

6.3. Diccionario de datos



6.1. Introducción - Visión panorámica del AE

6.2. Diagramas de flujo de datos

6.3. Diccionario de datos

6.4. Modelado de la lógica de los procesos

6.5. Modelado de datos

6.6. Historia de vida de las entidades

6.7. El proceso de Yourdon

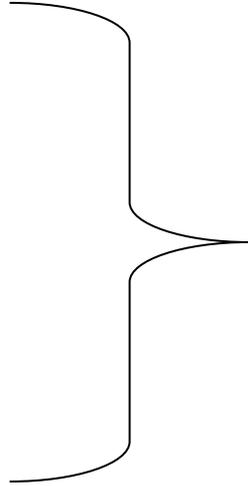
Diccionario de datos (DD)



- “Es un conjunto de información (datos) sobre datos”
- Objetivos del DD:
 - Glosario de términos
 - Establecer terminología estándar
 - Proporcionar referencias cruzadas
 - Proporcionar control centralizado para cambios
- Evolución histórica: desde el directorio/diccionario de datos hasta el diccionario de recursos de información

Posibles elementos para definir en el DD

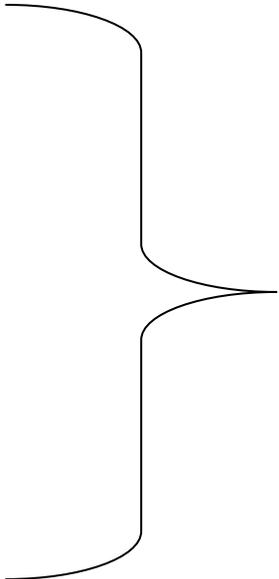


- Flujos de datos
 - Procesos
 - Ficheros
 - Entidades externas
 - Estructuras de datos
 - Datos elementales
 - Cualquier otra cosa que el analista considere conveniente
- 
- Mínimo necesario

Información requerida para cada elemento del DD



- Nombre
- Tipo de elemento
- Breve descripción
- Sinónimos
- Observaciones



Mínimo necesario

Información requerida para cada elemento del DD (II)



- Frecuencias y fechas
- Volúmenes (Ks estimadas, n° líneas impresas, etc.)
- Cuellos de botella, valores máximos y mínimos (tablas, ficheros, impresos, entradas de datos)
- Referencia o código de impreso
- Rango de valores permitido y clase (numérico, alfanumérico, etc.)
- Miniespecificaciones (sólo procesos)
- Referencias cruzadas
- Usuarios afectados
- Cualquier otra información que se considere de interés

Soporte del DD



- Manual
- Editor/procesador de textos
- Base de datos
- **Automático e integrado
(sw. específico)**

Descomposición top-down de datos



$$A = B + C$$

$$B = B1 + B2 + B3$$

$$C = C1 + C2$$

A, B, C, B1, B2, B3, C1, C2

todos están definidos en el DD

■ Ejemplos de descomposición:

- Ficheros en “subficheros” o registros
- Procesos en subprocessos
- Flujos en “subflujos”
- Estructuras de datos en datos elementales

Operadores relacionales

- "=" — es equivalente a
- "+" — y
- "<>" — o (inclusivo: al menos una de las opciones)
- "[]", "|" — o (exclusivo: sólo una de las opciones)
- "1{ }N" — iteraciones entre 1 y N veces del término entre llaves
- "()" — opcional

Operadores relacionales (II)

- Actualmente (Yourdon 93) " $\langle \rangle$ " no se usa
(en System Architect tampoco)
- Se utiliza "[]" , "|" con combinaciones de "(")" y "+"
- Ejemplos:
 - direccion-cliente = \langle direccion-envio, direccion-facturacion \rangle
 - * se puede expresar como *
 - dirección-cliente = [direccion-envio | direccion-facturacion |
direccion-envio + direccion-facturacion]
 - * si se admite que direccion-cliente esté vacío *
 - direccion-cliente = (direccion-envio) + (direccion-facturacion)

Operadores relacionales (III)

■ "*" ... "*" — comentario

■ @ — identificador de campo clave en un almacén (también, alternativamente, se puede subrayar la clave)

■ Ejemplos:

Solicitud-destino = @nºascensor + (nºplanta)
= nºascensor + (nºplanta)

* ambas definiciones son equivalentes *

Ejemplos DD



pedido = cupon-correos + (pago-previo)

etiqueta = 1{carácter}8

nº-de-telefono =

*cualquier secuencia correcta de dígitos que
provoca una llamada *

[extension-local | 9 + numero-exterior]

extension-local = * sólo dentro del edificio *

primer-digito + 3{ cualquier-digito}3

primer-digito = [1|2|3|4|5|6|7]

cualquier-digito = [0|1|2|3|4|5|6|7|8|9]

¿Hasta cuándo especificar?

- El proceso de descomposición finaliza en los términos autocontenidos
- Ejemplo
 - persona = apellidos + nombre + n^oss + edad
 - ¿ "edad" es autocontenido?
 - edad = 1{digito}2

Sinónimos



- Origen:
 - Distintos usuarios dan distintos nombres a los mismos objetos
 - El analista introduce, **por error**, un nombre distinto para un objeto ya nombrado
 - Distintos analistas que trabajan en el mismo proyecto dan nombres distintos a un mismo objeto
- Los sinónimos deben evitarse siempre que sea posible

Ejemplos DD (II)



Nombre: hoja-verde

Sinónimos: petición, solicitud

Tipo: sinónimo

Observaciones:

Nombre: estado

Sinónimos: estado-cliente, EST\$

Tipo: elemento de datos

Valores y significado:

OK.- Cuenta en buen estado

C.- Cuenta cerrada

D.- Cuenta en "números rojos" * cliente moroso *

Observaciones:

Ejemplos DD (III)



Nombre: peticion

Sinónimos: solicitud, hoja-verde

Tipo: flujo de datos

Composición: [peticion-estado-cliente | peticion-stock | peticion-estado-de-un-pedido | petición-de-materia-prima]

Pertenece a: * ninguno *

Observaciones:

Nombre: Contabilidad de proyectos

Sinónimos: Cuentas

Tipo: fichero

Composición: { @nº-de-proyecto + descripción-proyecto + cuenta-del-gabinete + { nombre-del-empleado + fecha-ingreso } }

Organización: * secuencial, por número de proyecto *

Observaciones:

Elementos y estructuras de datos



- Son la base sobre la que se definen los flujos de datos, los almacenes y las entidades del diagrama E/R.
- Un elemento de datos es una pieza de información atómica.
- Una estructura de datos es un registro, compuesto por otras estructuras o elementos de datos.

Ejemplos DD (IV)

- (Flujo de datos) “Libros prestados” =
[“Libros entregados” |
“Libros devueltos”]

donde “Libros entregados” y “Libros devueltos” son **estructuras de datos**.

- “Libros entregados” = { ISBN + Copia-ID }

- “Libros devueltos” = { ISBN + Copia-ID }

donde ISBN y Copia-ID son **elementos de datos**

6.4. Modelado de la lógica de los procesos



6.1. Introducción - Visión panorámica del AE

6.2. Diagramas de flujo de datos

6.3. Diccionario de datos

6.4. Modelado de la lógica de los procesos

6.5. Modelado de datos

6.6. Historia de vida de las entidades

6.7. El proceso de Yourdon

Miniespecificaciones (ME)



- Proceso primitivo \Rightarrow miniespecificación
- La ME describe las reglas sobre cómo realizar el proceso para transformar las entradas en salidas
- La ME indica el proceso a realizar, la transformación de datos, no el algoritmo (que se selecciona en el proceso de diseño)

Herramientas para describir la lógica de los procesos



- Lenguaje estructurado
 - Tablas de decisión
 - Árboles de decisión
 - Pre y post-condiciones
- (son alternativas no excluyentes)

Lenguaje estructurado



- Vocabulario (restringido) de una lengua (español, inglés, etc.)
 - Verbos imperativos
 - Términos definidos en el DD
 - Palabras reservadas para formulación lógica (mayúsculas)
- Sintaxis de la programación estructurada

Lenguaje estructurado (II)



- Los objetos de una ME (sujetos de las sentencias) serán términos del DD o bien términos locales
- Los términos locales se definen explícitamente dentro de una ME, y son conocidos, relevantes y significativos sólo dentro de esa ME (por tanto, no es imprescindible su inclusión en el DD)
- Ejemplo:
variables utilizadas para cálculos intermedios, como sumas parciales, dentro de un proceso.

