner al día a tales archivos, es por medio de un copiado de todo el archivo desde un dispositivo a otro. Los datos de transacciones son almacenados en la misma secuencia que el archivo maestro, los registros son cotejados y subsecuentemente fusionados para crear una nueva versión actualizada. Las copias viejas o anteriores, son retenidas para usarlas en la eventualidad de una corrupción del archivo o una caída del sistema. Los archivos secuenciales son apropiados sólo para procesamiento en lote y no se pueden usar en sistemas en línea. Esta técnica de organización de archivos, también incurre en el desaprovechamiento del sistema en términos de información indeseable, se tiene que copiar desde un dispositivo a otro. Su principal ventaja recae en el hecho, de que tales archivos pueden almacenarse sobre cintas magnéticas relativamente baratas;

gráficos de archivos serial y secuencial (no importan)

Archivo secuencial

- **Directa o aleatoria:** (también denominada relativa)

Una alternativa a la organización secuencial es la organización **aleatoria** llamada a veces también organización de direccionamiento directo. Con este método, que puede ser utilizado con los dispositivos de almacenamiento directo como los discos y los tambores magnéticos, los registros no requieren estar escritos o tener acceso a ellos en forma secuencial, pero pueden ser manejados al azar o en forma aleatoria. Cada posición del medio de almacenamiento tiene una dirección, y cuando el registro se procesa puede ser asignado a cualquier posición de almacenamiento en el dispositivo, sin considerar si las posiciones precedentes ya han sido utilizadas. En otras palabras al crear el archivo de nómina que hemos discutido anteriormente, las transacciones pueden ser procesadas en cualquier orden, y escritas en muchas posiciones a través del almacenamiento. No tienen que estar escritas siguiendo un orden o secuencia.

Para acceder a un registro almacenado utilizando una organización aleatoria, los registros anteriores no necesitan ser examinados antes que tal registro. La CPU puede ir directamente al registro que se desea sin haber buscado en los anteriores del archivo. Obviamente, por tanto cuando se tiene que acceder a un registro, la organización aleatoria es mucho más rápida que la secuencial. Sin embargo, cuando hay un gran número de registros que deben ser localizados y leídos, y las transacciones pueden ser clasificadas en el mismo orden que el archivo maestro, la organización secuencial puede ser mucho más eficiente.

Los archivos aleatorios, utilizan el espacio de almacenamiento disponible muy eficientemente y son apropiados para sistemas en línea, en los cuales el tiempo de acceso a los registros es un factor importante.

- **Secuencial indexada:** un archivo con esta organización consta de tres áreas:
  - **Area de índice:** es un archivo secuencial creado por el sistema en el que cada registro establece una división (segmento) en el área primaria y contiene la dirección de comienzo de segmento y la clave mas alta del mismo, de esta manera el sistema accede de forma directa a un segmento del área primaria a partir del área de índices de forma similar a la búsqueda de un capítulo de un libro en su índice.
  - **Area primaria:** contendrá los registros de datos ordenados en forma ascendente por su campo clave.

- **Área de excedente:** es aquella en la que estará todo aquel registro que sobrepasa la extensión del archivo o sea que este presenta una necesidad de espacio adicional.

Un uso común de los dispositivos de direccionamiento directo para organización de archivos implica formar y mantener índices para el almacenamiento secuencial y no secuencial.

El índice que se emplea en las técnicas de organización de archivos es comparable al índice que figura en algunos libros: tiene una lista de claves de registros y direcciones de datos relacionados. En un libro, el identificador o clave es un tema o nombre, y la ubicación es un número de página. Los índices de los archivos consisten en una clave de registros y una dirección de almacenamiento. Para localizar un registro, se busca en el índice hasta hallar la clave apropiada del registro. La dirección correspondiente a la clave es donde se encuentra almacenado el registro. El sistema de organización indexada (o con índice) economiza espacio asignado pero no utilizado en el medio de almacenamiento. Puesto que el índice mantiene la localización de la ubicaciones de almacenamiento únicamente los espacios de almacenamiento realmente utilizados son los que necesitan ser asignados. Esto contrasta con los métodos de almacenamiento directo y de distribución donde el espacio se asigna para un uso posterior.

Los índices se utilizan en las organizaciones secuencial y no secuencial con índices. El método secuencial indexado combina una eficiente capacidad de lotes (es decir, un procesamiento secuencial de los archivos) con una operación de búsqueda que es eficiente, ya que los registros pueden ser accedidos en forma aleatoria en caso necesario. El archivo se crea secuencialmente utilizando las claves de los registros para colocar al archivo en orden, ya sea en forma alfabética o numérica con los números de identificación en forma ascendente. Las secciones de almacenamiento físicas llamadas bloques, se indexan (o indizan) luego.

En la organización secuencial indexada, el índice direcciona a un registro específico en el archivo. Sin embargo, la dirección en el índice normalmente no apunta directamente a la ubicación del registro (si bien esto varía con las implementaciones de los diferentes proveedores) si no al bloque en el cual se encuentra almacenado. La clave del último registro en el bloque (y por lo tanto, el valor más alto de la clave en él) aparece en el índice junto con la dirección del bloque que lo contiene. Si se quisiera encontrar un registro que tuviera un valor de clave de 0242, se haría que el procesador buscara al índice para hallar la primera entrada que tuviera un valor mayor que o igual a 0242. La dirección que correspondiera a esa clave sería el número del bloque que contiene el registro.

En archivos particularmente grandes, el índice puede llegar a ser demasiado largo para ser buscado con eficiencia. En estos casos se requiere formar un índice de los índices. El índice maestro, como se lo llama, contiene la dirección más alta listada en un subíndice en particular. Tales subíndices pueden ser creados para cada uno de los cilindros. Los subíndices a su vez pueden apuntar a índices de más bajo nivel que pertenezcan a un bloque único en el disco. Si un archivo está almacenado en diez cilindros y cada uno de estos contiene veinte bloques, pueden tenerse tres diferentes niveles de índice: un índice maestro que indica el cilindro donde se encuentra almacenado el registro, un índice de cilindros que señala el grupo de bloques en los cuales el registro ha sido almacenado, y un índice de los bloques que indica el bloque en particular en el cual está el registro. Por lo tanto, para encontrar un registro en particular, se debe efectuar una búsqueda primeramente en el índice maestro; que señalará al siguiente índice de nivel inferior, y así sucesivamente.

El método no secuencial indexado implica el uso de un índice para localizar los registros, pero aquí los registros de los datos no están almacenados de acuerdo con ningún orden o secuencia. (puesto que no hay un ordenamiento en estos registros, a esta clase de organización también se le llama un índice aleatorio). En consecuencia, a los registros no puede accederse sin el uso del índice. Para facilitar el procesamiento de los archivos organizados según el método no secuencial de índices, es necesario un índice mucho mayor, ya que se requiere una entrada para cada uno de los registros del archivo. Las inserciones son fáciles: un nuevo registro simplemente se agrega al final del archivo y se añade al índice. Las supresiones también son fáciles: una entrada se elimina del índice sin dificultad. Puesto que hay más entradas de índices, se requiere de un

mayor mantenimiento para ajustar el índice, especialmente si se utilizan múltiples niveles. Los archivos no secuenciales con índices pueden ser procesados en forma secuencial, pero esto es un proceso muy lento ya los índices se deben analizar primeramente para cada uno de los registros.

### Modos de acceso

Sea un soporte de información que contiene un archivo se denominan modos de acceso a la forma en que el dispositivo que maneja el soporte se posiciona en un lugar del mismo para realizar una operación de lectura o escritura de un registro. El modo de acceso lo decide el programador de la aplicación en función del soporte utilizado y del tipo de organización. Hay dos modos básicos:

- **Secuencial:** supone acceder inicialmente al primer registro del archivo y después consecutivamente a todos los sucesivos, hasta llegar al registro deseado. Este modo de acceso se puede utilizar con cualquier soporte y organización.
- **Directo:** solamente se puede realizar en los denominados soportes direccionables como los discos magnéticos y consiste en el posicionamiento sobre cualquier registro sin necesidad de haber accedido antes a la anterior.

<b>directa contra secuencial</b>		
	<b>archivos de acceso directo</b>	<b>archivos de acceso secuencial</b>
<b>pros</b>	<ul style="list-style-type: none"> <li>• Rápido acceso a registros determinados</li> </ul>	<ul style="list-style-type: none"> <li>• Conservan espacio</li> <li>• Disponibles para sistemas basados en cintas</li> </ul>
<b>contras</b>	<ul style="list-style-type: none"> <li>• Desaprovechan espacio</li> <li>• Necesitan discos</li> </ul>	<ul style="list-style-type: none"> <li>• Lentos y engorrosos</li> </ul>
<b>aplicaciones adecuadas</b>	<ul style="list-style-type: none"> <li>• Datos predecibles que estén en un formato definido</li> <li>• Cuando se accede a un pequeño número de registros diferentes; por ejemplo, en una biblioteca donde los clientes solicitan detalles de libros determinados</li> </ul>	<ul style="list-style-type: none"> <li>• Grandes cantidades de datos sin estructurar</li> <li>• Cuando la mayoría de los registros de un archivo se procesan en una única ejecución del programa; por ejemplo, en un sistema de salarios, donde se debe pagar a cada empleado</li> </ul>

## UNIDAD II

### ¿POR QUÉ ES INTERESANTE EL ANALISIS DE SISTEMAS?

Es más interesante que la programación de computadoras porque involucra el estudio de las interacciones entre personas, grupos heterogéneos de personas, computadoras y organizaciones.

### SISTEMAS

Sistema: grupo de elementos interdependientes o que interactúan regularmente formando un todo como:

- – un grupo de cuerpos que interactúan bajo la influencia de fuerzas relacionales

- un conjunto de sustancias que tienden al equilibrio
  - – un grupo de órganos del cuerpo que juntos llevan a cabo una o más funciones
- el cuerpo considerado como una unidad funcional
  - un grupo de fuerzas u objetos naturales
  - un grupo de aparatos o una organización que forma una red, especial para distribuir algo o para servir a un propósito común, teléfono, calefacción, autopsia o procesamiento de datos.

### TIPOS COMUNES DE SISTEMAS

Como podemos ver de las definiciones anteriores existen tipos de sistemas, de hecho casi todo aquello con lo cual entramos en contacto durante nuestra vida cotidiana es un sistema o bien parte de un sistema.

