

# Diagramas de flujo de datos

elaborado a partir del libro

DeMarco T.,

"Structured analysis and system specification", Yourdon

Press

(Prentice Hall) (1979)

# SSA

El SSA hace uso de herramientas:

- Diagramas de flujo de datos (**DFD**)
- Diccionario de datos (**DD**)
- Otras: lenguaje natural estructurado, tablas y árboles de decisión.

# SSA

## Contenido del Documento de Especificación:

- Recopilación de los DFDs
- Diccionario de datos
- Recopilación de descripciones de actividades (mini-especificaciones de las primitivas funcionales)

# Diagramas de Flujo de datos

## DFD

los diagramas de flujo de datos son una herramienta de representación del flujo de información

características:

- notación gráfica
- representan el flujo de información
- fácilmente inteligibles
- Permiten descomposición en submodelos

# Diagramas de Flujo de datos

En un DFD aparecen elementos de los tipos siguientes:

- **flujos de datos:** indican flujo de información.
- **procesos** (o actividades): transforman la información.
- **ficheros:** son almacenes de información en los que ésta se guarda a la espera de un procesamiento posterior.
- **entidades externas:** son fuentes o sumideros de información (podría darse el caso de alguna mixta)

# DFDs: notación

Nombre del flujo

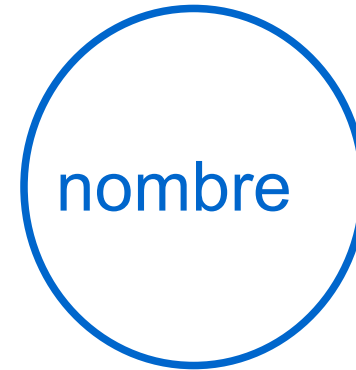
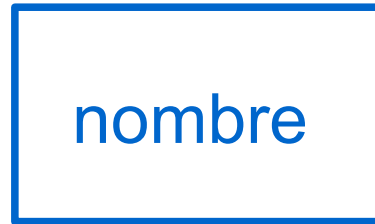