Lenguaje estructurado - Sintaxis



- Sentencia declarativa simple (secuencia)
- Estructura de decisión
- Estructura de repetición
- Combinaciones de las estructuras anteriores

Sentencias declarativas



- Concisión
- Evitar verbos ambiguos (manejar, realizar, procesar, etc.)
- Utilizar verbos precisos que describan acciones concretas (imprimir, enviar, acumular...)
- Mencionar expresamente el objeto de la sentencia, preferiblemente utilizando los términos del DD
- Ejemplos:
 - Recoger INF-CLIENTE
 - Separar PETICION
 - Archivar PETICION en F-PETICION *fichero*
 - Enviar DATOS-CLIENTE a DPTO-CLIENTES

Estructura de decisión

SI Condición

Acción(es)

CASO Condición:Acción(es)

SINO

Acción(es)

■ Ejemplos:

a) SI Valor-capital-actual es menor que 600€

Asignar Cantidad-depreciada = Valor-capital-actual = 0

SINO

Asignar Cantidad-depreciada = 10% de Valor-capital-actual

b) Seleccionar la política que se aplica:

Caso 1: (Costo-de-pedido > 1000€) :

enviar por avión

Caso 2: (Costo-de-pedido entre 100€ y 1000€) :

enviar por correo urgente

Caso 3: (Costo-de-pedido < 100€) :

enviar por correo normal

Estructura repetitiva



REPETIR (condición de selección)

Acción(es)

HASTA (condición de terminación)

MIENTRAS (condición)

Acción(es)

FIN MIENTRAS

■ Ejemplo:

REPETIR para cada registro-de-pasajero en fichero-de-reservas

Acumular Cantidad-debida en Total

Construir registro Nuevo-débito

Escribir Nuevo-débito en el diario

HASTA final de fichero-de-reservas

Estructura repetitiva (II)

a) PARA CADA cliente en fichero-cuentas

Acceder al registro de cuenta del fichero-cuentas

Si estado-cuenta es moroso y balance < 10

Poner estado-cuenta en pendiente

Acumular balance-cuenta en total-pendiente

Asignar a fecha-última-transacción la fecha de hoy

b)

REPETIR para cada cliente en fichero-cuentas

Acceder al registro de cuenta del fichero-cuentas

Si estado-cuenta es moroso y balance < 10

Poner estado-cuenta en pendiente

Acumular balance-cuenta en total-pendiente

Asignar a fecha-última-transacción la fecha de hoy

HASTA que no haya más clientes

Lenguaje estructurado - Observaciones



- Utilizar “funciones” o “subrutinas”
- Subrayar los términos del DD o usar con mayúsculas
 - en SA, usar comillas
- Evitar sentencias largas e imprecisas
- Usar indentación o notación de bloque
- Usar paréntesis para las combinaciones de condiciones lógicas (and, or, not)

Lenguaje estructurado.

Ejemplos (I) *(Yourdon 93) Apéndice F*

PROCESO 3.1: PRODUCIR RECIBOS EFECTIVO

COMIENZA

efectivo-recolectado = 0

MIENTRAS haya más registros en **DINERO**

LEER siguiente registro en **DINERO**

ENVIAR **dinero** *en (Yourdon 93) pone "DESPLEGAR"*

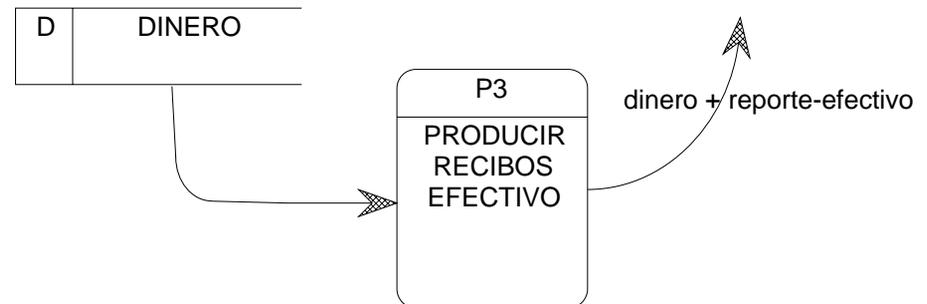
efectivo-recolectado = efectivo-recolectado + **cantidad-dinero**

FIN-MIENTRAS

reporte-efectivo = efectivo-recolectado

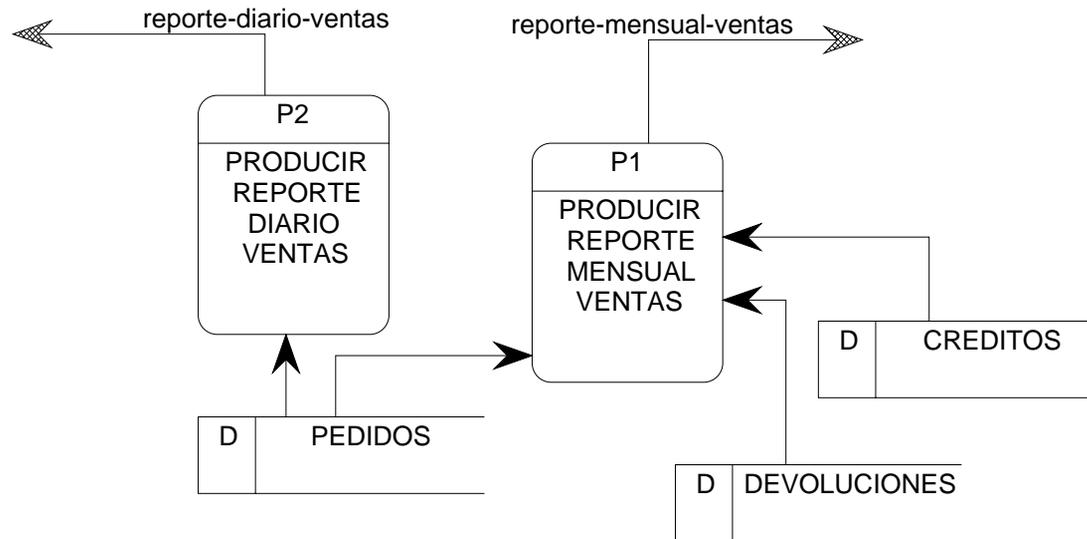
ENVIAR **reporte-efectivo**

TERMINA



Lenguaje estructurado.

Ejemplos (II) (Yourdon 93) Apéndice F



Lenguaje estructurado.

Ejemplos (III) (Yourdon 93) Apéndice F

PROCESO 3.2: PRODUCIR REPORTE DIARIO VENTAS

COMIENZA

total-diario = 0

MIENTRAS haya más **pedido** en **PEDIDOS** con **fecha-pedido** = fecha actual

 LEER siguiente **pedido** con **fecha-pedido** = fecha actual

 SUMAR **numero-factura, nombre-cliente, nombre-compañía, pedido-total**
 como nuevo renglón en **reporte-diario-ventas**

 SUMAR **total-pedidos** a total-diario

FIN_MIENTRAS

SUMAR total-diario como nuevo renglón en **reporte-diario-ventas**

ENVIAR **reporte-diario-ventas**

TERMINA

Lenguaje estructurado.