Dado que nuestro objetivo son los sistemas de información empezaremos por dividirlos en dos categorías:

- **SISTEMAS NATURALES:** la gran mayoría de los sistemas no están hechos por el hombre sino que son naturales y sirven a sus propios fines. Es conveniente dividir los sistemas naturales en dos subcategorías:
  - **sistemas físicos:**
    - sistemas estelares: galaxias, sist.solares, etc.
    - sistemas geológicos: ríos, cordilleras, etc.
    - sistemas moleculares: org. complejas de átomos

También hay que tener en cuenta sistemas físicos que suelen ser importantes para modelar otro sistema.

- **sistemas vivientes:** comprenden toda la gama de animales y plantas que nos rodean, al igual que la raza humana.

Algunas de sus propiedades y características familiares pueden utilizarse para ayudar a ilustrar y entender mejor los sistemas hechos por el hombre.

En algunos casos, diseñan sistemas automatizados para reemplazar a sistemas vivos; y en otros, los investigadores consideran a los sistemas vivos como componentes de sistemas automatizados. Los sistemas vivientes y los sistemas hechos por el hombre a menudo forman parte de un metasistema mayor, y entre más entendamos acerca de ambos, mejor analista de sistemas seremos.

- **SISTEMAS HECHOS POR EL HOMBRE:** Un buen número de sistemas son construidos, organizados y mantenidos por humanos e incluyen (o podemos clasificar en ):
  - sistemas sociales: organización de leyes, doctrinas
  - una colección organizada y disciplinada de ideas: sistema decimal, biblioteca
  - sistemas de transporte: redes de carreteras, canales, aerosillas, etc.
  - sistema de comunicación: teléfono, telex, fax
  - sistema de manufacturas: fábricas
  - sistema financiero: contabilidad, bolsa de valores, etc.

En la actualidad la mayoría de estos sistemas incluye las computadoras. De hecho muchos no podrían sobrevivir sin ellas, sin embargo es importante señalar que dichos sistemas existían antes de que hubieran computadoras, de hecho algunos sistemas continúan por completo sin computarizar, otros contienen a la computadora como componente.

Como analista de sistemas, usted naturalmente supondrá que todo sistema con el que se encuentre deberá computarizarse, y el cliente o usuario con quien usted interactúa generalmente supondrá tal predisposición. En la mayoría de los casos podremos determinar si tiene sentido utilizar una computadora para llevar a cabo las funciones del sistema sólo tras haber modelado sus comportamiento esencial.

### ¿POR QUÉ NO DIBIERAN COMPUTALIZAR ALGUNOS SISTEMAS DE INFORMACIÓN?

Puede haber muchas razones, las más comunes son:

- **costos:** puede ser más barato llevando a cabo las funciones y almacenando la información en forma manual.
- **conveniencia:** un sistema automatizado puede ocupar demasiado espacio o bien consumir demasiada electricidad, o bien ser una molestia.
- **seguridad:** si el sistema de información mantiene datos confidenciales, el usuario pudiera no creer que el sistema sea lo suficientemente seguro, tal vez desea mantener la información físicamente protegida y bajo llave.
- **facilidad de mantenimiento:** el usuario puede argumentar que un sistema de información computarizado sería costoso, excepto por el hecho de que no hay ningún miembro del personal que puede encargarse del mantenimiento de hardware y/o software de manera que nadie podría reparar el sistema si este sufriera un desperfecto, ni habría quién pudiera efectuar cambios o mejoras.
- **políticas:** la comunidad usuaria pudiera recelar que las computadoras amenazan con privarla de sus empleos o con volver mecánicos sus trabajos, pero dado que se trata del sistema del usuario sus recelos son de primera importancia. Si no desean tener un sistema automatizado harán todo lo posible para que falle.

SISTEMAS AUTOMATIZADOS: sist. hechos por el hombre que interactúan con o son controlados por una o más computadoras, Aunque hay diferentes todos tienden a tener componentes en común:

- hardware
- software
- personas: los que operan el sist., los que proveen material de entrada o consumen, etc.
- datos: información que el sistema recuerda durante un período.
- procedimientos: las políticas formales e instrucciones de operaciones del sistema.

Una manera de ordenar por categorías los sistemas automatizados es por su *aplicación*: sistemas de manufactura, sistemas de contabilidad, sistemas de defensa militar, etc. aunque esta técnica no resulta muy útil.

Una división en categorías más útiles de los sistemas automatizados es la siguiente:

- **sistema en línea:** es aquel que acepta material de entrada directamente del área donde se creó. También es el sistema en el que el material de salida o el resultado se devuelve directamente a donde es requerido. Una característica común de los sistemas en línea es que entran datos a la computadora o se les recibe de ella en forma remota, otra característica es que los datos almacenados se organizan de tal manera que los componentes individuales de información puedan ser recuperados modificados o ambas cosas, **rápidamente y sin tener necesariamente que efectuar accesos a otros componentes de información del sistema.** Esto contrasta con los sistemas en lotes o batch, la información suele recuperarse de una manera secuencial, lo cual significa que el sistema lee todos los registros de la base de datos, procesando y actualizando aquellos par los cuales haya actividad.
- **sistema de tiempo real:** puede definirse a aquel que controla un ambiente recibiendo datos, procesándolos y devolviéndolos con la suficiente rapidez como para influir en dichos ambientes en ese momento.



con la suficiente rapidez, esto es algo Característico de los siguientes sistemas:

- **Sistema de control de proceso:** se utiliza para verificar y controlar refinerías, procesos químicos y operaciones de maquinado.
- **Sistema de cajeros automáticos:** para depósitos y retiros sencillos.
- **Sistemas de alta velocidad para adquisición de datos:** obtienen datos de telemetría a altas velocidades, satélites en órbita, o computadoras que capturan cantidades enormes de datos de experimentos de laboratorio.
- **Sistemas de guías de proyectiles:** los sistemas computacionales que deben rastrear la trayectoria de un proyectil y hacer ajustes continuos a la orientación y empuje de los propulsores.
- **Sistemas de conmutación telefónica:** son sistemas computacionales que controlan la transmisión de voz y datos en miles de llamadas telefónicas detectando los números marcados, condiciones de ocupado y todas las demás condiciones de una red telefónica típica.
- **Sistemas de vigilancia:** Sistemas computacionales que detectan los signos vitales de diversos pacientes por ejemplo: temperatura y pulso, y son capaces ya sea de ajustar el medicamento administrado o de hacer sonar la alarma si los signos vitales se mantienen fuera de ciertos límites predeterminados.

Otra característica que diferencia a los sistemas de tiempo real de sistemas en línea es que estos últimos suelen interactuar con las personas, mientras que los sistemas de tiempo real usualmente interactúan tanto con personas como con un ambiente que generalmente es autónomo y a menudo hostil.

Los sistemas de tiempo real se caracterizan por lo siguiente:

- Simultáneamente se lleva a cabo el proceso de múltiples actividades.
- Se asignan prioridades diferentes a diferentes procesos, algunos requieren servicio inmediato mientras otros se pueden aplazar por periodos razonables.
- Se interrumpe una tarea antes de concluir, para comenzar otra de mayor prioridad.
- Existe gran comunicación entre tareas, dado que muchas tratan diferentes aspectos de un proceso general.
- Existe acceso simultáneo a datos comunes, tanto en memoria como en el almacenamiento secundario, por lo cual se requiere de un elaborado proceso de sincronización y semáforos para asegurar que los datos comunes no corrompan.
- Existe un uso y asignación dinámicos de memoria RAM en el sistema computacional, dado que a menudo resulta poco económico asignar suficiente memoria fija para manejar situaciones pico de alto volumen.

– **Sistema de apoyo a decisiones y sistemas de planeación estratégica:** son sistemas operacionales que ayudan a llevar a cabo los detalles del trabajo cotidiano de una organización, también se conocen como sistemas de procesamiento de transacciones.

Los sistemas **de apoyo a la toma de decisiones**, como implica el término, estos sistemas computacionales no toman decisiones por sí mismos, sino ayudan a los administradores y a otros profesionistas trabajadores del conocimiento de una organización a tomar decisiones inteligentes y documentadas acerca de los diversos aspectos de la operación.

No operan en forma regular, más bien, se utilizan de manera ad hoc, cuando se les necesita. Ej: programas de hoja de cálculo, sist. de análisis estadístico, prog. de pronóstico de mercado, etc. No solo recuperan y exhiben los datos, sino que también realizan varios tipos de análisis matemáticos y estadísticos de los mismos.

Los sistemas de **planeación estratégica** son utilizados por los gerentes en jefe para evaluar y analizar la misión de la organización. Ofrecen consejos más amplios y generales de la naturaleza del mercado, las preferencias de los consumidores, el comportamiento de la competencia, etc.

Los **sistemas operacionales** representan la base sobre la cual se cimentan los sistemas de apoyo a decisiones y de planeación estratégica. Los sistemas operacionales crean los datos requeridos por los sistemas de nivel superior y continúan actualizando los datos de una manera continua.

- **Sistemas basados en el conocimiento:** son los sistemas expertos, dichos sistemas se asocian con el campo de la inteligencia artificial, la cual podemos definir como: la meta es producir programas capaces de imitar el desempeño humano en una gran variedad de tareas inteligentes.

### **Jerarquía De Los Sistemas De Proceso De Información:**

**Sistemas De**

**Planificación Estratégica**

**Sistemas De**

**Apoyo A Las Decisiones**

**Sistemas Operacionales**

### **Principio de sistemas generales:**

Existen principios generales que son de interés particular para quienes creen sistemas de información e incluyen los siguientes puntos:

- Entre más especializado sea el sistema, menos capaz es de adaptarse a circunstancias diferentes. Entre más general sea un sistema, menos óptimo será para una situación determinada, pero entre más óptimo sea, para tal situación menos adaptable será a nuevas circunstancias.
- Cuanto mayor sea el sistema, mayor es el número de sus recursos que deben dedicarse a su mantenimiento diario.
- Los sistemas siempre forman parte de sistemas mayores y siempre pueden dividirse en sistemas menores. Este punto, es importante por dos razones:
- Sugiere una forma obvia de organizar un sistema que estamos tratando de desarrollar por el procedimiento de dividirlo en sistemas menores.
- Sugiere que la definición del sistema que estamos tratando es arbitraria. Escoger lo que deberá abarcar un sistema y definirlo cuidadosamente para que todo el mundo conozca su contenido, es una de las actividades más importantes al definir el sistema.
- Los sistemas crecen, esto pudiera no ser verdad para todos los sistemas, pero muchos sistemas sí crecen y resulta importante reconocerlo, pues a menudo omitimos tomar esto en cuenta al comenzar el sistema

### **Los participantes en el juego de los sistemas**

Ellos son o los principales participantes que encontraremos en un proyecto de desarrollo de sistemas: usuarios, administradores, auditores y personal de control de calidad y verificación de normas, diseñadores de sistemas, programadores y operadores.

