

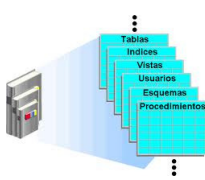
CLASE 9

Diccionario de Datos

Página

1

Definición



El diccionario de datos es la segunda herramienta de modelado más importante que discutiremos; aunque no tiene la presencia y el atractivo gráfico de los DFD, los diagramas de entidad relación y los diagramas de transición de estados, es crucial. Sin el diccionarios de datos, el modelo de los requerimientos del usuario no puede considerarse completo; todo lo que se tendría es un borrador rudimentario, una “visión del artista” del sistema. [...]

El diccionario de datos de frases casi se autodefine. 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 analista tengan un entendimiento común de todas las entradas, salidas, componentes de almacenes y cálculos intermedios. El diccionario de datos define los 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 (por ejemplo el domicilio de un cliente), que puede descomponerse en unidades más elementales (como 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. [...]

Notación del Diccionario de Datos



“Existen muchos esquemas de notación comunes utilizados por el analista de sistemas. El que se muestra a continuación es de los más comunes y utiliza varios símbolos sencillos:

Página

2

=	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”. ¹

Por ejemplo, veamos una definición de la entidad **Clientes**:

Ciudad	= ** descripción (registro) de ciudad **
	= @codigo_ciudad + nombre_ciudad + codigo_pais.id
CIUDADES	= ** entidad (archivo) ciudad **
	= {ciudad}
cliente	= ** descripción (registro) de cliente, datos personales **
	= @codigo_cliente + nombre + direccion + telefono
	codigo_pais.id + codigo_ciudad.id + sexo +
	codigo_estado_civil.id + saldo + estado
CLIENTES	= ** entidad (archivo) cliente **
	= {cliente}
codigo_cliente	= ** identificador de cliente, candidato a clave primaria, identificador único **
	= {dato_numerico}
codigo_ciudad	= ** identificador de la ciudad, candidato a clave primaria, identificador único **
	= {dato_numerico}
codigo_ciudad.id	= ** identificador de la ciudad, información tomada de la entidad ciudades **
	= {dato_numerico}
codigo_estado_civil	= ** identificador del estado civil, candidato a clave primaria, identificador único **
	= {dato_numerico}
codigo_estado_civil.id	= ** identificador del estado civil, información tomada de la entidad estados civiles **
	= {dato_numerico}

¹ Edward Yourdon. (1993). Análisis Estructurado Moderno. Prentice-Hall Hispanoamericana, SA. México. Pág. 210 a 214.

codigo_pais	= ** identificador de país, candidato a clave primaria, identificador único **
codigo_pais.id	= {dato_numerico}
	= ** identificador de país, información tomada de la entidad países **
	= {dato_numerico}
dato_alfabetico	= [“a” – “z” “A” – “Z” “ ”]
dato_numerico	= [0 – 9]
direccion	= ** dirección particular del cliente **
	= nombre_calle + (numeracion_casa)
estado	= ** estado que habilita o no las operaciones al cliente. (A)ctivo, (I)nactivo y (B)aja **
	= [“A” “I” “B”]
estado_civil	= ** descripción (registro) estado civil **
	= @codigo_estado_civil + nombre_estado_civil
ESTADOS_CIVILES	= ** entidad (archivo) estado civil **
	= {estado_civil}
nombre	= ** nombre del cliente **
	= {dato_alfabetico}
nombre_calle	= ** nombre de la calle de residencia **
	= {dato_alfabetico}
nombre_ciudad	= ** nombre de la ciudad **
	= {dato_alfabetico}
nombre_estado_civil	= ** nombre o descripción del estado civil **
	= {dato_alfabetico}
nombre_pais	= ** nombre del país **
	= {dato_alfabetico}
numeracion_casa	= ** numero de la casa **
	= {dato_numerico}
pais	= ** descripción (registro) país **
	= @codigo_pais + nombre_pais
PAISES	= ** entidad (archivo) país **
	= {pais}
saldo	= ** saldo del cliente, siempre valor positivo o cero **
	= {dato_numerico}
sexo	= ** identifica al sexo del cliente **
	= [“femenino” “masculino”]
telefono	= ** teléfonos del cliente **
	= {dato_numerico}

Algunas observaciones a tener en cuenta:

- El diccionario de datos se presenta en un solo color. A modo didáctico se pintó de colores diferentes las distintas entidades para su mejor comprensión (CLIENTES, CIUDADES, PAISES y ESTADOS CIVILES).
- El uso del * (asterisco) indica un comentario pero no debe abusar de su uso. Su utilización en el ejemplo es simplemente para agregar mayor comprensión a cada definición.