Ejemplos (IV) (Yourdon 93) Apéndice F

PROCESO 3.3: PRODUCIR REPORTE MENSUAL VENTAS

COMIENZA

total-ventas = 0

total-devoluciones = 0

total-créditos = 0

MIENTRAS haya más **pedido** en **PEDIDOS** con **fecha-pedido** de este mes

SUMAR **total-pedidos** a **total-ventas**

FIN_MIENTRAS

MIENTRAS haya más **devolución** en **DEVOLUCIONES** con **fecha-devolución** de este mes

SUMAR **valor-devolución** a **total-devoluciones**

FIN_MIENTRAS

MIENTRAS haya más **crédito** en **CREDITOS** con **fecha-crédito** de este mes

SUMAR **monto-de-crédito** a **total-créditos**

FIN_MIENTRAS

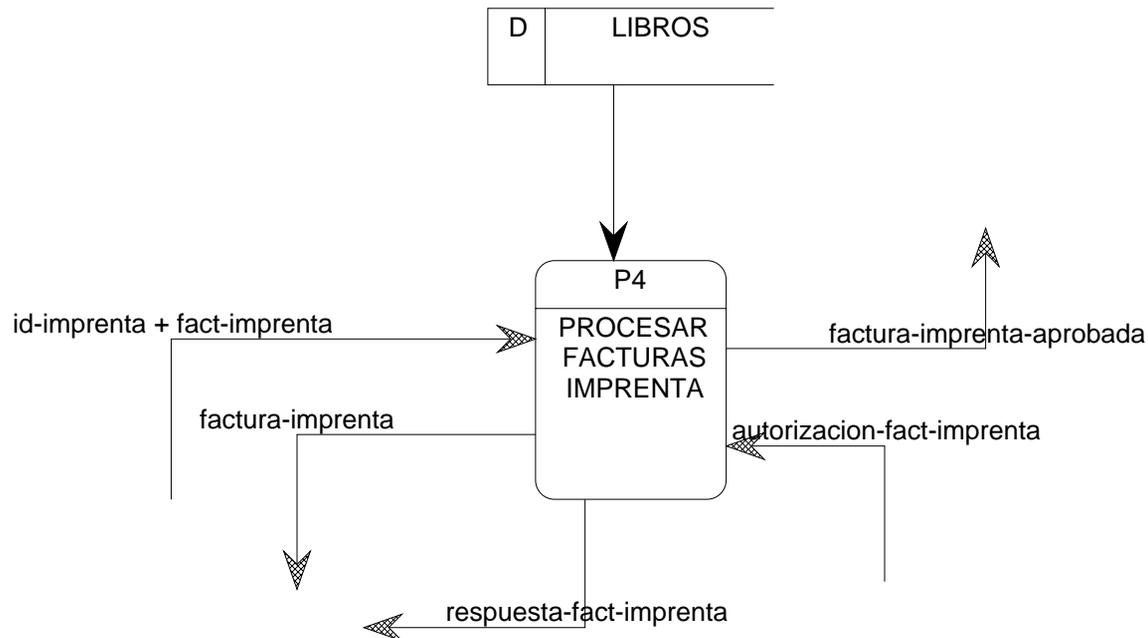
reporte-mensual-ventas = total-ventas, total-devoluciones, total-créditos

ENVIAR **reporte-mensual-ventas**

TERMINA

Lenguaje estructurado.

Ejemplos (V) *(Yourdon 93) Apéndice F*



Lenguaje estructurado.

Ejemplos (VI) (Yourdon 93) Apéndice F

PROCESO 4.4: PROCESAR FACTURA IMPRENTA

COMIENZA

ENCONTRAR **libro** en **LIBROS** con **clave-libro** que corresponda con **clave-libro** en **fact-imprensa**

SI no se encuentra registro

respuesta-fact-imprensa = "No existen pedidos pendientes para este libro"

ENVIAR **respuesta-fact-imprensa**

OTRO

ENVIAR **factura-imprensa** (a administración para su aprobación)

ACEPTAR **autorización-factura-imprensa**

SI **autorización-factura-imprensa** = "NO"

respuesta-fact-imprensa = "Factura rechazada; comuníquese con la administración para discutirlo"

ENVIAR **respuesta-fact-imprensa**

OTRO

respuesta-factura-imprensa = "Factura aceptada"

ENVIAR **respuesta-factura-imprensa**

factura-imprensa-aprobada = **fact-imprensa**

ENVIAR **factura-imprensa-aprobada**

FIN_SI

FIN_SI

TERMINA

Lenguaje estructurado.

Ejemplos (VII) (Yourdon 93) Apéndice F

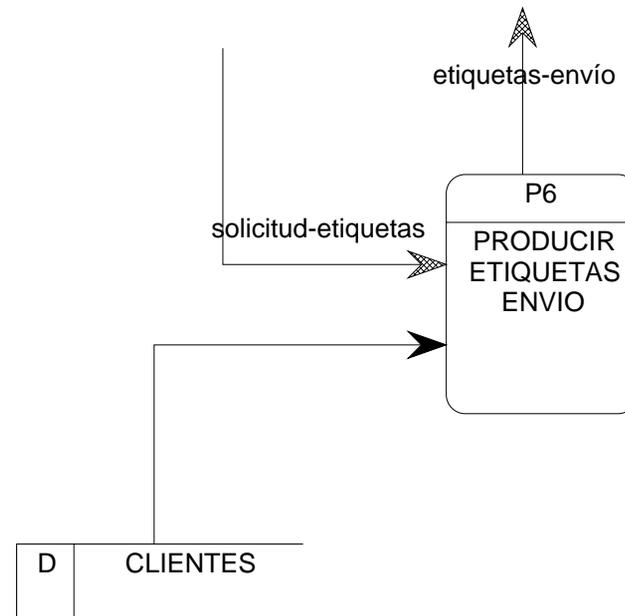
PROCESO 6.1: PRODUCIR ETIQUETAS ENVIO

COMIENZA

ORDENAR **CLIENTES** por **código-postal** en **etiquetas-envío**

ENVIAR **etiquetas-envío**

TERMINA



Tablas de decisión

Encabezamiento	Reglas		
Estados de condición			
Sentencia de acción			

Anotación de condición

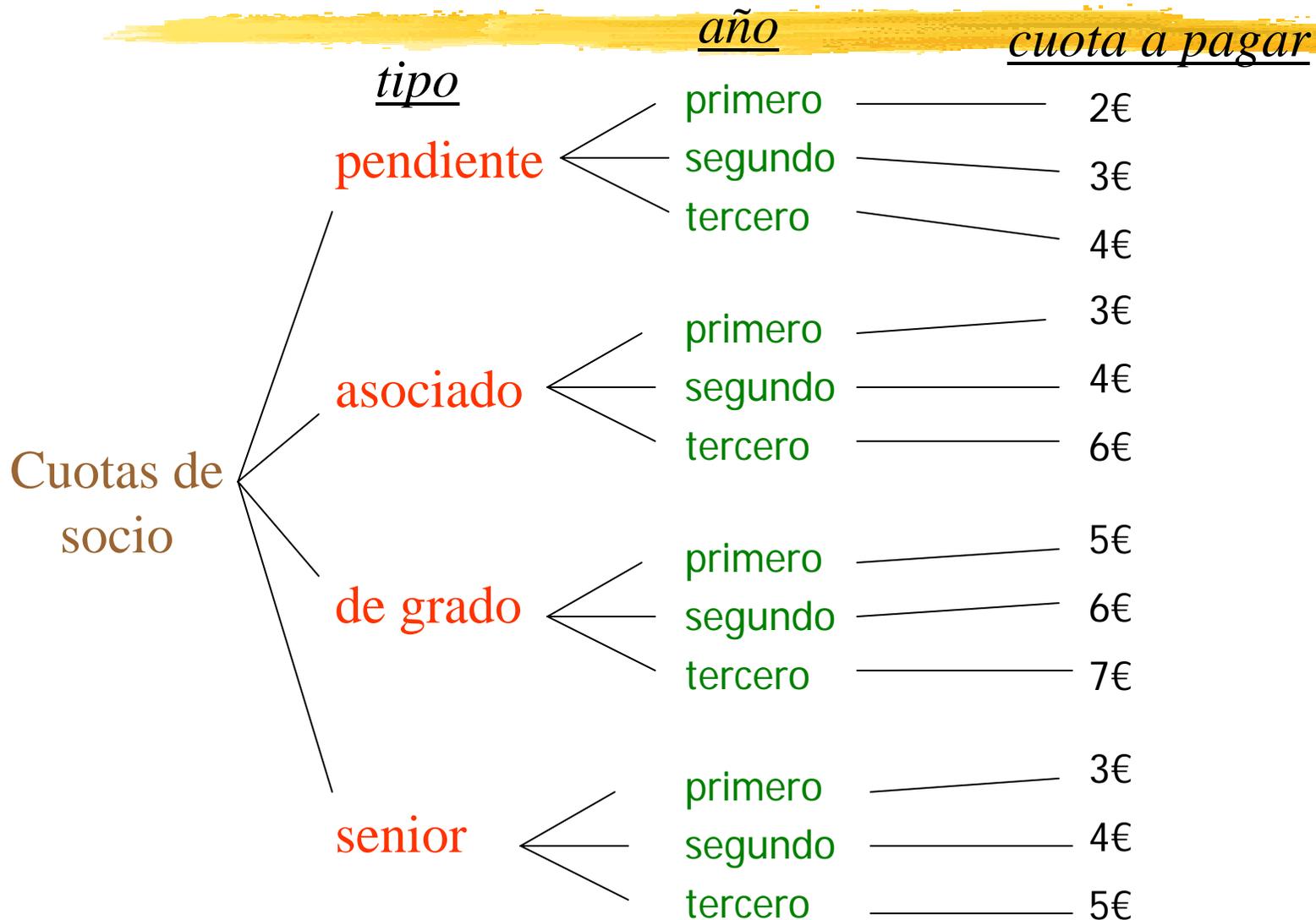
Anotación de acción

Autorización de tarjeta de crédito	1	2	3	4
Compra inferior a 50€	Y	N	N	N
Compra entre 50 y 100€		Y	N	N
Compra superior a 100€			Y	N
Autorizado automáticamente	X			
Dar número de autorización		X	X	
Anotar en la cuenta			X	
Error				X