**Usuarios**

Es el participante más importante y se conoce como usuario. Éste es aquel o aquellos para quien se construye el sistema. Es la persona a la que se tendrá que entrevistar a menudo a fin de conocer que deberá tener el nuevo sistema para que éste sea exitoso.

Debe hacerse notar que el término USUARIO **NO** se refiere así mismo. En algunas organizaciones se quiere evitar ese problema y se utiliza el término CLIENTE o dueño para identificarlo.

Generalmente es fácil identificar al usuario, pero hay gran número de situaciones en las que no se conoce la identidad del verdadero usuario o bien en las que hay poca oportunidad de que éste interactúe con el analista. Por esto hay una gran posibilidad de malos entendidos.

De la relación con los usuarios o clientes, podemos sacar dos conclusiones importantes.

- Siempre que sea posible, el analista deberá tratar de establecer contacto directo con el usuario. Aún si se encuentran involucradas otras personas como intermediarios. Para esto es importante tener reuniones de trabajo con los clientes. quien es la persona que recibirá el sistema.
- Si no es posible comunicarse directamente con el cliente o usuario, la documentación generada por el analista se vuelve más importante, porque ésta tendrá que llegar en forma explicativa a los clientes, o sea para describir el comportamiento de un sistema de manera formal y rigurosa.

#### ¿Cómo los podemos clasificar a los usuarios?

- por categoría de trabajo o nivel de supervisión
- por nivel de experiencia en el procesamiento de datos *o la interpretación y seguimiento de un sistema informático.*

#### Clasificación de los usuarios por categoría de trabajo o nivel operacional

- **Usuarios operacionales:** Son oficinistas, administradores y operadores que son los que más probablemente tendrán contacto diario con el nuevo sistema.

Cuando se trabaja con usuarios de este nivel hay que tener en cuenta tres cosas: a) los usuarios de este nivel se preocupan mucho por las funciones que tendrá el sistema, pero es más probable aún que se preocupen por los detalles de interfaz humana (teclado, forma en que aparecen las cosas en la pantalla, etc.). b) tienden a poseer un programa local del sistema, por lo general son conocedores del trabajo específico que hacen y de las personas con las que tienen comunicación inmediata, a menudo no están familiarizados con el panorama general, rara vez se debe a torpeza sino a que no tienen interés en el asunto. *c) suelen pensar en los sistemas en términos físicos, es decir, en términos de la tecnología a utilizar o en términos de la tecnología que imagina que pudiera utilizarse.*

- **Usuarios supervisores:** Son empleados supervisores, administran a un grupo de usuarios operacionales y son responsables de sus logros. Lo importante sobre estos usuarios es que: a) son usuarios operacionales que han sido promovidos, por lo que usualmente están familiarizados con el trabajo de sus subordinados por lo que se puede suponer que estarán de acuerdo con sus necesidades, preocupaciones y perspectivas. b) una razón por la cual no hubiera comunicación entre el usuario supervisor y el operacional es porque el primero debe regirse por un presupuesto. *c) por lo general es el que ve al nuevo sistema como una forma de reducir el número de usuarios operacionales (por despido o arrepentimiento) o de evitar que aumente su número al crecer el volumen de trabajo.* d) a menudo actúa como intermediario entre el analista y los usuarios operacionales, arguyendo que están demasiado ocupados para perder el tiempo hablando con el analista. e) a menudo piensa en los mismos términos físicos que el operacional y su perspectiva es tan local como la de este último. *e) será este usuario con el cual el analista tendrá el contacto primario, es el que definirá los requerimientos reales y las necesidades.*

- **Usuarios ejecutivos:** generalmente no se involucran directamente en el proyecto, sino que sacan provecho de la resultante, pero es muy importante el apoyo logístico para que éste sistema se pueda desarrollar. Lo importante sobre el usuario ejecutivo es: a) pueden proporcionar la iniciativa para el proyecto. b) no fueron previamente operacionales o si lo fueron, habrá sido hace tanto tiempo que cualquier experiencia que tenga al respecto será obsoleta. c) se preocupan más por los detalles estratégicos y las ganancias/perdidas a largo plazo (no le dan tanta bola a los costos menores, terminales, oficinistas). d) se interesan más en el panorama global de sistema, suelen no interesarse por los detalles. e) pueden trabajar con modelos abstractos de un sistema.

USUARIO OPERACIONAL	USUARIO SUPERVISOR	USUARIO EJECUTIVO
Usualmente tiene un panorama local	Puede o no tener un panorama local	Tiene un panorama global
Hace funcionar el sistema	Generalmente, está familiarizado con la operación	Provee la iniciativa par el proyecto
Tiene una visión física del sistema	Lo rigen consideraciones presupuestarias	No tiene experiencia operacional directa
	Actúa a menudo como intermediario entre los usuarios y los niveles superiores de administración	Tiene preocupaciones estratégicas

#### Clasificación de los usuarios por nivel de experiencia

Se puede diferenciar entre:

- **Amateurs:** es aquél que jamás ha visto una computadora y que exclama con frecuencia que él no entiende todo este asunto de las computadoras, el problema con este usuario es que encuentre difícil de entender el lenguaje que el analista usa para describir las características, funciones y opciones del sistema aún cuando se evite usar terminología de computación. Dejando de lado en entendimiento de la tecnología computacional, se el usuario ni siquiera puede entender el modelo de un sistema hay poca probabilidad que le satisfaga el sistema cuando por fin este terminado.
- **Novato presuntuoso:** es una persona que ha tenido que ver con uno o dos proyectos de desarrollo de sistemas o peor aún, es un usuario que posee una computadora personal, y por lo común, alega saber exactamente lo que quiere que el sistema haga y suele señalar todos los errores que analista cometió en el último proyecto. A menudo entra en discusiones sobre la tecnología específica que sabe usar para realizar el sistema.
- **Pequeño grupo de expertos:** desde luego que hay usuarios que realmente entienden el análisis de sistemas, también la tecnología de las computadoras, el único problema pudiera ser que el usuario y el analista obtengan tal placer de la discusión sobre herramientas y técnicas de análisis de sistemas, que se olviden por completo de que su verdadero objetivo es implantar un sistema.

#### **Administración (administradores)**

El analista de sistema entra en contacto con diversos tipos de administradores:

- **Administradores usuarios:** están a cargo de varias personas en el área operacional, donde se implementará el sistema. Por lo general son administradores de nivel medio, que desean sistemas que produzcan una variedad de informes internos y de análisis a corto plazo. Los informes internos suelen ser informes financieros, operacionales, de falla y otros.
- **Administradores de informática:** son las personas encargadas del proyecto en sí del sistema, y los administradores de nivel superior encargados de la administración global y distribución de los recursos de todo el personal técnico de la organización de creación o desarrollo de sistemas.

- **Administradores generales:** son los administradores de nivel superior que no están directamente involucrados con la organización de la información ni son de la organización usuaria, pero por su poder llevan a cabo en forma global el proyecto.

Hay varios puntos que conviene tener en cuenta acerca de los administradores:

- Cuando más alto nivel ocupen menos probable es que sepan o que les importe saber de la tecnología de las computadoras.(no olvide tratar con ellos las caract. esenciales del sist.)
- Las metas y prioridades de la administración pudieran entrar en conflicto con las de los usuarios, incluso la administración pudiera querer imponerles un sistema y obligarlos a usarlo.
- Pudiera ser que la administración no esté dando los recursos, los fondos o el tiempo que los usuarios crean necesarios para implantar un sistema efectivo.
- Por sus diferentes puntos de vista y opiniones puede suceder que algunos estén a favor del nuevo sistema y otros rotundamente en contra.
- Puede ser que fuerzas externas obliguen a la administración a acelerar determinados proyecto, a quitarle recursos o, de plano, abandonarlo.

### **Audidores, control de calidad y departamento de normas o standares**

Según sea el tamaño de su proyecto y la naturaleza de la organización pueden haber:

- Auditores
- Personal de control de calidad
- Miembros del departamento normas o estándares participando en su proyecto

Se ha agrupado a éstas personas en una sola categoría por que su objetivo y perspectiva se parecen en general si no es que son iguales. El objetivo general de éste equipo es asegurar que el sistema se desarrolle de acuerdo con diversos estándares o normas externos, es decir externos al su proyecto.

*Hay tres problemas que se deben resolver cuando se está trabajando con este grupo:*

- No se involucra sino hasta el final del proyecto. Después que se haya terminado con el trabajo de análisis, el diseño y la programación, y cuando se ha comenzado con la prueba formal, a estas alturas, por supuesto, es muy difícil hacer cambios importantes en el sistema.
- A menudo están familiarizados con alguna notación o formato para la documentación de requerimientos del sistema, por eso es necesario asegurarse de que los modelos del sistema sean comprensibles.
- Por desgracia los miembros de éste grupo a menudo se interesan más por la forma que por el contenido, o sea que si la documentación del sistema no tiene la presentación exacta que se exige va a poder ser rechazada.

### **El analista de sistemas**

Es el personaje clave en cualquier proyecto de desarrollo de sistemas, en un sentido más amplio el analista desempeña varios papeles.

- **Arqueólogo y escribano:** como analista uno de sus principales virtudes es descubrir detalles y documentar la política de un negocio que pudiera existir solo como tradiciones triviales, o sea transmitidas de generación en generación por los usuarios.
- **Innovador:** el analista debe distinguir entre síntomas, problemas del usuario y causas. Con sus conocimientos de la tecnología de la computadora , el analista, debe ayudar a explorar aplicaciones novedosas y más útiles, así como formas nuevas de hacer negocios. Aunque muchos de los sistemas antiguos solo se limitaban a perpetuar el negocios originales del usuario pero muy velozmente, hoy

los analistas se enfrentan al desafío de ayudar al usuario para poder solucionar sus problemas inherentes a la toma de decisiones. (como encontrar mercados y productos convenientes).