- En el diccionario de datos se describen en forma alfabética la totalidad de los paquetes de datos, de ahí el orden, como un diccionario normal.
- La utilización de **id** al final de un paquete de datos significa que ese paquete de datos es una clave primaria de otra entidad, por ejemplo `codigo_pais.id`, hace referencia a la entidad país. Se relaciona con esa entidad pues necesita sacar otros datos de ella, como ser `nombre_pais`.
- Los paquetes en mayúsculas y en plural representan a las entidades (archivos) y los paquetes en minúsculas y singular las descripciones (registros).
- Por ejemplo, CIUDADES (archivo) es una colección (repetición) de ciudad (registro), por ello el uso de llaves {}.

Elemento de datos opcionales

Un elemento de datos opcionales puede estar o no presente como un componente de un elemento de datos compuesto. Se representa con la notación **()**.

Ejemplo:

`direccion_cliente` = `direccion_remision` + (`direccion_cobranza`)

Iteración

Indica la ocurrencia repetida de un componente de un elemento de datos. Se usa la notación **{ }**.

Ejemplo:

`pedido` = `nombre_cliente` + `direccion_remision` + {`detalle_pedido`}

Puede indicarse opcionalmente los límites superiores e inferiores.

Ejemplo:

`pedido` = `nombre_cliente` + `direccion_remision` + **1** {`detalle_pedido`} **12**

Selección

Elección entre un conjunto de opciones alternativas. Las opciones se delimitan por corchetes **[]** y se separan por los caracteres de barra vertical **|**.

Ejemplo:

`estado_alumno` = [`“Activo”` | `“Egresado”` | `“Graduado”`]

Sinónimos

Un sinónimo (“alias”) es un nombre alternativo para un elemento de datos. Debe tener una referencia cruzada con el nombre principal u oficial del dato. Se debe evitar el uso de sinónimos siempre que sea posible.

Ejemplo:

`cliente` = ** `alias de comprador` **

Presentación del Diccionario de Datos al usuario

- El usuario debe ser capaz de leer y comprender el diccionario de datos para confirmar el modelo.
- Probablemente, el usuario verifique la exactitud del diccionario de datos en combinación con el diagrama de flujo de datos, el diagrama de entidad relación, el diagrama de transiciones de estado o con las especificaciones de procesos, en el momento en que lee alguno de estos.

Verificaciones del Diccionario de Datos

El analista de sistemas puede realizar correcciones al DD sin ayuda del usuario, para validar que el mismo esté completo y sea consistente con los demás componentes del modelo.

Algunas preguntas que debe realizar son:

- ¿Todos los flujos del DFD están definidos en el DD?
- ¿Están definidos todos los componentes de los elementos de datos compuestos?
- ¿Se usó la notación correctamente en todas las definiciones del DD?
- ¿Existe algún elemento en el diccionario de datos que no está referenciado en los DFD, en los DER o en los DTE?

Implementación del Diccionario de Datos

- El enfoque más fácil es hacer uso de un recurso automatizado (computarizado) para introducir las definiciones del diccionario, verificar la corrección y la consistencia y producir información adecuada.
- Sin embargo, las herramientas automatizadas pueden tener algunas limitaciones:
 - Longitud de los nombres
 - Otras limitaciones en los nombres (uso de guiones, prefijos, etc.)
 - Pueden exigir la definición de atributos físicos.

Todas las exigencias de implementación son irrelevantes en esta actividad de Análisis.

Declaración de Objetivos

Página

6

Definición



Es una definición textual, concisa y breve de los objetivos del sistema.

Ejemplo:

Para el sistema de procesamiento de libros Ajax, podría ser:

Gestionar todos los detalles de los pedidos de libros, remisiones, facturación y cobranzas a clientes en mora. Debe proporcionar información sobre los pedidos de libros a otros sistemas, como marketing y contabilidad.

Haciendo un análisis de la declaración de objetivos se puede concluir cuanto sigue:

El objetivo abarca varios subsistemas, como ser:

- 1) Subsistema de pedido de libros (pedido también incluye la remisión al cliente)
- 2) Subsistema de facturación (incluye la cobranza de clientes en mora)
- 3) Subsistema de integración de datos que lo relaciona con los sistemas de marketing y contabilidad.