flujos

procesos

ficheros

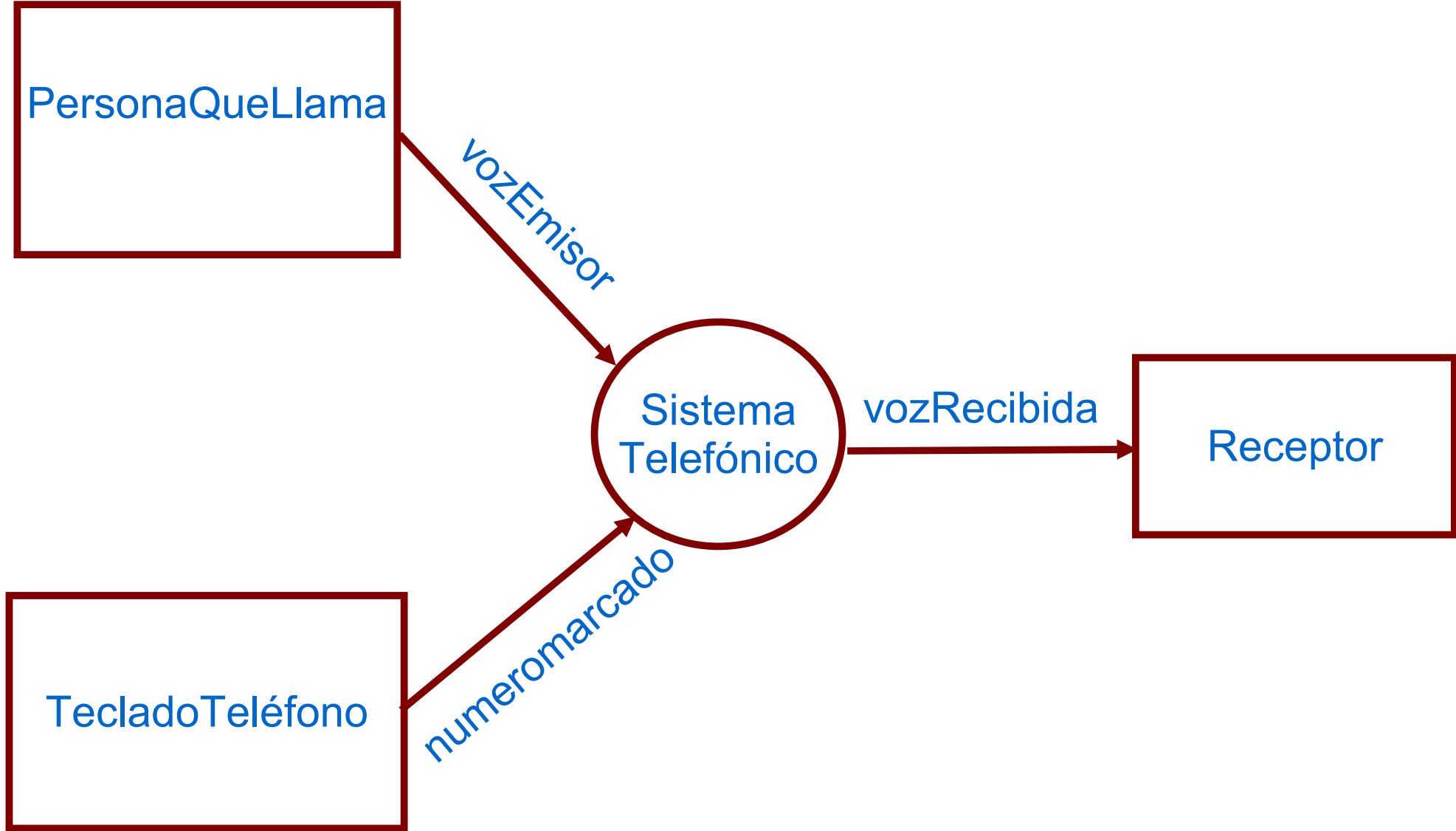
entidades  
externas



# DFDs: reglas

Las reglas que deben respetarse en la elaboración de los DFD son:

- Los nombres deben ser unívocos, en el conjunto de diagramas de un sistema.
- Los flujos pueden converger o divergir.
- Los procesos y ficheros no pueden poseer sólo flujos de entrada, ni tampoco de salida. (Un elemento de estas características debería reflejarse como entidad externa)
- Los flujos no pueden contener información de control.