Autorización de tarjeta de crédito	1	2	3
Valor de la compra p	$p > 100€$	$50€ < p < 100€$	$0 < p < 50€$
Autorizar automáticamente			X
Asignar autorización	X	X	

Se han desarrollado procesadores de tablas de decisión que generan automáticamente el código del proceso correspondiente.

Árboles de decisión



Comparativa

Uso	AD	TD	Lenguaje estructurado	Lenguaje narrado simplificado
<i>Verificación lógica</i>	Moderada	Muy buena	Buena	Moderada
<i>Visualización de la estructura lógica</i>	Muy buena (pero sólo decisiones)	Moderada (sólo decisiones)	Buena (para todo)	Moderada (para todo, pero depende del autor)
<i>Simplicidad</i>	Muy buena	Muy pobre	Moderada	Buena
<i>Validación por el usuario</i>	Buena	Pobre (si el usuario no está formado en TD)	Pobre-Moderada	Buena
<i>Especificación de programa</i>	Moderada	Muy buena	Muy buena	Moderada
<i>Editado por la máquina</i>	Pobre	Muy buena	Moderada (necesita sintaxis)	Pobre
<i>Cambios</i>	Moderada	Pobre	Buena	Buena

Pre y post-condiciones (Yourdon 93)



- Útiles para representar la acción a realizar sin entrar en los detalles del algoritmo
- Particularmente útiles cuando:
 - El usuario tiene tendencia a describir el proceso en términos de un algoritmo particular
 - El analista está razonablemente seguro de que existen muchos algoritmos alternativos
 - El analista desea que el diseñador/programador explore varios algoritmos, pero no quiere enredarse con el usuario en discusiones acerca del mérito relativo de cada uno

Precondiciones

■ Entradas disponibles

- "llega el dato X" * en (Yourdon 93) pone "ocurre" *

■ Relaciones entre las entradas

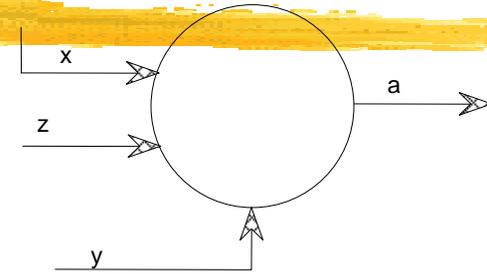
- "llegan detalles de pedido y detalles de envío con el mismo número de cuenta"
- "llega un pedido con fecha de entrega de más de 60 días"

■ Relaciones entre entradas y almacenes

- "hay un pedido-de-cliente con número-de-cta-de-cliente que corresponde con un número-de-cta-de-cliente del almacén de clientes"

■ Relaciones entre almacenes distintos (o dentro del mismo almacén)

- "hay un pedido en el almacén de pedidos cuyo número-de-cta-del-cliente corresponde con un número-de-cta-del-cliente en el almacén de clientes"
- "existe un pedido en el almacén de pedidos con fecha-de-envío igual a la fecha actual"



Post-condiciones



- Salidas producidas
 - “se producirá una factura”
- Relaciones entre entradas y salidas
 - “la **factura-total** se calcula como suma de **precios-unitarios-de-artículos** más **costos-de-envío**”
- Relaciones entre salidas y almacenes
 - “el **balance-actual** en el almacén **INVENTARIO** se incrementará con **cantidad-recibida**, y el nuevo **balance-actual** se producirá como salida de este proceso”
- Cambios en los almacenes
 - “el **pedido** se anexará al almacén de **PEDIDOS**”
 - “el registro de **clientes** se eliminará del almacén de **clientes**”

Pre y post-condiciones

Ejemplos (Yourdon 93)



- ESPECIFICACIÓN DE PROCESO 3.5: CALCULAR EL IMPUESTO SOBRE VENTAS
 - Precondición 1
 - Llega DATOS-VENTA con TIPO-ITEM que corresponde con CATEGORÍA-ITEM en CATEGORÍAS-IMPUESTO
 - Postcondición 1
 - IMPUESTO-SOBRE-VENTA se hace igual a $\text{MONTO-VENTA} * \text{IMPUESTO}$
 - Precondición 2
 - Llega DATOS-VENTA con TIPO-ITEM que no concuerda con CATEGORÍA-ITEM en CATEGORÍAS-IMPUESTO
 - Postcondición 2
 - Se genera mensaje de error

Pre y post-condiciones

Ejemplos (II) (Yourdon 93)



■ Precondición 1

- El comprador llega con un número-de-cta que corresponde con un número de cuenta en CUENTAS, cuyo código-de-status es "válido"

■ Postcondición 1

- Se produce una factura con número-de-cuenta y monto-de-venta

■ Precondición 2

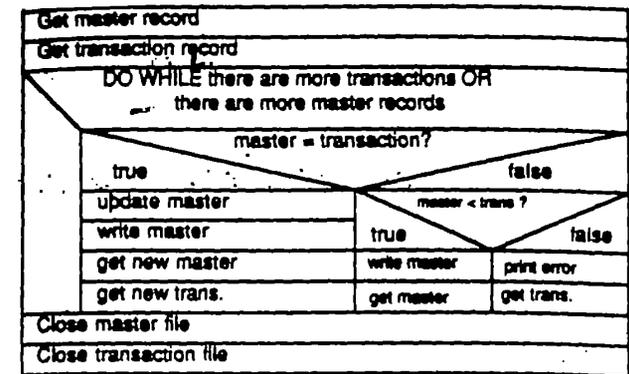
- La precondición 1 falla por algún motivo (el número-de-cta no se encuentra en CUENTAS, o el código-de-status no es "válido")

■ Postcondición 2

- Se produce un mensaje de error

ME - Otras técnicas

- Grafos y diagramas propios del usuario
 - Si son claros, se pueden agregar a la especificación como redundantes.
- Diagramas Nassi-Shneiderman



- Flowcharts

- Lenguaje narrativo

No recomendadas



Sirve para descripción breve

6.5 Modelado de datos



6.1. Introducción - Visión panorámica del AE

6.2. Diagramas de flujo de datos

6.3. Diccionario de datos

6.4. Modelado de la lógica de los procesos

6.5. Modelado de datos

6.6. Historia de vida de las entidades

6.7. El proceso de Yourdon

Objetivo

- Obtener una representación de la información del sistema independiente de aplicaciones y dispositivos físicos
 - ⇒ Facilitar cambios en los requisitos, SGBD, equipos físicos
- Con el análisis estructurado moderno de Yourdon el modelado de datos cobra la misma importancia que el modelado de procesos.
- Técnicas de modelado de datos en AE:
 - E/R \leftarrow *RECOMENDADO*
 - DED (Diagramas de Estructura de Datos)



Representa esquemas
relacionales, jerárquicos, CODASYL

DED

- BD lógica, no simplificada ni optimizada, a efectos de validación por el usuario (esta especificación pasaría al implementador de la BD)



Diseño externo, lógico

- BD optimizada y normalizada, lista para ser implementada físicamente



Diseño físico

DED (II)

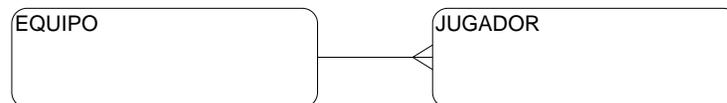
■ "E/R limitado"