- **Mediador:** el analista a menudo se encuentra en el medio, entre usuarios administradores y personal superior (y otros participantes). Es tentador para el analista imponer su propia opinión respecto a como debe hacerse el sistema o cuáles son las funciones que debe cumplir, pero su labor primordial es obtener un consenso y eso requiere el delicado arte de la diplomacia y la negociación.
- **Jefe de proyecto:** dado que el analista es la persona con más experiencia dentro de un grupo que elabora un proyecto, hay una tendencia natural a asignar al analista a responsabilidad de la administración total del proyecto.

Esto significa que se requieren: facilidades para la conducción de grupos de personas para entrevistar a los usuarios, mediar en desacuerdo y sobrevivir a las inevitables batallas políticas que se dan en todo proyecto. Por lo tanto el analista debe tener:

- *Conocimiento de la aplicación:* para entender y apreciar los asuntos de usuarios.
- *Conocimientos en hardware y software últimos:*
- *Mente lógica y organizada:* capaz de ver un sistema de diferentes perspectivas, debe poder dividirlo en niveles de subsistemas y ser capaz de pensar en el sistema en términos abstractos además de físicos.

### **Diseñadores de sistemas**

Generalmente en la actualidad el analista y el diseñados son la misma persona o grupo de personas, aún cuando sean personas distintas es importante que se mantengan en contacto directo a lo largo de todo el proyecto. La razón por la cuál se necesita esta retroalimentación continua entre el diseñador y el analista es la siguiente: *el analista tiene que ofrecer información detallada suficiente como para que el diseñador pueda elaborar un diseño tecnológicamente superior y el diseñador debe proveer suficiente información para que el analista pueda darse cuenta de si los requerimientos que del usuario está documentando son tecnológicamente posibles. Basándose en la información recibida, en analista posiblemente tendrá que negociar con el usuario para modificar otros requerimientos.*

### **Los programadores analistas**

Existen razones por las cuales los analistas y los programadores pudieran tener un contacto reducido o nulo entre sí: a menudo se lleva acabo el trabajo siguiendo una secuencia muy estricta en algunos proyectos de desarrollo de sistemas. Por eso la labor del analista se hace primero y se termina por completo antes que comience la labor de programación. El analista en algunos proyectos de desarrollo de sistemas define las prioridades que tendrá que tener en cuenta para la resolución de problemas.

Es probable que si haya algún contacto entre programadores y analistas, por lo sgte.:

- En los proyectos pequeños, una sola persona hace tanto de analista como de diseñador y por lo tanto interactúa con el programador, o una persona realiza las tareas de diseñador y programador.
- En analista a veces sirve de administrador del proyecto, así que aunque haya concluido su labor de especificación de los requerimientos del sistema, aún estará involucrado en el proyecto y tendrá algún contacto con el programador.
- Si algo falta, o está mal o confuso, el programador tiene dos opciones, pedirle un aclaración al analista o bien preguntarle al usuario.

### **Personal de operaciones**

El analista debe entrar en contacto con el personal de operaciones responsable del centro de computo, la red de telecomunicaciones, la seguridad del hardware y del software, además de la ejecución de los programas, el montaje de los discos y el manejo de la salida de las impresoras. Sin la aprobación del personal de operaciones, sólo se podría construir un sistema realmente independiente, nunca uno integrado o generalizado.

### Herramientas del análisis estructurado

Gran parte de la labor como analista, involucra el modelado del sistema que desea el usuario. El término modelo puede parecer algo formal pero representa un concepto que se maneja durante la mayor parte de la vida. Consideremos lo siguiente tipos de modelos:

**Mapas:** Modelos bidimensionales de nuestro mundo.

**Globos terráqueos:** Modelos tridimensionales de nuestro mundo.

**Diagrama de flujo:** Representaciones esquemáticas de las decisiones y la secuencia de actividades para llevar a cabo un determinado procedimiento.

**Dibujos arquitectónicos:** Representación esquemática de un edificio, o puente, etc.

**Partituras musicales:** Representaciones gráficas y textuales de notas musicales y tiempos de una pieza musical.

¿Por qué construimos modelos? ¿ Por qué no se construye simplemente el sistema mismo?

La respuesta es que podemos construir modelos de manera tal que enfatizamos ciertas propiedades críticas del sistema mientras que simultáneamente desacentuamos otros de sus aspectos. Esto nos permite comunicarnos con el usuario, sin distraernos con asuntos y características ajenas al sistema y si nos damos cuenta de que nuestra comprensión de los requerimientos del usuario no fue la correcta, podemos hacer cambios en el modelo o desecharlo y hacer uno nuevo si es necesario.

Por esta razón el analista hace uso de herramientas de modelado para:

- Concentrarse en las propiedades importantes del sistema y al mismo tiempo prestar atención a otras menos importantes.
- Discutir cambios y correcciones de los requerimientos del usuario, a BAJO COSTO y con el riesgo mínimo.
- Verificar que el analista comprenda correctamente el ambiente del usuario y que lo haga respaldado con información documental para que los diseñadores del sistema y los programadores puedan construir el sistema.

### Herramientas del modelado del sistema

- Diagrama de flujos de datos
- Diagrama de entidad – relación
- Diagrama de transición de estados
- **Diagrama de flujos de datos:**

Ilustra las funciones que el sistema debe realizar. Consisten en procesos, agregados de datos, flujos y terminadores.

- Los **procesos** se representan por medio de círculos o burbujas en el diagrama, representan las diversas

funciones individuales que el sistema lleva a cabo. Las funciones transforman entradas en salidas.

- Los **flujos** se muestran por medio de flechas curvas, son las conexiones entre los procesos y representan la información de dichos procesos.(E/S), describen movimientos de bloques o paquetes de información de una parte del sistema a otra, *datos en movimientos*.
- Los **agregados de datos**, se representan por medio de dos líneas paralelas o mediante una elipse. Muestran colecciones o agregados de datos. Cuando se termine de construir el sistema los agregados existirán como *archivos o bases de datos*, o **almacenes** que se utilizarán para modelar una colección de paquetes de *datos en reposo*.
- Los **terminadores**, gráficamente se representan con un rectángulo, muestran las entidades externas con las que el sistema se comunica, comúnmente es una persona o un grupo, pero fuera del control del sistema que está modelando. Puede ser otro sistema con el cual se comunica.

Aunque el Diagrama De Flujos De Datos proporciona una visión global, bastante conveniente de los componentes funcionales del sistema, no da detalles de éstos, para mostrar detalles a cerca de que información se transforma y de cómo se transforma, se ocupan dos herramientas esenciales y textuales adicionales que son:

- El diccionario de datos
- La especificación del proceso

#### El modelado de datos almacenados: El diagrama de entidad–relación

Aunque el diagrama de flujo es una herramienta muy útil para modelar los sistemas, solo resalta un aspecto principal del sistema: sus funciones. La notación de los agregados de datos en los diagramas de flujos de datos, muestra la existencia de uno o más grupos de datos almacenados, pero deliberadamente dice muy poco acerca de sus detalles.

Todos los sistemas almacenan y usan información acerca del ambiente en el cual interactúan, a veces la información es mínima, pero en la mayoría de los sistemas actuales es bastante complejo. No solo deseamos conocer en detalle que información hay en cada agregado de datos, sino que también queremos conocer la relación que existe entre agregados. Este aspecto no es resaltado por el diagrama de flujo de datos, pero si el diagrama de entidad–relación.

*El diagrama de entidad – relación consta de dos componentes principales:*

- **Tipo de objetos:** se representa por medio de un rectángulo en el diagrama, esto representa una colección o grupo de objeto cuyos miembros juegan algún papel en el desarrollo del sistema y pueden ser identificados de manera única y ser descripto por uno o más atributos.
- **Relaciones:** se representan por medio de rombos en el diagrama y son las series de conexiones o asociaciones entre los tipos de objetos que están conectados con la relación por medio de líneas o flechas.

CLIENTE

COLOCA RECIBE

PEDIDO FACTURA

ESPECIFICA

LIBRO DE CONTABILIDAD

*También son componentes del DER:*



- **Indicadores asociativos de tipo de objeto:** representa algo que funciona como objeto y como relación, representa una relación acerca de la cual se desea mantener alguna información. Se escribe dentro de una caja rectangular conectada por medio de líneas dirigidas a un rombo de relación sin nombre.
- **Indicadores de subtipo/supertipo:** consisten en objetos de una o más subcategorías, conectados por una relación, subtipos se conectan al supertipo por medio de una relación sin nombre, el supertipo se conecta a la relación con una línea que contiene una barra.

Es necesario acompañar el diagrama de entidad–relación con información textual detallada. El diccionario de datos también puede usarse.(definiciones de relaciones)

### Modelado De Estructura De Los Datos: El diagrama de transición de estados

El comportamiento dependiente de tiempo, es decir, la secuencia con la cual se hará el acceso a los datos y se ejecutarán las funciones, es un tercer aspecto de muchos sistemas complejos. Para algunos sistemas este aspecto no es importante puesto que la secuencia es esencialmente trivial, así en muchos sistemas (no son en tiempo real ni están en línea) la función  $n$  no puede llevar a cabo su labor hasta que recibe la entrada de que requiere, y esta entrada se produce como una salida de una función  $n - 1$  y así sucesivamente.

Sin embargo muchos sistemas en línea y en tiempo real, tanto en los negocios como el de las ciencias, tienen complejas relaciones en el tiempo que deben modelarse tan cuidadosamente como las funciones y la relación entre datos. Por ejemplo, muchos sistemas de tiempo real deben responder dentro de un margen breve de tan solo microsegundos a ciertas entradas provenientes del ambiente exterior. Y deben estar preparados para recibir diversas combinaciones y secuencias de entrada a las cuales se debe responder adecuadamente.

La herramienta de modelado que utilizamos para describir este aspecto de comportamiento de un sistema es el diagrama de transición de estados. Un diagrama típico es el comportamiento de una lavadora controlada por una computadora. En este diagrama los rectángulos representan los estados en los que se puede encontrar el sistema, cada estado representa entonces un período durante el cual el sistema sigue algún comportamiento observable. Las flechas que conectan a los rectángulos representan un cambio de estado o transición de un estado a otro. Hay una o más condiciones asociadas con cada cambio de estado y una o más acciones, es decir, respuestas, salidas o actividades que se llevan a cabo como parte del cambio de estado.