La declaración de objetivos no tendría por qué abarcar varios subsistemas, dependiendo de la estrategia de análisis se podría tomar cada subsistema como uno solo y desarrollarlo por separado, en cuyo caso tendremos una declaración de objetivos para cada sistema o subsistema.

Nunca se debe incluir como parte de la declaración de objetivos: “facilitar en forma rápida la información...”, “permitir la entrada sin errores de los datos...” o “sacar información útil para el usuario...”. Se da por sentado que los sistemas que serán construidos tendrán velocidad y eficiencia requerida. La declaración de objetivos debe delimitar el alcance del sistema. Todo lo que está dicho, será contemplado, de ahí su importancia de ser clara y completa.

Para un sistema de Control de Stock para una farmacia la declaración de objetivos podría ser tan simple como: *Gestionar el control del stock de los productos farmacéuticos*, o *Gestionar las entradas y salidas que afectan al stock de la farmacia*, así de simple.

El texto debe expresarse en un párrafo no mayor a cinco líneas.

Lista de Eventos

Página

7

Definición



Lista narrativa de los “estímulos” que ocurren en el mundo exterior y a los cuales nuestro sistema debe responder.

Nomenclatura

Sujeto + Verbo + Complemento (se agrega más detalles de lo que realiza el evento).

Ejemplo:

Lista parcial de eventos para un Sistema de Facturación:

- 1) Cliente compra mercaderías (A)
- 2) Cajero realiza apertura/cierre de caja (A)
- 3) Cliente anula compra de mercaderías (A)
- 4) Cliente devuelve mercaderías compradas (A)
- 5) Cajero solicita informe para rendición de caja (I/C)
- 6) Cajero consulta total de ingresos por rango de fechas (I/C)
- 7) Gerencia solicita informe de artículos vendidos por tipo de cliente (I/C)
- 8) Gerencia establece porcentaje de descuento por cantidad de productos vendidos (A)
- 9) Es tiempo de diariamente anular pedidos no confirmados por los clientes (A)

La lista de eventos debe ser enumerada, de uno en adelante. Esa enumeración no implica ninguna secuencia, menos de ejecución o importancia del evento. Se hace simplemente para referenciar al evento, identificador utilizado en otras herramientas.

Los sujetos coinciden con los terminadores (única excepción corresponden a los eventos que inician con “**Es tiempo de...**”), entes externos al sistema de información que se está modelando y que interactúa con el sistema. Se puede observar mejor con el Diagrama de Contexto en mano (este diagrama modela la relación del mundo exterior – los terminadores – con el sistema).

En cuanto a un sujeto externo, como se menciona más arriba, la única excepción se ve en el evento número 9. Es tiempo de... hace referencia al propio sistema que se está modelando y es la única forma válida de llamarlo (nos referimos al sujeto). Se utiliza cuando se quiere indicar que hay un evento que debe ejecutarse automáticamente (por el propio sistema en ejecución), que no depende de un ente externo. En nuestro ejemplo, para los pedidos de los clientes que no se confirmen, el sistema debe anularlos en forma automática dada una condición.

¿Cuándo lo hace?. Para el ejemplo del evento 9, diariamente. Se puede poner el plazo de ejecución después de “es tiempo de”, los plazos posibles pueden ser diariamente, semanalmente, mensualmente, semestralmente, anualmente o cualquier otro periodo que sea necesario. En caso de que no exista un periodo y el evento se tenga que ejecutar permanentemente (a cada instante) no se escribe nada, se omite el plazo. Por ejemplo: Es tiempo de determinar si los ingresos en caja llegaron al límite establecido e informar al supervisor de caja.

En teoría se utilizaría ese evento para informar cuándo la recaudación llega a un monto establecido, el sistema informaría vía mensaje al supervisor de cajeros en su pantalla. Esto se haría para que los supervisores no tengan que molestar a cada cajero varias veces al día para preguntarles cuánto recaudaron. El sistema, en el monitor que tiene cada supervisor de caja, estaría totalizando los ingresos de cada caja que, cuando llega a un monto establecido (monto que se ingresa en otro evento), informaría el número de caja en la pantalla del supervisor para que el mismo haga el retiro de efectivo correspondiente y así evitar que las cajas tengan mucho dinero sin necesidad.