■ Sólo interrelaciones de grado 2

- Ternarias: descomponer

■ Cardinalidad sólo 1:N

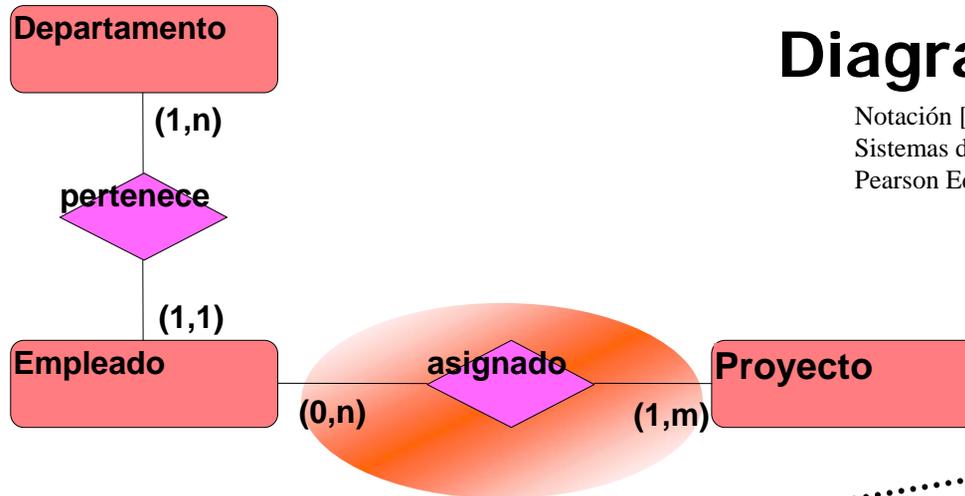
- Otras cardinalidades:
- Cardinalidad 1:1
 - Agrupar las dos entidades
 - Conservar las dos entidades, con una interrelación en cualquier sentido
- Cardinalidad M:N
 - Entidad auxiliar con dos relaciones 1:N



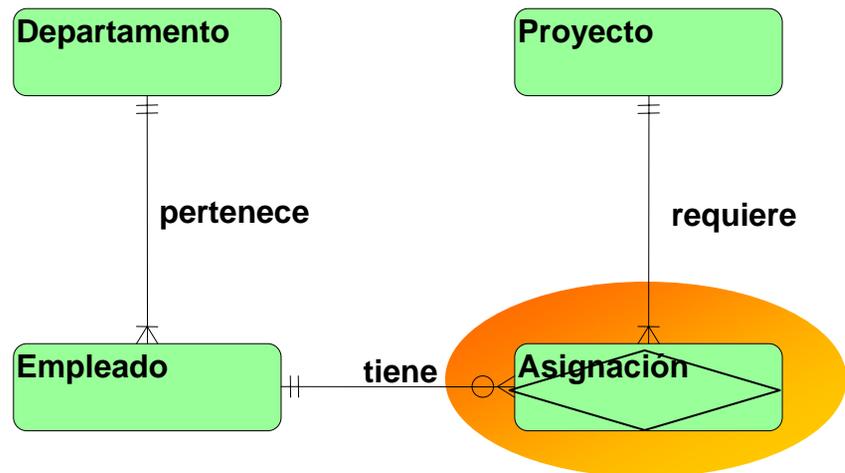
DED. Ejemplo

Diagrama E-R

Notación [EN2002] (Chen) Elmasri, R.; Navathe, S.B. : "Fundamentos de Sistemas de Bases de Datos". 3ª Ed. Madrid [etc.]: Addison-Wesley, Pearson Educación, 2002.



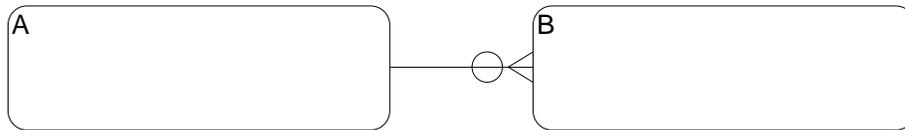
DED



DED. Interrelaciones

■ Interrelaciones OPCIONALES

- Interrelación opcional en el extremo B y obligatoria en el A



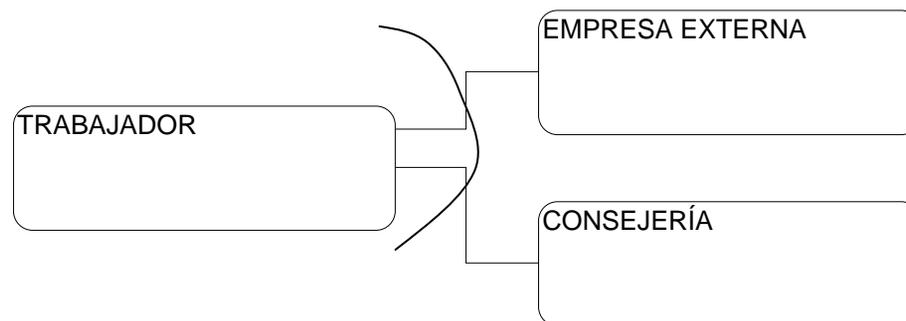
- “ \forall ocurrencia de A pueden \exists o no una o varias ocurrencias de B, y para cada ocurrencia de B existe una ocurrencia de A asociada”

DED. Interrelaciones (II)

■ Interrelaciones EXCLUSIVAS

- Dos o más interrelaciones entre varias entidades son exclusivas si la existencia de una implica la no existencia de la otra

– P.ej. En la CARM...



(notación De Miguel Piattini)

6.6. Historia de vida de las entidades



6.1. Introducción - Visión panorámica del AE

6.2. Diagramas de flujo de datos

6.3. Diccionario de datos

6.4. Modelado de la lógica de los procesos

6.5. Modelado de datos

6.6. Historia de vida de las entidades

6.7. El proceso de Yourdon

HVE. Bibliografía



Para todo este apartado:

- Guía de técnicas de Métrica v.2.1.
Ministerio de Administraciones Públicas.
1996.

(HVE no es mencionada en Métrica 3)

HVE



- Describe la posible evolución de las entidades de datos del sistema
- VISION DEL COMPORTAMIENTO, que complementa:
 - Visión estática (E/R o DED)
 - Visión de procesos y flujos (funcional) (DFDs)
- HVE se basa en entidades de datos (identificadas en DED), y transacciones o eventos (de los DFDs)
- Deben ser coherentes HVE, DED, DFD

HVE. Objetivos



- Registrar la secuencia de los cambios de las entidades en el tiempo
 - Determinar los estados posibles
 - Determinar los cambios de estado
 - Identificar interacciones producidas por eventos

HVE



- (En principio) existe una HVE por cada entidad del sistema
 - ¿Realmente es necesario?
- HVE describe la “sucesión de eventos” que afectan a dicha entidad, cuyos efectos pueden ser:
 - Crear/dar de baja a la entidad
 - Modificar sus atributos

HVE. Elementos



■ Entidades de datos

- Cualquier objeto sobre el que el sistema guarda información (tienen atributos)

■ Eventos

- Sucesos que activan un proceso que afecta los datos del sistema

■ Efectos

- Resultado de la acción de un evento sobre una entidad

■ Nodo

- Agrupación de eventos en una caja

■ Cajas vacías

- Representan el caso en que ningún evento afecta a la entidad

Eventos



- EXTERNOS. Por activación externa
 - ej. solicitudes de alta, baja, modificación, etc.
- PERIÓDICOS. Activación dependiente del tiempo (automáticos) sin estímulo externo
 - ej. "back up" periódico
- TRIGGERING. Activados internamente por cumplimiento de determinadas condiciones
 - ej. alarma activada

Efectos



- Un evento puede tener distintos efectos sobre entidades diferentes.
 - Ej.: SOLICITAR APERTURA CTA. BANCARIA
 - Crea (o actualiza) entidad CLIENTE
 - Crea entidad CUENTA
- Un evento puede tener efectos distintos sobre ocurrencias de una misma entidad.
 - Ej.: entidad CUENTA; ev. REALIZAR TRANSFERENCIA
 - Efectos: para una ocurrencia: HACER APUNTE EN EL DEBE
 - Para la otra ocurrencia: HACER APUNTE EN EL HABER