INACTIVO

COMENZAR ALTO

HABILITAR EL LLENADO DESHACER EL LLENADO

LLENADO

LAVADORA LLENA ALTO

HABILITAR EL LAVADO DESHACER EL LAVADO

LAVADO

CICLO LAVADO CONCLUIDO CICLO SECADO CONCLUIDO

HABILITAR EL SECADO CENTRIFUGADO DESHACER EL SECADO

SECADO Y CENTRIFUGADO

## CENTRIFUGADO

Diagrama de transición de datos

### El modelado de la estructura de los programas: El diagrama de estructuras

Esta herramienta adicional sirve para representar la jerarquía de software, es decir la jerarquía de módulos o los que a veces se conocen como subrutinas o procedimientos, esta jerarquía de software resulta de la utilización de las herramientas principales por el analista.

Es una herramienta excelente para los diseñadores de sistemas, no es el tipo de modelo que normalmente se mostraría al usuario, pues modela un aspecto de la implantación del sistema, no de sus requerimientos.

## **EL CICLO DE VIDA DEL PROYECTO**

### EL CONCEPTO DE CICLO DE VIDA DE UN PROYECTO

Cada vez son más las empresas grandes y pequeñas que están adoptando un ciclo de vida uniforme y único para sus proyectos. Esto se conoce como plan del proyecto, o metodología del desarrollo del sistema o, simplemente, la forma en la que hacemos las cosas aquí. El manual de ciclo de vida del proyecto suele ser un libro voluminoso, que ofrece un procedimiento común a seguir para desarrollar un sistema computacional que puede orientar a cualquier miembro de la organización de desarrollo de sistemas.

El ciclo de vida de un proyecto tiene 3 objetivos principales:

- Definir las actividades a llevarse a cabo en un proyecto de desarrollo de sistemas.
- Lograr congruencia entre la multitud de proyectos de desarrollo de sistemas en una organización.
- Proporcionar puntos de control y revisión administrativa de las decisiones sobre continuar o no con un proyecto.

El ciclo de vida del proyecto no está a cargo del proyecto, sino que el administrador debe administrar en todo el sentido de la palabra. La única ayuda que puede proporcionarle el ciclo de vida de un proyecto es que puede organizar las actividades del administrador, aumentando la probabilidad de que se aborden los problemas pertinentes en el momento adecuado.

### EL CICLO DE VIDA DEL PROYECTO CLÁSICO

Lo que caracteriza al ciclo de vida de un proyecto como clásico son dos aspectos, primero, una fuerte tendencia a la implantación ascendente del sistema y segundo, la insistencia en la progresión lineal y secuencial de una fase a la siguiente.

#### El ciclo de vida ascendente: **Implantación Ascendente**

Se espera que los programadores lleven a cabo primero sus pruebas modulares, luego las pruebas del subsistema, y finalmente las pruebas del sistema mismo. Este enfoque también se conoce como ciclo de vida de cascada.

El enfoque ascendente presenta un gran número de dificultades serias:

- Nada está hecho hasta que todo está terminado. Si el proyecto se atrasa y la fecha límite cae precisamente en medio del proceso de prueba del sistema, no habrá nada que mostrarle al usuario.
- Las fallas más triviales se encuentran al comienzo del período de prueba y las más graves al final.

Tales fallas pueden obligar a la recodificación de un gran número de módulos y pueden tener un impacto devastador sobre el calendario.

- La eliminación de fallas suele ser extremadamente difícil durante las últimas etapas de prueba del sistema, a menudo suele ser difícil determinar en cuál módulo la contiene.
- El administrador del proyecto descubre que necesita una gran cantidad de horas–máquinas para probar el sistema, el proyecto suele retrasarse mucho.

### Progresión secuencial:

Es una insistencia en que las fases se sucedan secuencialmente. El enfoque secuencial no permite el tratamiento de fenómenos reales como los relacionados con el personal, la política de la compañía, o la economía. Otros problemas asociados con el ciclo de vida del proyecto clásico o secuencial, es que durante los meses o años que toma desarrollar el sistema el usuario pudiera cambiar de parecer respecto a lo que debe hacer el sistema.

### EL CICLO DE VIDA SEMIESTRUCTURADO

En este se muestran dos detalles obvios no presentes en el enfoque clásico:

- La secuencia ascendente de codificación, la prueba de módulos y prueba del sistema se reemplaza por implantación de arriba hacia abajo, que es un enfoque en el cual los módulos de alto nivel se codifican y prueban primero, seguidos por los de bajo nivel, más detallados.
- El diseño clásico se reemplaza por el diseño estructurado.

La implantación descendente significa que se pondrán en ejecución paralelamente parte de la codificación y de las pruebas. Esto difiere de las fases secuenciales, y puede darse la retroalimentación entre la codificación, la prueba y la eliminación de las fallas.

A menudo la implantación descendente ofrece retroalimentación entre el proceso de implantación y el de análisis, aunque esto no se muestre específicamente y aunque el usuario y el administrador del proyecto de proceso de datos pudieran negar que esté sucediendo.

Otro punto a tratar acerca del ciclo de vida semiestructurado es que tenemos una gran parte del trabajo que se realiza bajo el nombre de *diseño estructurado* que es en realidad un esfuerzo manual por enmendar especificaciones erróneas. Los que llevan a cabo el diseño estructurado han supuesto tradicionalmente que se les daría una especificación clásica, en consecuencia, su primera tarea es transformar la especificación en un paquete de diagramas de flujo de datos, de diccionarios de datos, de diagramas de entidad relación y de especificaciones de procesos.

### El ciclo de vida estructurado del proyecto

Al presentar el análisis estructurado, además de extenderse con la idea de la retroalimentación entre una parte del proyecto y otra se crea un tipo de ciclo de vida del proyecto totalmente distinto, que es el ciclo de vida estructurado.

En este ciclo tenemos individuos o grupos que proporcionan las entradas al equipo del proyecto, (como son los usuarios, administradores, personal de operación), y que interactúan con una serie de actividades tales como:

Act 1. La encuesta: se conoce como el estudio de factibilidad o como el estudio inicial de negocios, empieza cuando el usuario solicita que una o más partes de su sistema se automaticen. Sus principales objetivos son: – identificar a los usuarios responsables y crear un campo de actividad inicial del sistema. – identificar las

deficiencias actuales en el ambiente del usuario, comprenderá la lista de funciones que hacen falta o que se hacen mal en el sistema actual. – establecer metas y objetivos para un sistema nuevo. – determinar si es factible automatizar el sistema y de ser así, sugerir escenarios aceptables. – preparar el esquema que se usará para guiar el resto del proyecto.

Act 2. El análisis de sistema: el propósito principal es transformar sus dos entradas o insumos o factores principales, las políticas del usuario y el esquema del proyecto, en una especificación estructurada. Esto implica modelar el ambiente del usuario con diagramas de flujo de datos, diagramas de entidad–relación, diagramas de transición de estados y demás herramientas. Este proceso se hace para crear un modelo esencial que representa una descripción formal de lo que el nuevo sistema debe hacer.

Act 3. El diseño: se dedica a asignar porciones de la especificación (modelo esencial) a procesadores adecuados, sean máquinas o humanos, y a labores apropiadas o tareas, particiones, etc. dentro de cada procesador. Se dedica a la creación de una jerarquía apropiada de módulos de programas y de interfases entre ellos par implantar la especificación creada en la act.2.

Act 4. Implantación: esta actividad incluye la codificación y la integración de módulos en un esquema progresivamente más completo del sistema final, por esto incluye tanto programación estructurada como implantación descendente.

Act 5. Generación de pruebas de aceptación: las especificaciones deben tener toda la información necesaria para definir un sistema que sea aceptable desde el punto de vista del usuario. Una vez generada la especificación, puede comenzar la actividad de producir un conjunto de casos de prueba de aceptación desde la especificación estructurada.

Act 6. Garantía de calidad: se conoce como la prueba final o la prueba de aceptación. Esta actividad requiere como entrada los datos de la prueba de aceptación generada en la act.5 y el sistema integrado producido en la act.4. También se le llama **control de calidad**.

Act 7. Descripción de procedimiento: es la generación de un descripción formal de las partes del sistema que se harán en forma manual, lo mismo que la descripción de cómo interactuarán los usuarios con la parte automatizada del nuevo sistema. El resultado esta actividad es un manual par el usuario.

Act 8. Conversión de bases de datos: involucra más trabajo que el desarrollo de programas (en algunos casos) de computadora para el nuevo sistema. En otros casos pudiera no haber existido una base de datos que convertir. En general esta actividad requiere como entrada la base de datos actual del usuario, al igual que la especificación del diseño producida por medio de la act.3.

Act 9. Instalación: esta es la actividad final, sus entrada son el manual del usuario producido en la act.7, la base de datos convertida que se creó en la act.8 y el sistema aceptado producido por la act.6, en algunos casos la instalación pudiera significar simplemente un cambio de la noche a la mañana al nuevo sistema, en otros casos la instalación pudiera ser un proceso gradual en el que un grupo tras otro de usuarios van recibiendo manuales y entrenamiento y comenzando a usar el nuevo sistema.

Nada implica que toda una actividad debe concluir antes de comenzar las otras. Por el contrario podrían estar dándose paralelamente. Debido a este aspecto de *no secuencial* se usa la palabra actividad en el ciclo de vida estructurado en lugar de fase, que es más convencional y se refiere a hacer una y solo una actividad.

### Implantación radical contra implantación descendente conservadora

El enfoque radical del ciclo de vida del proyecto estructurado es aquel en el que las actividades se llevan a cabo paralelamente desde el principio del proyecto, en cambio en el enfoque conservador del ciclo de vida del

proyecto estructurado, la actividad N completa se termina antes de comenzar con la N+1.

Estos son los puntos extremos de una gama de opciones, existe un infinito número de opciones entre los extremos radical y conservador.

### El ciclo de vida de prototipos

Una variación del enfoque descendente, en general se lo conoce como el enfoque de prototipos: consiste en capturar un conjunto inicial de necesidades e implantarlas rápidamente con la intención declarada de expandirlas y refinarlas iterativamente al ir aumentando la comprensión que del sistema tienen el usuario y quien lo desarrolla. Se realiza mediante el descubrimiento evolutivo y gradual y no a través de la previsión omnisciente...

Esto suena como el enfoque descendente radical pero su principal diferencia es que este implica tarde o temprano se construirá un modelo en papel completo del sistema.