No es recomendable que los eventos de I/C se automaticen, es decir, no deberíamos escribir es tiempo de emitir mensualmente estados de cuenta de clientes. Si bien puede considerarse como válido ya que necesariamente los estados de cuenta del cliente deben imprimirse cada fin de mes. Su automatización requerirá que el servidor de datos esté prendido, que la impresora esté prendida y en línea y finalmente, de que exista el papel suficiente para soportar la totalidad de los estados de cuenta por cliente que haya. Asociado a un “rezo” para que la impresora no tenga ningún desperfecto en el momento de la impresión o que simplemente no se tranque con el papel. Mejor sería dejar la responsabilidad a un departamento o persona y que imprima el informe cuando así lo requiera, ejemplo: Departamento de Cobranzas solicita impresión de estados de cuenta por cliente.

Un aspecto muy importante con respecto a los sujetos en los eventos es que el evento es válido si el sujeto es el originador puro del evento y no un simple intermediario. Como ejemplo citemos un evento mal redactado: **Vendedor registra compra de mercaderías realizadas por el cliente**. Si bien puede existir en la realidad un vendedor que atienda personalmente al cliente e incluso que éste sea quien cargue en el sistema la compra realizada por el cliente, no es el vendedor el sujeto válido originador del evento. El sujeto para ese evento sigue siendo el cliente. Si no existe un cliente que compre, nunca se ejecutará dicho evento. Sin embargo existe el vendedor, que no carga nada en el sistema porque no hay clientes o no hay ventas. El evento se escribe como está en el evento 1.

Se llama intermediario por ejemplo también a la tecnología. Podría el cliente comprar por teléfono, ¿escribirían Teléfono registra compra de mercaderías realizadas por el cliente?, o podría realizar el cliente compras por Internet a través de una aplicación Web. ¿A quién pondrían como sujeto?. Siempre al cliente, claro!. Independientemente de cómo hoy se realice la compra o el pedido del cliente, es él el único originador del evento (quién toma la decisión de comprar).

En la normalidad de los casos, se asocia a un evento con una opción de menú, y éste con un programa. Esto es simplemente un pensamiento a futuro puesto que en el análisis poco y nada debiera preocuparnos como nuestro modelo será llevado a cabo en la realidad (diseño / programación). Por ello es importante entender que un evento debería englobar todas las acciones posibles sobre una misma cuestión. Por ello sería incorrecto poner dos eventos en el caso de la apertura y cierre de caja. La correcta solución está en el evento 2.

Lo mismo sería para los datos personales del cliente. No haría cuatro eventos, para alta, baja, modificación y consulta de datos. Con uno sólo es más que suficiente. El evento correcto sería Cliente mantiene datos personales.

Es lógico pensar que si un evento hace (agrega información al sistema), exista otro evento que deshaga lo hecho, en la mayoría de los casos. Para el evento 1 existe el evento 3. El primero permite registrar las compras realizadas por el cliente y el tercero permite anular una compra ya registrada en el sistema. Decimos que es lógico siempre y cuando el analista de sistemas lo considere así, a pedido del usuario.

Al final de cada evento se escribe una (A) o una (I/C), que indica A de actualización e I/C de informe o consulta. Es simplemente a modo didáctico y no se escriben. Sirve para enfatizar los eventos que actualizan información de los que simplemente consultan datos y los muestran por pantalla o los imprimen.

Los eventos de actualización necesariamente afectan a los almacenes, creando, borrando o modificando datos.

Lista de eventos incorrectos

Siguiendo el ejemplo de la lista parcial de eventos para el Sistema de Facturación, agregamos:

- 10) Es tiempo de actualizar artículos por venta de mercaderías
- 11) Cajero imprime factura contado por la compra de cliente
- 12) Supervisor genera asientos contables por las ventas realizadas
- 13) Gerente de ventas devuelve mercaderías vencidas al proveedor

El evento 10 es incorrecto pues la actualización de las mercaderías se realiza inmediatamente después de confirmarse la compra (con el evento 1) y en forma automática (lo hace el sistema), incluso en el mismo programa. No es correcto que lo pongamos como un evento por separado pues no podrá nunca ejecutarse por sí solo ya que depende de otro evento. Claramente no es un evento.

El evento 11 es parecido al anterior. La factura se imprime “sola” después de confirmar la compra del cliente con el evento 1. No debe depender la impresión de la factura de la decisión de algún usuario (terminador). Si lo que quieren decir es que se REIMPRIME la factura, es otra cosa. Es posible poner una opción en el menú que permita a los cajeros reimprimir las facturas que por algún motivo no salieron, por ejemplo se tranca la impresora en el momento de imprimir y sale mal. En ese caso claramente el evento debe decir eso: Cajero reimprime factura de compra de mercaderías. Esto se haría para salvar el inconveniente de la mala o nula impresión.