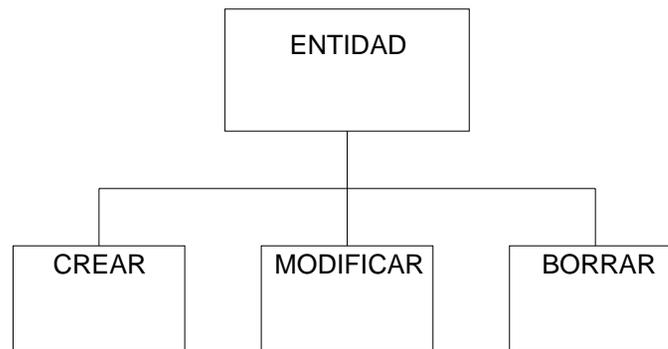
Efectos (II)



- Tipos de efectos:
 - I : insertar
 - M : modificar
 - B : borrar

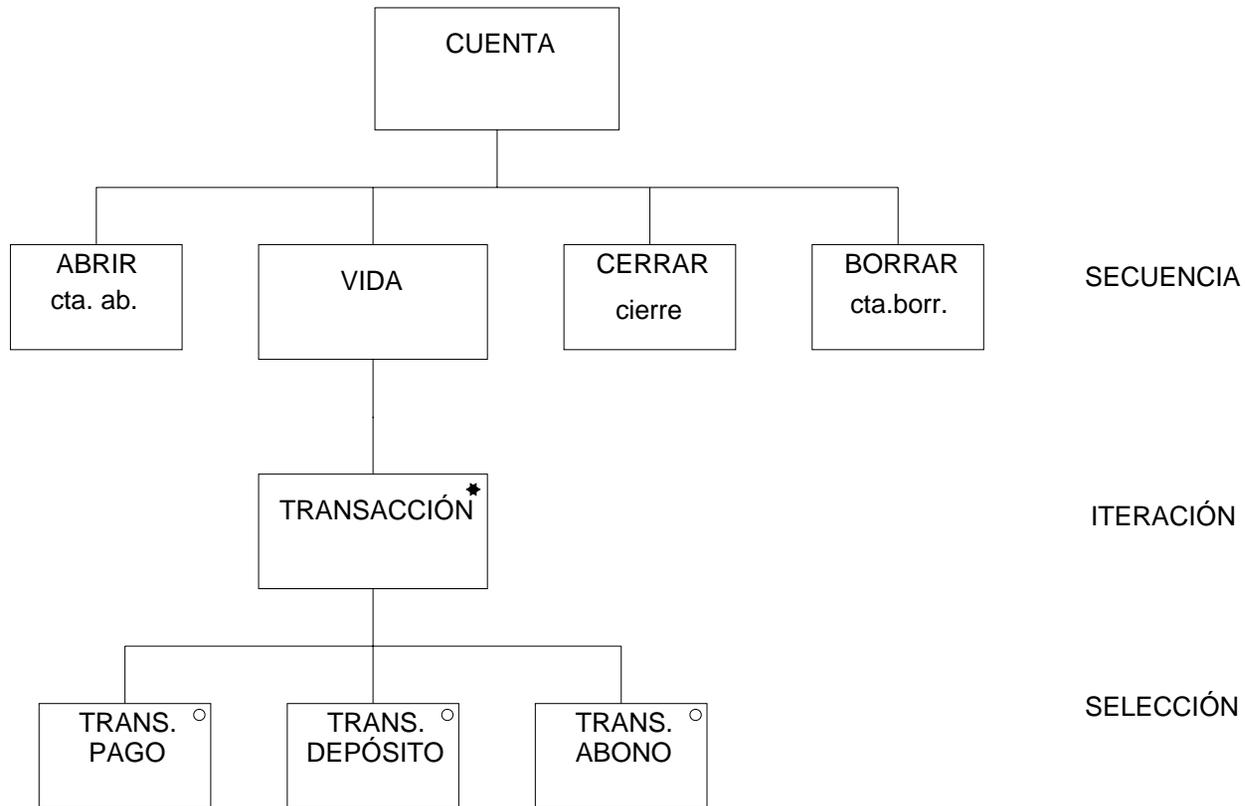
Nodo

- Es una abstracción gráfica que mejora la legibilidad.

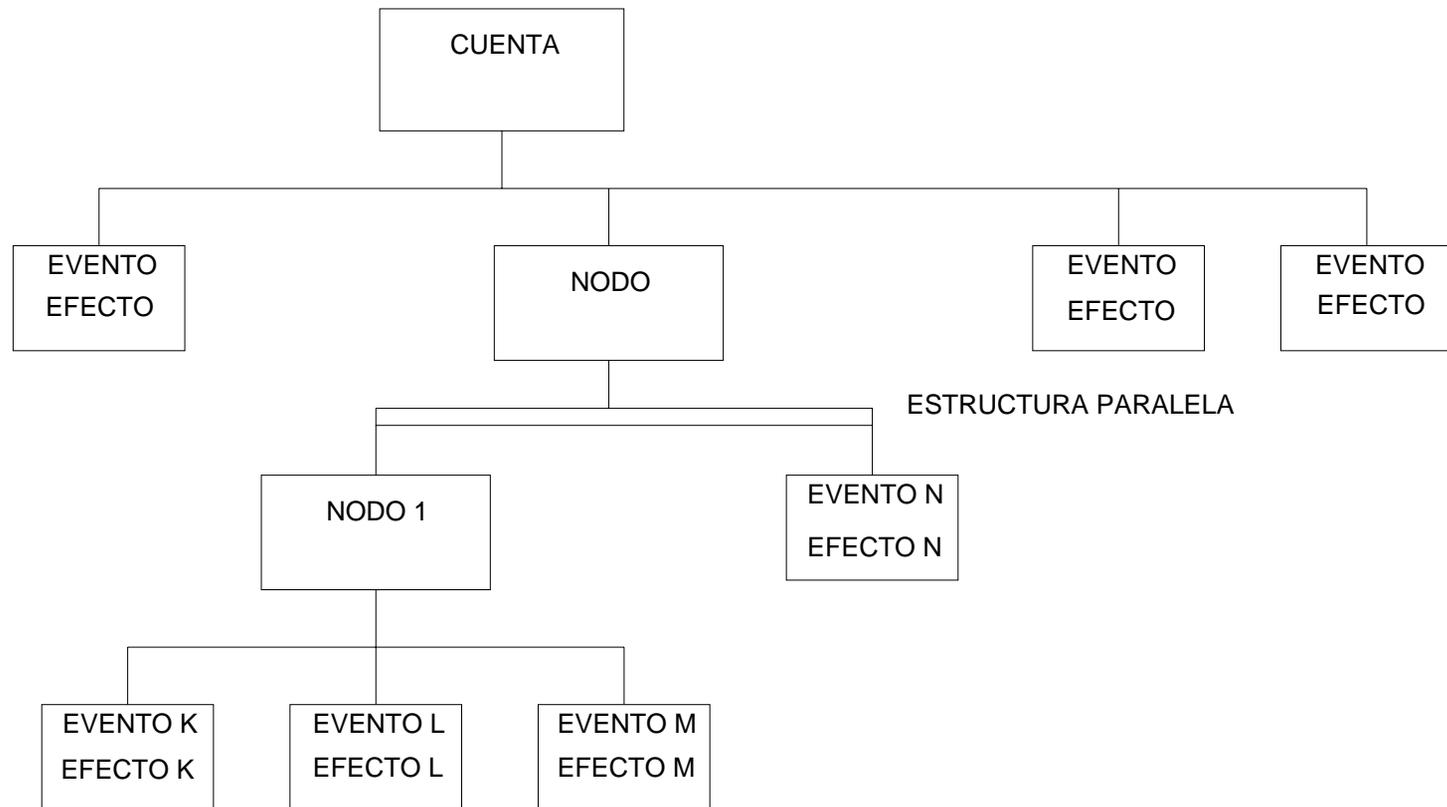


- "Entidad" es un nodo que agrupa todos los eventos que le afectan

HVE. Notación (MAP 95)