### ASPECTOS IMPORTANTES EN EL DESARROLLO DE SISTEMAS

Los aspectos más importantes en el desarrollo de sistemas son:

- PRODUCTIVIDAD
- CONFIABILIDAD
- MANTENIBILIDAD

### PRODUCTIVIDAD: EL RETRASO EN LAS APLICACIONES

Este es el problema más visible al que se enfrenta el desarrollo de sistemas. La sociedad y los negocios modernos parecen exigir cada vez más: más sistemas, más complejidad y todo más rápido. El retraso se presenta en tres tipos diferentes:

- Retraso visible: sistemas nuevos que los usuarios han pedido oficialmente y que han sido aprobados y financiados , pero sin embargo los proyectos no se han iniciado porque no existen los recursos adecuados (esto es, analistas, programadores, etc.).
- Retraso invisible: existen sistemas nuevos que el usuario saben que necesitan, pero no se han molestado en pedirlos oficialmente, porque aún están aguardando a que se concluyan proyectos del retraso visible.
- Retraso desconocido: estos son sistemas nuevos que los usuarios ni siquiera saben que quieren todavía, pero que serán identificados en cuanto se termine alguno de los sistemas del retraso visible o del invisible.

Esto sugiere que el problema del retraso tiene una pequeña porción visible y una gran parte oculta, siendo por lo tanto un problema de primera importancia para la organización de desarrollo de sistemas que lleva a cabo su planeación y sus presupuestos basándose sólo en las exigencias conocidas y visibles de sus servicios.

Otro aspecto importante de la productividad es el tiempo necesario para desarrollar un sistema individual, un sistema nuevo se asocia con una oportunidad de negocio que el usuario percibe, y esa oportunidad a menudo tiene un margen de oportunidad, es decir, un período tras el cual ésta habrá desaparecido y ya no se necesitará el sistema nuevo.

Otro motivo del problema es que algunos proyectos resultan inútiles y la administración los cancela antes de que se terminen.

El problema de la productividad ha existido por muchos años en el desarrollo de sistemas, muchas organizaciones están buscando la forma de reducir su retraso en las aplicaciones y disminuir el tiempo requerido para desarrollar un nuevo sistema. Entre las técnicas más comúnmente utilizadas están las siguientes:

- Contratar más programadores y analistas.
- Contratar programadores y analistas más talentosos y darles mejores condiciones de trabajo.
- Permitir a los usuarios desarrollar sus propios sistemas.
- Mejores lenguajes de programación.
- Atacar el problema del mantenimiento.
- Disciplinas de ingeniería de software: es el conjunto de herramientas, técnicas y disciplinas conocidas como ingeniería de software o técnicas estructuradas que incluyen la programación estructurada, el diseño estructurado y el análisis estructurado, además disciplinas relacionadas como la métrica de software, pruebas de corrección de programas control de calidad de software.
- Herramientas automatizadas para desarrollo de sistemas. Vemos que una razón del problema de la productividad es que mucho del trabajo de desarrollo de un sistema automatizado de información se realiza, irónicamente, de manera manual, los programadores y analistas tradicionalmente han sido los últimos en gozar de los beneficios de la automatización en su propio trabajo. Ahora hay estaciones de trabajo que automatizan la mayor parte de la tediosa labor de desarrollar sistemas y mantener diagramas y modelos gráficos. Estas herramientas automatizadas también se encargan de una variedad de actividades de revisión de errores y en algunos casos pueden incluso generar código directamente de la especificación, eliminando de esta manera la actividad manual de manera absoluta.

Muchos de estos enfoques de productividad pueden usarse en conjunto y de esta manera pueden doblarse la productividad de la organización.

Existen tres razones por las que un analista debe ser muy sensible al asunto de la productividad:

- La calidad del trabajo hecho por el analista puede tener un impacto tremendo sobre el tiempo que se invierte en pruebas, dado que muchos errores usualmente se asocian con errores de análisis.
- Algunas de las técnicas de productividad son de relevancia directa para el analista. Vale la pena pensar qué puede hacerse para volver a usted y a su trabajo más productivo.
- La productividad del análisis de sistema es un asunto políticamente delicado, pues a menudo le parece al usuario que se hace muy poco durante la fase de análisis.

## CONFIABILIDAD

Otro gran problema al que se enfrentan los que desarrollan sistemas es la confiabilidad. Los sistemas producidos por organizaciones en todo el mundo están llenos de errores y son casi imposibles de modificar.

Los errores del software van desde los sublimes hasta los ridículos. Un error trivial pudiera consistir en salidas (resultados) correctas, pero que no se imprimen o presentan como el usuario desearía. Un error moderadamente serio sería un caso en el cual el sistema se rehusa a reconocer ciertos tipos de entradas, pero el usuario final puede hallar alguna forma de saltar el problema. Los errores serios serían los que ocasionan que todo el programa deje de funcionar, con una pérdida asociada de dinero o de vidas humanas.

Muchos de los errores de software jamás se reportan porque el individuo o la organización culpables preferirían no admitirlo públicamente.

En muchos casos nadie está seguro respecto a cuántos errores tiene un sistema, pues 1) algunos errores jamás se encuentran antes de que caduque el sistema y 2) el proceso de documentación y registro de errores es tan descuidado que la mitad de los errores encontrados no se reportan.

Otro punto a señalar acerca de los errores de software es que no son fáciles de enmendar, cuando alguien descubre que el software no funciona correctamente deben suceder dos cosas: 1) el programador debe identificar el origen y la naturaleza del error y 2) debe encontrar la manera de corregir el error sin afectar algún otro aspecto de la operación del sistema.

## MANTENIBILIDAD

La corrección de errores sobre la marcha es un aspecto del mantenimiento. El mantenimiento también involucra la modificación de un sistema para reflejar cambios en el hardware, modificaciones para apresurar ciertos aspectos operacionales o modificaciones para reflejar cambios en los requerimientos del usuario final del sistema.

El mantenimiento de software es un problema primordial en muchas organizaciones. El resultado, en más de alguna organización de proceso de datos, es que los sistemas existentes que se crearon hace ya algún tiempo simplemente no pueden modificarse para ajustarlos a las nuevas exigencias del gobierno, de la economía, del clima o de la volubilidad del usuario.

El problema es peor aún, ya que si fuera simplemente un caso de que el software fuese malo, se podría considerar desecharlo o reemplazarlo. Pero muchas organizaciones jamás han capitalizado su software (los costos se determinan cada año), y sus políticas de administración y de negocios hacen que se vuelva prohibitivamente cara para reemplazar los sistemas antiguos, y un problema aún más fundamental es que en la mayoría de las organizaciones, no existe una descripción coherente de lo que se supone que los sistemas deben hacer. Aunque se pueda argumentar que la mantenibilidad es enteramente función de la implantación (es decir, algo que preocupa a los programadores), es imposible mantener un sistema si no existe un modelo preciso y actualizado de sus requerimientos. Esto, a fin de cuentas, es labor del analista.

## OTROS ASUNTOS

Además de la productividad, la fiabilidad y la mantenibilidad, el analista debe preocuparse de otros asuntos, los cuales varían de una organización a otra y de un proyecto a otro, pero por lo general incluye lo siguiente:

- Eficiencia: un sistema debe operar mediante salidas o productos (o resultados) apropiados (usualmente medidas en transacciones por segundo) y con un tiempo de respuesta adecuado para las terminales en línea.
- Transportabilidad: la mayoría de los sistemas nuevos se realizan en una marca de computadoras, pero pudiera haber necesidad de desarrollar el sistema de tal manera que se le pueda mudar fácilmente entre computadoras.
- Seguridad: el nuevo sistema debe evitar el acceso no autorizado, además de la actualización y la eliminación no autorizadas de datos delicados.

## CAMBIOS EN EL ANALISIS DE SISTEMAS

### EL MOVIMIENTO HACIA EL ANALISIS ESTRUCTURADO

Hasta fines de los años 70, el analista escribía lo que entendía de los requerimientos del usuario en un enorme documento que consistía primariamente en una narrativa. Los primeros autores de textos de análisis estructurado señalaron que estos pesados tomos (a menudo conocidos como especificación funcional) se veían afectados por diversos problemas importantes:

- Eran monolíticos: había que leer completamente la especificación, de principio a fin, para poder entenderla. Es una falla importante, pues existen muchas situaciones en las que el analista quisiera leer y comprender una parte de la especificación sin tener que leer necesariamente las demás.

- Eran redundantes: a menudo se repetía la misma información en diversas partes del documento. El problema con esto es que si se tenían que hacer cambios en los aspectos de los requerimientos, el cambio debía reflejarse en varias partes del documento (hoy sería un problema mucho menor).
- Eran ambiguas: el reporte detallado de los requerimientos podía ser (y a menudo era) interpretado de diferente manera por el usuario, el analista, el diseñador y el programador.
- Eran imposibles de mantener: por todas las razones descritas anteriormente, la especificación funcional era casi obsoleta para cuando llegaba el final del proceso de desarrollo del sistema (es decir cuando el sistema se ponía en operación), y a menudo era obsoleto para el final de la fase de análisis. Esto significa que la mayoría de los sistemas que se desarrollaron durante décadas pasadas no tienen definiciones actualizadas de las políticas de negocios que se supone que llevan a cabo.

Mientras se debatían todos estos problemas, ya se estaba adaptando un conjunto complementario de ideas en el área de programación y diseño, normalmente conocidas como diseño y programación estructurados, que prometían grandes mejoras en la organización, codificación, prueba y mantenimiento de los programas de computadora.

Como resultado ha habido un movimiento gradual (ya que aceptarlo le ha llevado a la profesión de desarrollo de sistema varios años) tendiente a hacer especificaciones funcionales que sean:

- Gráficas: compuestas de una variedad de diagramas, apoyadas con material textual detallado que sirve de referencia más que como cuerpo principal de la especificación.
- Particionadas: de tal manera que se puedan leer independientemente porciones individuales de la especificación.
- Mínimamente redundantes: de tal manera que los cambios en los requerimientos del usuario puedan incorporarse normalmente en sólo una parte de la especificación.

Este enfoque que se conoce como estructurado, se utiliza ahora en la mayoría de las organizaciones de desarrollo de sistema orientados a los negocios, al igual que en gran número de las orientadas hacia la ingeniería.