El evento 12 está mal ya que se trata de una “invasión” a otro sistema que en éste momento no lo estamos modelando. Desde el sistema de facturación queremos generar asientos contables (registros) en otro sistema, el de Contabilidad. No forma parte del objetivo del sistema de facturación. La responsabilidad del sistema de facturación o lo que debiera abarcar sería contar con todos los datos para que luego, el Sistema de Contabilidad, solicite al sistema de facturación los datos y genere desde ahí (desde contabilidad) los asientos correspondientes. No nos metamos a otros sistemas, bastante ya tenemos con el nuestro.

El evento 13 habla de mercaderías vencidas y su devolución al proveedor. No forma parte del objetivo del sistema de facturación. Sale fuera del alcance del sistema. Podría tratarse de un evento válido pero para el sistema de control de stock, por ejemplo.

Los eventos pueden ser de varios tipos:

a) Orientados por flujos

Asociados a un flujo de datos, es decir, el sistema toma conocimiento de la ocurrencia de un evento por la llegada de uno o más grupos de datos.

No existe necesariamente una correspondencia uno a uno entre los flujos de datos del diagrama de contexto y los eventos de la lista de eventos. En general, todo flujo de datos es la indicación de que ocurrió un evento o es exigido por el sistema con la finalidad de procesar un evento.

Ejemplo:

Proveedor remite factura

Usuario mantiene datos personales

b) Temporales

Disparados en un determinado momento (al cumplirse un plazo determinado).

Los eventos temporales no son disparados por flujos de datos de entrada, pero pueden solicitar entradas de uno o más terminadores, a través de flujo de datos.

Ejemplo:

Es tiempo de diariamente anular orden de producción

Es tiempo de informar nuevo saldo de cliente vía mensaje por celular

c) De control

Son casos especiales de eventos temporales: un estímulo externo que ocurra en un momento imprevisible.

Un evento de control está asociado a un flujo de control en el diagrama de contexto.

Un flujo de control puede ser considerado un flujo de datos binario: está ON u OFF y puede pasar de un estado a otro en cualquier momento, informando al sistema que debe ejecutar alguna acción inmediata.

Ejemplo:

Es tiempo de registrar peso de camión por datos suministrados vía báscula

Es tiempo de sacar fotografía a vehículo por cruce de luz roja por datos suministrados por sensor infrarrojo

Especificación de Procesos

Página
11

Definición

En caso de «expresión» haga
Caso «opción 1»:
 «instrucciones»
Caso «opción 2»:
 «instrucciones»
Caso «opción 3»:
 «instrucciones»
...
Caso «opción N»:
 «instrucciones»
SINO «instrucciones a realizar si no
se ha cumplido ninguna de
las condiciones anteriores»
Fin-Caso

Descripción textual de lo que debe hacerse para transformar los flujos de entradas en flujos de salidas.

Podemos usar cualquier método, siempre que satisfaga algunos requisitos:

Características de una buena especificación

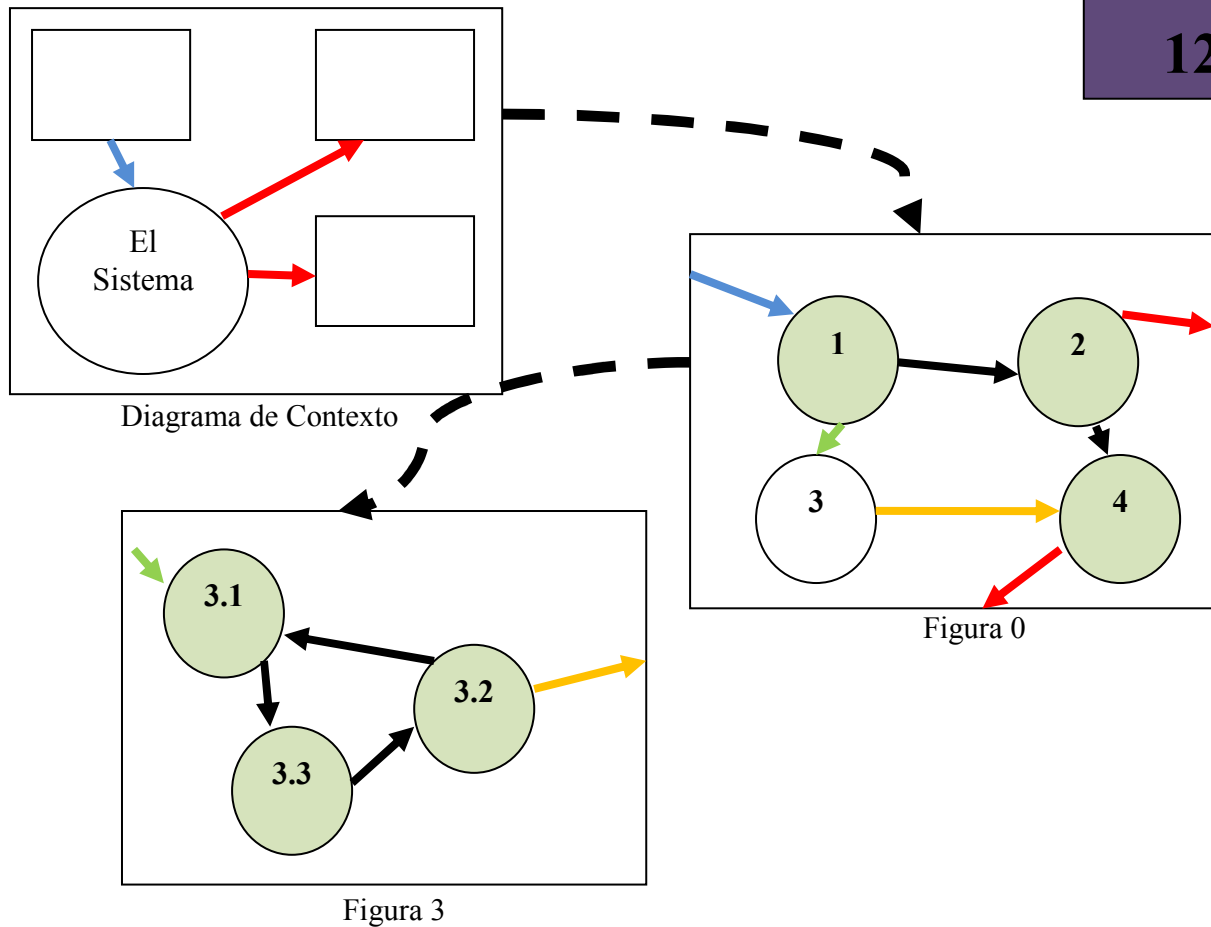
- Debe ser clara, no ambigua, concisa y completa
- Debe poder ser verificada por el usuario y por el analista de sistemas
- Debe expresarse de forma que pueda comunicarse en forma efectiva a las diferentes audiencias involucradas
- Debe ser lógica, no física. Es decir, no debe imponer (o implicar) decisiones arbitrarias de diseño e implementación.

Implementación de las especificaciones de procesos

- Solo deben desarrollarse especificaciones de procesos para las burbujas (procesos) de nivel más bajos (primitivas) del DFD.
- Todos los procesos de un determinado nivel son definidos por la red de procesos de nivel inferior siguiente. En otras palabras, la especificación de procesos para una burbuja de nivel es el DFD del nivel inmediatamente inferior. Escribir una especificación adicional de procesos en lenguaje estructurado sería no solamente superfluo sino también redundante; esto es, crearía una especificación que sería más difícil de mantenerse actualizada.

Veamos el siguiente ejemplo:

Página
12



Así como se muestra en el ejemplo de arriba, se realizarán especificaciones de procesos para las burbujas 1, 2, 3.1, 3.2, 3.3 y 4, ya que corresponden a las **burbujas primitivas**, presumiendo que las burbujas 1, 2 y 4 de la figura 0 no se descomponen.

Supongamos que la burbuja 3.2 se vuelve a descomponer en las burbujas 3.2.1, 3.2.2, 3.2.3 y 3.2.4, entonces, se necesitará realizar especificaciones de procesos para las burbujas 1, 2, 3.1, 3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.3 y 4.



Para pensar

- En el DD se deben describir los paquetes de datos que fluyen en el sistema. Un informe que es emitido por un terminador y que luego es remitido a otro terminador (flujo de salida), ¿deberá definirse también en el DD?. ¿Cómo?
- ¿Sería una buena estrategia como analista de sistemas lograr que la declaración de objetivos contemple la menor cantidad de subsistemas?
- ¿La lista de eventos, para un sistema promedio, supera 100, 200 o hasta 300 eventos?. ¿No son muchos?, ¿De dónde sacamos tantos eventos?
- ¿La especificación de procesos no debería ser un trabajo del programador?