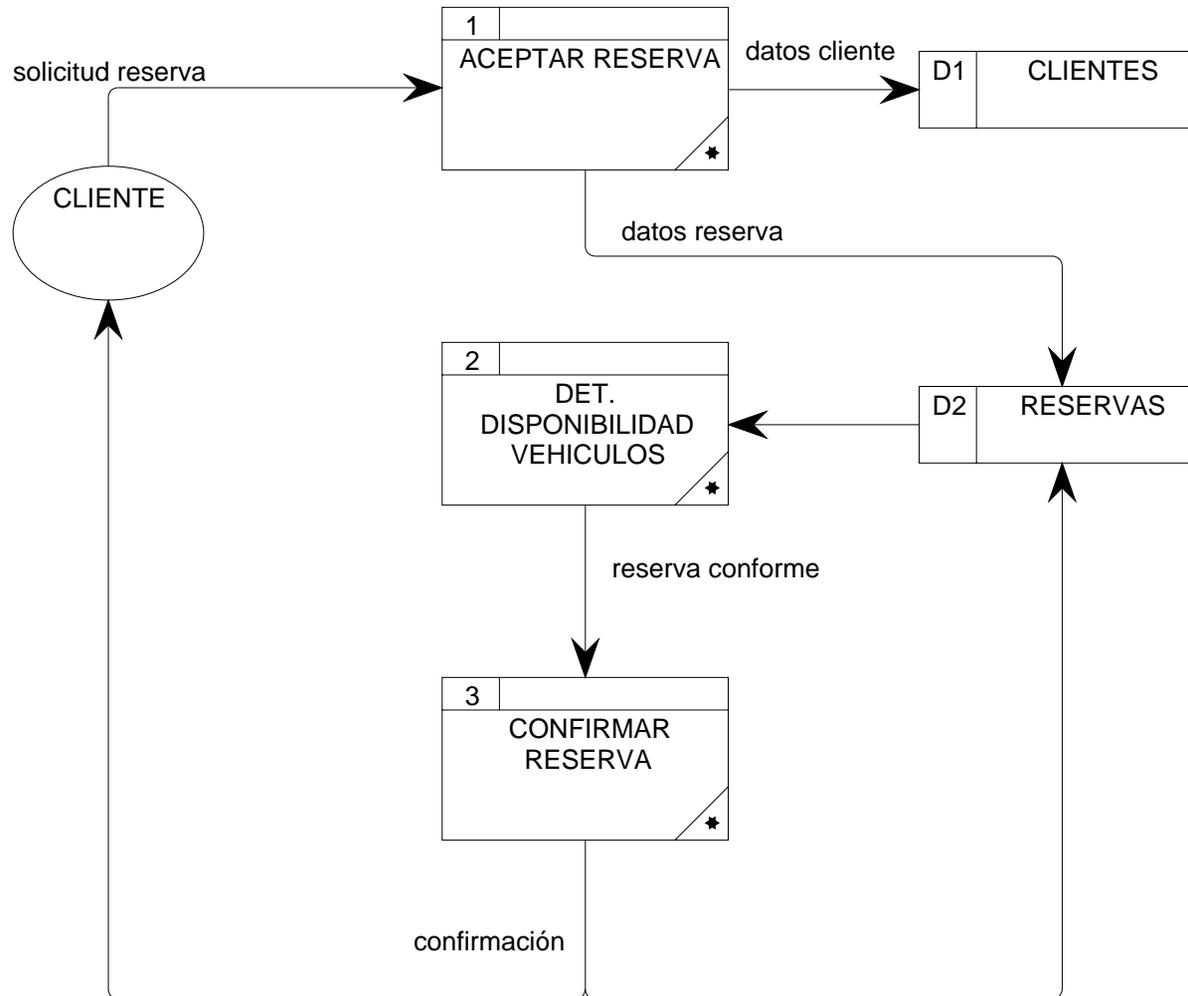


HVE. Notación (II) (MAP 95)



También se pueden poner saltos: R Q (En general, RX QX)

HVE. Ejemplo (MAP 95)



HVE. Construcción



■ 1. IDENTIFICAR EVENTOS

■ En el DFD anterior...

- E1: SOLICITUD DE RESERVA
- E2: SOLICITUD DE RESERVA EFECTUADA POR CLIENTE NUEVO
- E3: CONFIRMACIÓN DE RESERVA
- E4: ASIGNACIÓN DE UN CONDUCTOR A LA RESERVA

⇒ ¿Se te ocurren más eventos?

HVE. Construcción (II) (MAP 95)

■ 2. CONSTRUIR MATRIZ ENTIDAD/ EVENTO

I : Insertar

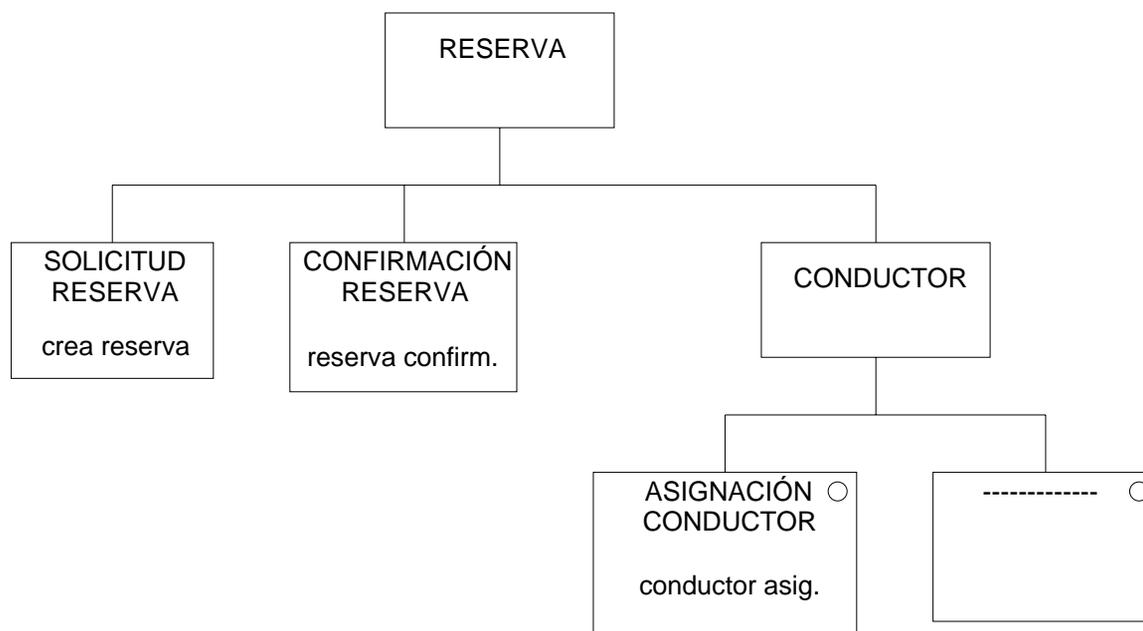
M : Modificar

B : Borrar

	EVENTOS				
	E1	E2	E3	E4	
ENTIDADES	CLIENTE	I	I		
	RESERVA	I		M	M
	CONDUCTOR	M			

HVE. Construcción (III) (MAP 95)