### CAMBIOS EN EL ANÁLISIS ESTRUCTURADO CLASICO

Varios años de experiencia práctica con el análisis estructurado clásico han señalado un buen número de áreas en las que es necesario hacer cambios o extensiones. Los principales cambios son:

- El énfasis en la construcción de modelos *físicos actuales* y *lógicos actuales* del sistema del usuario se han demostrado que es políticamente peligroso. Esto no significa que hayamos decidido evitar modelar el sistema actual del usuario en todos los casos, sino que simplemente lo reconocemos como una actividad políticamente peligrosa.
- La terminología, ahora nos referimos a modelos esenciales (modelos de la esencia del sistema) en lugar de modelos lógicos, y a modelos de implantación en lugar de modelos físicos.
- El análisis estructurado clásico no tiene manera de modelar el comportamiento dependiente del tiempo de un sistema; carece de la notación necesaria para mostrar interrupciones y señales, y para mostrar la sincronización y coordinación de distintas labores de proceso, para resolver esto se ha añadido la notación necesaria y una herramienta nueva completa, que incluye flujos de control, procesos de control y diagramas de transición de estados.
- El modelado de datos se hacía de una manera primitiva o se lo ignoraba, ahora para permitir el modelado de sistemas con relaciones complejas entre datos se introdujeron los diagramas de entidad-relación.
- Se ha añadido un nuevo enfoque conocido como división de eventos.

### EL SURGIMIENTO DE HERRAMIENTAS AUTOMATIZADAS DE ANÁLISIS



En un sistema muy grande puede haber 50, 100 o más diagramas de flujo de datos, diversos diagramas de entidad-relación y, potencialmente, varios diagramas de transición de estados; por lo que la cantidad de trabajo manual puede ser en verdad intimidante. Además del trabajo que se necesita para crear y mantener los diagramas, el análisis estructurado clásico requiere de una gran cantidad de trabajos para verificar los diagramas con el fin de asegurar que sean consistentes y estén completos; debido a que esta labor es detallada y aburrida, tiende a estar plagada de errores. En consecuencia, no se encontraban muchos de los errores de especificación que se debieran haber encontrado.

Muchos de estos problemas se pueden resolver con apoyo automatizado adecuado. Esto era bien conocido aun desde que por primera vez se introdujo el análisis estructurado clásico, pero el costo de la automatización estaba por encima de las posibilidades económicas de la mayoría de las organizaciones. Sin embargo, el desarrollo de poderosas estaciones de trabajo gráficas llevó a que varios proveedores ofrezcan productos (normalmente basados en PC) que dibujan diagramas de flujo de datos y otros, además de llevar a cabo una variedad de labores de revisión de errores.

Esto lleva primordialmente a un nivel más elevado de calidad en los modelos de sistemas producidos. También, de manera secundaria, hará a modelos gráficos del sistema visualmente más atractivos. En la medida en que los conceptos de revisión de errores sean automatizados, podrán eliminar problemas de productividad, confiabilidad y en la medida en que las herramientas de Ingeniería de software auxiliada por computadora (CASE) comiencen a generar programas directamente desde las especificaciones, también incluso se reducirá la necesidad de programadores.

### EL USO DE PROTOTIPOS

Algunos usuarios tienen dificultades al tratar con los modelos gráficos del análisis estructurado y prefieren alguna otra forma de modelar los requerimientos y comportamientos del sistema. Las herramientas de generación de prototipos se han considerado como una alternativa al análisis estructurado para tales usuarios.

Existe otra razón de la popularidad de los prototipos, y es que se considera que el análisis estructurado clásico consume demasiado tiempo; para cuando concluye la fase de análisis, el usuario habrá olvidado para qué quería en un principio el sistema. Esto suele ser resultado de alguno de los siguientes problemas:

- El equipo encargado del proyecto tardó demasiado en desarrollar modelos del sistema actual y luego se tuvo que tardar aún más modelando el nuevo.
- La organización invirtió previamente poco o nada de tiempo, en hacer análisis, pues prefirió codificar lo antes posible. El trabajo de análisis pudiere parecerles improductivo.
- Los primeros proyectos que se han realizado utilizando el análisis estructurado pudieron consumir más tiempo de lo normal, pues los analistas están aprendiendo nuevas técnicas y discutiendo respecto a la mejor forma de aplicar dichas técnicas.

Las herramientas de generación de prototipos (herramientas de software que permiten al analista construir un simulacro del sistema) se ven por ello como una solución efectiva a estos problemas. También el uso de prototipos tiende a concentrarse en el aspecto de la interfaz humana en los proyectos de desarrollo de sistemas.

## UNIDAD III

### HERRAMIENTAS DE MODELADO

Un modelo es un simulacro a bajo costo de un sistema complejo que se desea estudiar. Se construyen modelos de sistemas por tres motivos:

- Para enfocar características importantes del sistema y minimizar las menos importantes.
- Para discutir cambios y correcciones a los requerimientos del usuario, a bajo costo y con riesgo mínimo.
- Para verificar que se entiende el ambiente del usuario, y que se ha documentado de tal manera que los diseñadores y programadores puedan construir el sistema.

Cualquier herramienta de modelado que use debe tener las siguientes características:

- **Debe ser gráfica, con detalles textuales de apoyo apropiados:** no es requisito usar gráficas, pero hay que tener en cuenta que una imagen vale más que mil palabras, una imagen bien escogida puede transmitir de manera concisa y compacta una gran cantidad de información. Se utilizan para identificar los componentes de un sistema y su interfaz, todos los demás detalles se presentan en documentos textuales de apoyo, *especificaciones de proceso* y *el diccionario de datos*.
- **Debe permitir que el sistema sea visto en segmentos, en forma descendente:** deben permitir mostrar partes individuales del sistema de manera independiente, junto con una forma sencilla de moverse de una parte a otra del modelo del sistema, de manera descendente un sistema de información proporciona un mecanismo conveniente para pasar tranquilamente de un nivel alto a uno bajo.
- **Debe tener redundancia mínima:** importa esto porque es deseable mantenerlo en forma actualizada y precisa. Si el sistema cambia, entonces debe cambiar el modelo, si ha de tenerse actualizado, si cambia un aspecto local del sistema, preferiríamos cambiar sólo un aspecto local del modelo sin tener que cambiar algún otro.
- **Debe ayudar al lector a predecir el comportamiento del sistema.**
- **Debe ser transparente para el lector:** un buen modelo debe ser tan fácil de leer que el lector no tenga que detenerse a pensar siquiera que se trata de la representación de un sistema y no del sistema mismo.

#### Diagrama de Flujo de Datos:

Es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por *conductos* y *tanques de almacenamiento* de datos.

#### Componentes del DFD:

- El Proceso.
- El Flujo.
- El Almacén.
- El Terminador.

#### Guía para la construcción de un DFD:

Reglas que ayudarán para no elaborar un DFD incompleto o lógicamente inconsistente y que se dibuje un DFD grato a la vista:

- **Escoger nombres con significado para los procesos, flujos, almacenes y terminadores:** es importante etiquetar con precisión el proceso, de modo que quienes leen el DFD puedan confirmar que se trata de un modelo preciso. Si el proceso lo hace una sola persona se recomienda identificar **el papel o función** que la persona está representando y no su identidad o nombre (podría ser reemplazado, podría realizar diversas tareas en el sistema o puede atraer la atención hacia la manera en la que realiza la labor dada).
- **Numerar los procesos:** es una forma conveniente de referirse a los procesos, no importa mucho cómo se haga esto, mientras haya constancia en la forma de aplicar los números. El modelo de DFD es una red asincrónica que se intercomunican. Pues se numeran los procesos para hacer referencia a los procesos de manera mas conveniente cuando se discute sobre un DFD ( decir burbuja 1 es más fácil que editar

transacciones y reportar errores), y es más importante el hecho de que los números se convierten en base para la numeración jerárquica cuando se introduzcan los diagramas de flujo por niveles.

- **Redibujar el DFD tantas veces como sea necesario estéticamente:** tendrá que dibujarse y volverse a dibujar, a menudo hasta 10 veces o más antes de *ser técnicamente correcto, ser aceptable para el usuario y estar lo suficientemente bien dibujado como para que no sea embarazoso mostrarlo a la dirección de la organización.*
- **Evite los DFD excesivamente complejos:** esto significa que el diagrama debe ser fácilmente entendido, asimilado y placentero a la vista, una regla principal es que el DFD debe caber cómodamente en una hoja normal.
- **Asegurarse de que el DFD sea internamente consistente y que también lo sea con cualesquiera DFD relacionado con él:** evitar burbujas que tengan entradas pero no salidas, evitar burbujas de generación espontánea, que tienen salidas sin tener entradas, tenga cuidado con los flujos y procesos no etiquetados, tenga cuidado con los almacenes de solo lectura o solo escritura (deben tener entrada tanto como salida, excepto los externos que sirven de interfaz entre el sistema y algún terminador externo).

### DFD por niveles:

Es organizar el DFD global en una serie de niveles de modo que cada uno proporcione sucesivamente más detalles sobre una porción de l nivel anterior.

El DFD del primer nivel consta sólo de una burbuja, que representa el sistema completo, los flujos de datos muestran las interfaces entre el sistema y los terminadores externos, junto con los almacenes externos que puedan haber. El DFD que sigue del diagrama de contexto se conoce como la figura 0, representa la vista de más alto nivel de las principales funciones del sistema, al igual que sus principales interfaces, cada una de estas burbujas debiera numerarse para una referencia conveniente, los números también sirven como una manera adecuada de relacionar una burbuja con el siguiente nivel del DFD que la describe más a fondo.

### Extensiones del DFD para sistemas de tiempo real:

Se necesita alguna manera de modelar *flujos de control*, es decir señales o interrupciones, y se requiere una manera de mostrar *procesos de control*, este es, burbujas cuya única labor es coordinar y sincronizar las actividades de otras burbujas del DFD. Se muestran gráficamente con líneas punteadas en el DFD.

### EL DICCIONARIO DE DATOS

El diccionario de datos es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el análisis tengan un entendimiento común de todas las entradas, salidas, componentes de almacenes y cálculos intermedios.