■ 3. CONSTRUIR HVE INICIALES PARA TODAS LAS ENTIDADES

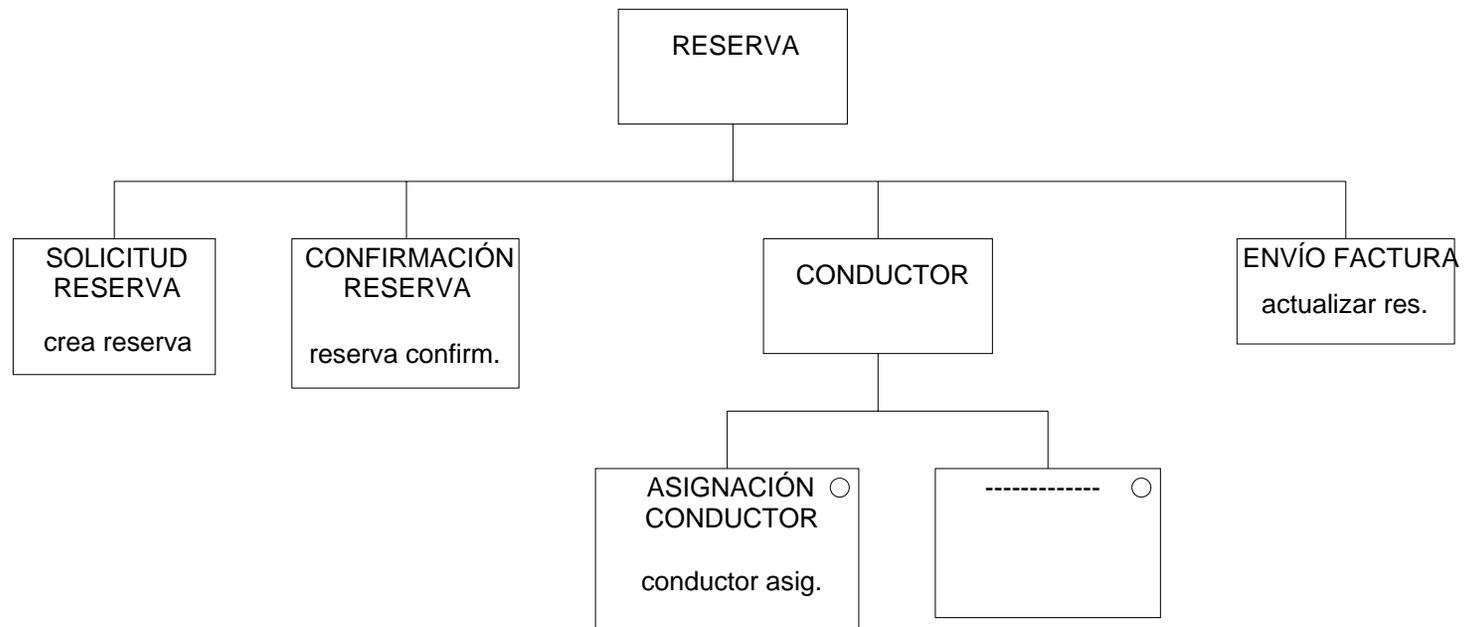


HVE. Construcción (IV) (MAP 95)

■ 4. REFINAMIENTO DE LAS HVE

■ NUEVO EVENTO

– E5: ENVÍO DE FACTURA

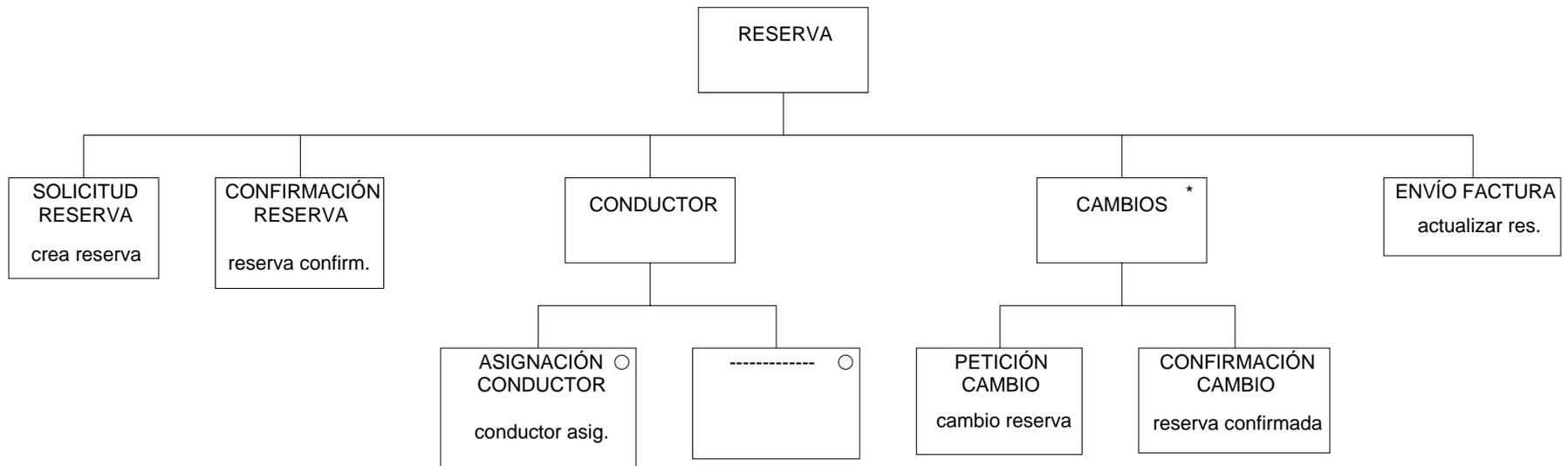


HVE. Construcción (V) (MAP 95)

■ 4. REFINAMIENDO DE LAS HVE

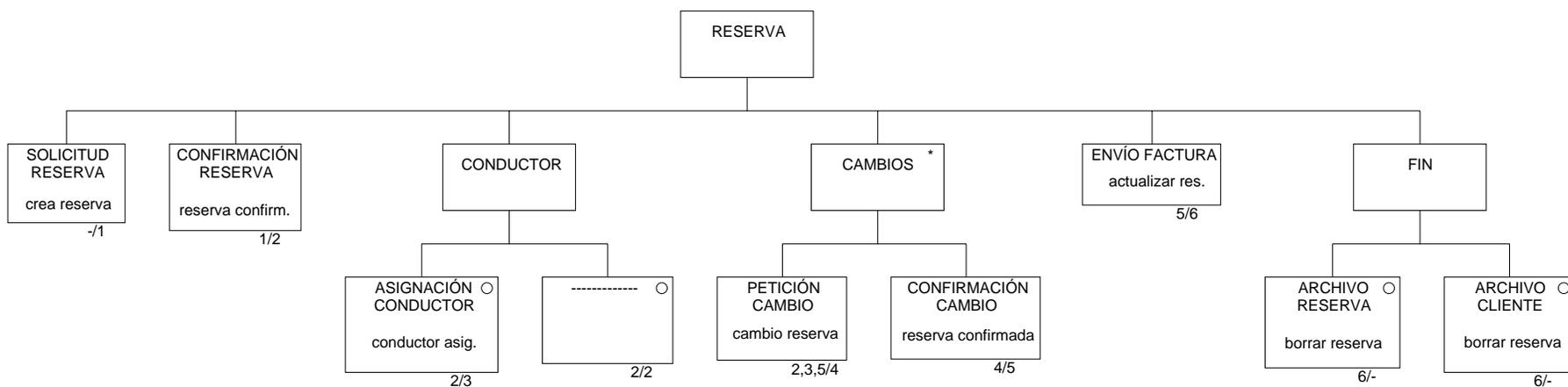
■ NUEVO EVENTO

– E6: PETICIÓN DE CAMBIO



HVE. Construcción (VI) (MAP 95)

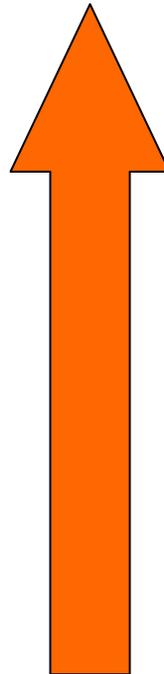
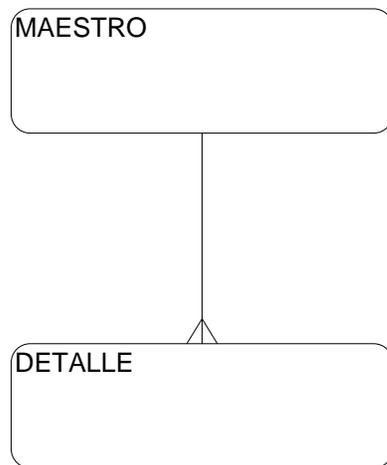
■ 5. AÑADIR INDICADORES DE ESTADO



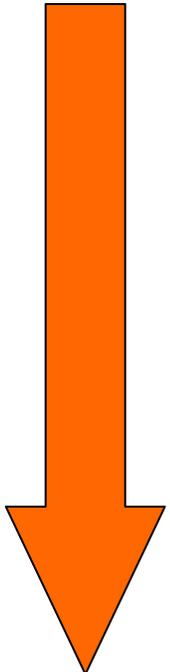
Modo de construcción

- Para cada entidad, en principio, debe haber un HVE

Empezar
por las
entidades
detalle



Y al final...
examinar los
efectos de los
eventos del
maestro en
los detalles



HVE. Relaciones con otras técnicas



- Asegurar coherencia “vista evolutiva” (dinámica o de comportamiento) (HVE) con
 - Vista estática o de datos (E/R o DED)
 - Vista funcional (DFD)
- Para ello, comprobar:
 - DFD: para cada evento en el HVE, existe un proceso en los DFDs del sistema que lo trata
 - E/R o DED: el modelo de datos permite reflejar las repercusiones que la actuación de un evento sobre una entidad tiene sobre otras entidades del sistema