Define datos haciendo lo siguiente:

- Describe el significado de los flujos y almacenes que se muestran en los DFD.
- Describe la composición de agregados de paquetes de datos que se mueven a lo largo de los flujos, es decir paquetes complejos (domicilio de un cliente) que pueden descomponerse en unidades más elementales (ciudad, estado y código postal).
- Describen la composición de los paquetes de datos en los almacenes.
- Especifica los valores y unidades relevantes de piezas elementales de información en los flujos de datos y en los almacenes de datos.
- Describe los detalles de las relaciones entre almacenes que se enfatizan en un diagrama de entidad-relación.

### La necesidad de la notación en el diccionario de datos:

En la mayoría de los sistemas en los que se trabaja, los paquetes de datos o elementos de datos, serán lo suficientemente complejos como para necesitar describirlos en términos de otras cosas. Los elementos complejos de datos se definen en términos de elementos más sencillos, y los sencillos en términos de los valores y unidades legítimos que pueden asumir. *Se vuelve algo tedioso describir la composición de los elementos de datos en una forma narrativa. Necesitamos una notación concisa y compacta (como los diccionarios normales de palabras ordinarias).*

Notación del diccionario de datos:

Existen muchos esquemas, uno de los más comunes y que utiliza varios símbolos sencillos es:

=	está compuesto de
+	y
( )	optativo (puede estar presente o ausente)
{ }	iteración
[ ]	seleccionar una de varias alternativas
* *	comentario
@	identificador (campo clave) para un almacén
	separa opciones alternativas en la construcción

Ejemplos:

nombre = título de cortesía + nombre + (segundo nombre) + apellido

título de cortesía = [ Sr. | Srita. | Sra. | Dr. | Profesor ]

nombre = { caracter legal }

segundo nombre = { caracter legal }

apellido = { caracter legal }

caracter legal = [ A - Z | a - z | 0 - 9 | ` | - | | ]

Para definir un dato se introduce con el símbolo = , y se lee *se define como* o *significa*. Por ejemplo: A= A+B, para definir por completo un dato, debe incluir el significado del dato dentro del contexto de la aplicación del este usuario, usando la notación \* \* ; debe incluir la composición del dato, si se compone de partes elementales con significado; también los valores que puede tomar el dato. Por ejemplo:

peso = \*peso del paciente al ser admitido al hospital\*

\*unidades: Kilogramos; gama 1-200\*

estatura = \*estatura del paciente al ser admitido al hospital\*

\*unidades: centímetros; escala: 20–200\*

Hay términos que se definen solos, es decir, cuyo significado es universal para todos los sistemas de información, o que no se necesitan aclarar más, en estos casos no se necesita un comentario narrativo, se puede usar la notación \*\* para indicar *sin comentarios* cuando estos datos se definan solos. Por ejemplo:

peso actual = \*\*

\*unidades: libras; escala: 1–400\*

sexo = \*\*

\*valores: [ M | F ]\*

La notación de iteración se usa para indicar la ocurrencia de un componente de un dato, se lee como *cero o más ocurrencias de*, en muchas ocasiones se querrá indicar los límites inferior y superior de la iteración. Es correcto especificar sólo el límite inferior, sólo el superior, ambos, o ninguno, esto se puede indicar así:

$a = 1 \{b\}$  ;  $a = \{b\} 10$  ;  $a = 1 \{b\} 10$  ;  $a = \{b\}$  . Otro ejemplo:

solicitud = nombre del cliente + domicilio de envío +  $1 \{ \text{artículos} \} 10$

El alias, es una alternativa de nombre para un dato, cuando se insiste en utilizar nombres distintos para decir lo mismo, el alias se incluye en el diccionario de datos para que esté completo, y se relaciona con nombres primarios u oficial del dato. Por ejemplo:

comprador = \* alias de cliente \*

#### Como mostrar el diccionario de datos:

El diccionario de datos lo crea el analista durante el desarrollo del modelo del sistema, pero el usuario debe ser capaz de leerlo y entenderlo para poder verificar el modelo.

**Construir un diccionario de datos es una de las labores más tediosas, y largas, del análisis de sistemas. Pero también es una de las más importantes: sin un diccionario formal que defina el significado de los términos, no se puede esperar precisión.**

#### ESPECIFICACIONES DE PROCESO:

Es la descripción de qué sucede en cada burbuja primitiva de nivel más bajo en un diagrama de flujo de datos. Define lo que debe hacerse para transformar entradas en salidas, es una descripción detallada de política de negocios del usuario que cada burbuja lleva a cabo.

Existe una variedad de herramientas que podemos utilizar para producir una especificación de proceso: tablas de decisiones, **lenguaje estructurado** (español, inglés, etc.), pre/pos condiciones, diagramas de flujo, diagramas de Nasse/Shneiderman, etc. Se puede usar cualquier método mientras cumpla dos requisitos cruciales:

- La especificación del proceso debe expresarse de una manera que puedan verificar tanto el usuario como el analista. (razón pro la cual se evita el lenguaje narrativo ya que es ambiguo).
- El proceso debe especificarse en una forma que pueda ser comunicada efectivamente al público amplio que esté involucrado.

- Se puede agregar esta tercera : No debe imponer o implicar decisiones de diseño e implantación arbitrarias. El trabajo como analista consiste en destilar de esto la esencia de lo que se trate una política y no cómo se lleva a cabo hoy en día.

Las especificaciones de proceso sólo se desarrollan para los procesos de más bajo nivel en un conjunto de diagramas por niveles en un DFD.

Las tres herramientas principales de especificación de proceso son:

- Lenguaje estructurado (español, inglés, etc.)
- Pre/post condiciones
- Tablas de decisión

### Lenguaje estructurado:

Es un subconjunto de todo el idioma con importantes restricciones sobre el tipo de frases que pueden utilizarse y la manera en que pueden juntarse dichas frases. Su propósito es hacer un balance razonable entre la precisión del lenguaje formal de programación y la informalidad del lenguaje cotidiano. Por ejemplo, una frase puede ser una ecuación algebraica como  $X = (Y * Z) / (Q + 14)$  o una frase imperativa que consista en un verbo y un objeto como por ejemplo CALCULAR  $X = (Y * Z) / (Q + 14)$ .

El lenguaje estructurado permite que se combinen frases en unas cuantas formas limitadas que se toman de las construcciones acostumbradas de la programación estructurada.

- La construcción SI–ENTONCES–OTRO se utiliza para describir frases alternativas que se deben realizar según el resultado de la decisión binaria. Ejemplo:

**SI** edad–del–cliente es mayor que 65

fijar cuota a cuota–ancianos

**OTRO**

fijar cuota a cuota–normal

**FIN SI**

- La construcción CASO se utiliza par describir frases alternativas que se efectuarán basándose en los resultados de una decisión multivaluada. Ejemplo:

**HACER CASO**

CASO edad–del–cliente < 13

fijar cuota a cuota–niños

CASO edad–del–cliente >12 y edad–del–cliente < 20

fijar cuota a cuota–adolescente

CASO edad–del–cliente > 19 y edad–del–cliente < 65

fijar cuota a cuota–adultos

OTRO

fijar cuota a cuota–ancianos

FIN CASO

- La construcción HACER–MIENTRAS se usa para describir una frase que deberá llevarse a cabo repetitivamente hasta que alguna condición booleana se haga verdadera. Ejemplo:

HACER–MIENTRAS haya más artículos en el pedido–del–cliente

precio–extendido = precio–unitario \* cantidad–de–unidades

FIN HACER

- Otra estructura es REPITE–HASTA , que ejecuta una frase especificada por lo menos una vez antes de hacer una prueba para ver si debe repetirse. Ejemplo:

Se pueden construir frases compuestas a partir de combinaciones de frases sencillas y las estructuras sencillas que se presentaron anteriormente.

Pre/post condiciones:

Son una manera conveniente de describir la función que debe realizar el proceso, sin decir mucho acerca del algoritmo o procedimiento que se utilizará, es un enfoque útil cuando:

- El usuario tiene tendencia a expresar la política llevada a cabo por la burbuja en términos de un algoritmo particular que ha estado utilizando durante décadas.
- El analista está razonablemente seguro de que existen muchos algoritmos distintos que podrían usarse.
- El analista desea que el programador explore varios de estos algoritmos pero no quiere involucrarse personalmente con tales detalles y, sobre todo, no quiere enredarse en discusiones con el usuario acerca del mérito relativo de cada uno.

Un ejemplo de especificaciones de proceso escritas con este enfoque es:

ESPECIFICACIONES DE PROCESO 3.5: CALCULAR EL IMPUESTO SOBRE VENTAS

Precondición 1

Ocurre DATOS–VENTA con TIPO–ITEM que corresponde con

CATEGORIA–ITEM en CATEGORIAS–IMPUESTO

Postcondición 1

IMPUESTO–SOBRE–VENTA se hace igual a MONTO–VENTA \* IMPUESTO

Precondición 2

Ocurre DATOS–VENTA con TIPO–ITEM que no concuerde con

## CATEGORÍA–ITEM en CATEGORIA–IMPUESTO

### Postcondición 2

Se genera MENSAJE–ERROR

Las **precondiciones** describen todas las cosas, si hay, que deben darse antes de que el proceso pueda comenzar a ejecutarse. Describen lo siguiente:

- Qué entradas se encuentran disponibles
- Qué relación debe existir entre las entradas
- Qué relaciones deben existir entre entradas y almacenes de datos
- Qué relaciones deben existir entre almacenes o dentro de un almacén dado

Las **postcondiciones** describen que debe darse cuando el proceso ha concluido. Describen lo siguientes:

- Las salidas que generará o producirá el proceso
- Las relaciones que existirán entre los valores de salida y los valores originales de entrada
- Las relaciones que existirán entre valores de salida y los valores en uno o varios de los almacenes
- Los cambios que se hayan dado en los almacenes

### Tablas de decisión:

Si el proceso debe producir alguna salida o tomar alguna acción basada en decisiones complejas es preferible una tabla de decisiones, si las decisiones se basan en diversas variables distintas, por ejemplo: datos de entrada, y si dichas variables pueden tomar diversos valores, entonces la lógica expresada por el lenguaje estructurado o por las pre/post condiciones probablemente sea tan compleja que el usuario no la comprenderá.

Una tabla de decisiones se crea listando todas las variables relevante o entradas y todas las acciones relevantes en su lado izquierdo, las variables y acciones están separadas por una línea horizontal gruesa, a continuación se lista en columnas separadas cada combinación posible de valores de las variables, cada columna usualmente se conoce como regla. Una regla describe una acción o acciones que deben llevarse a cabo para una combinación específica de valores de las variables.

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---