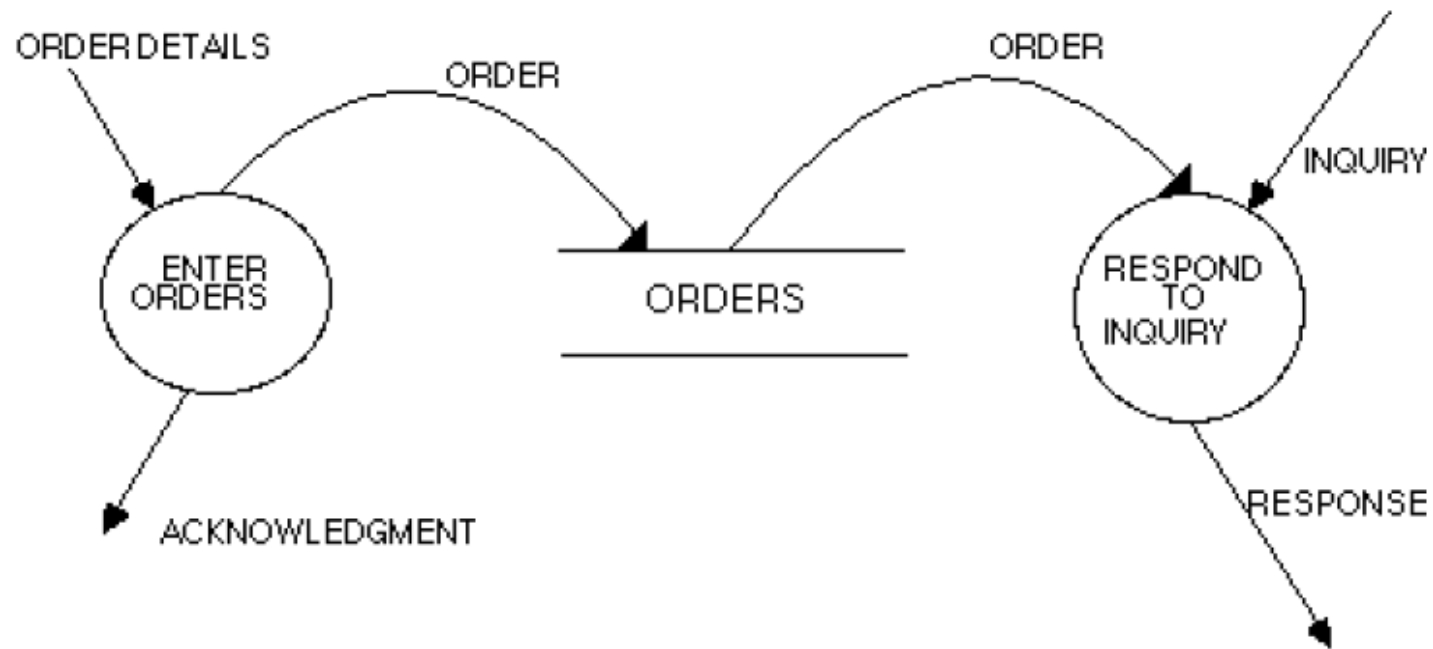
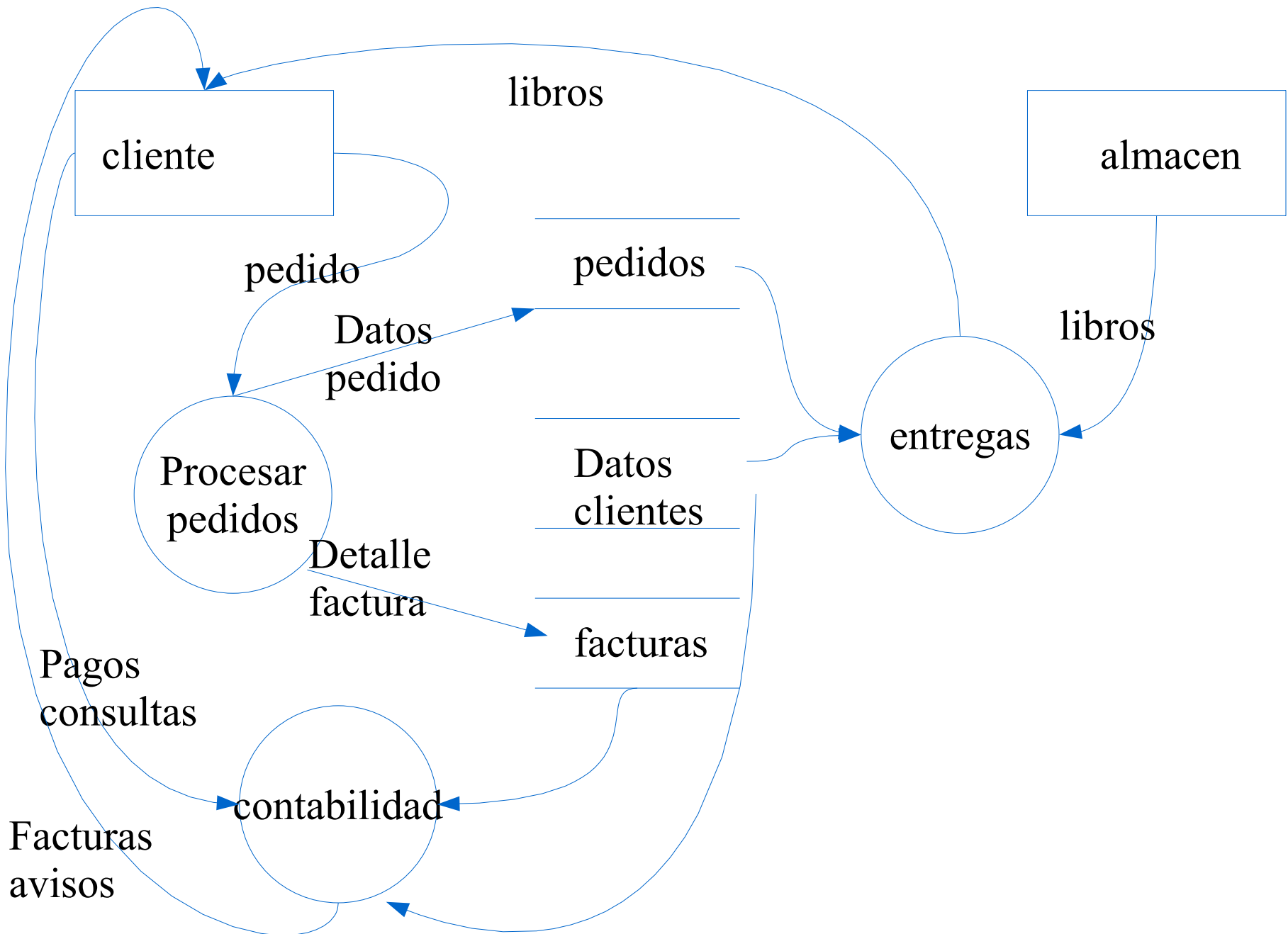
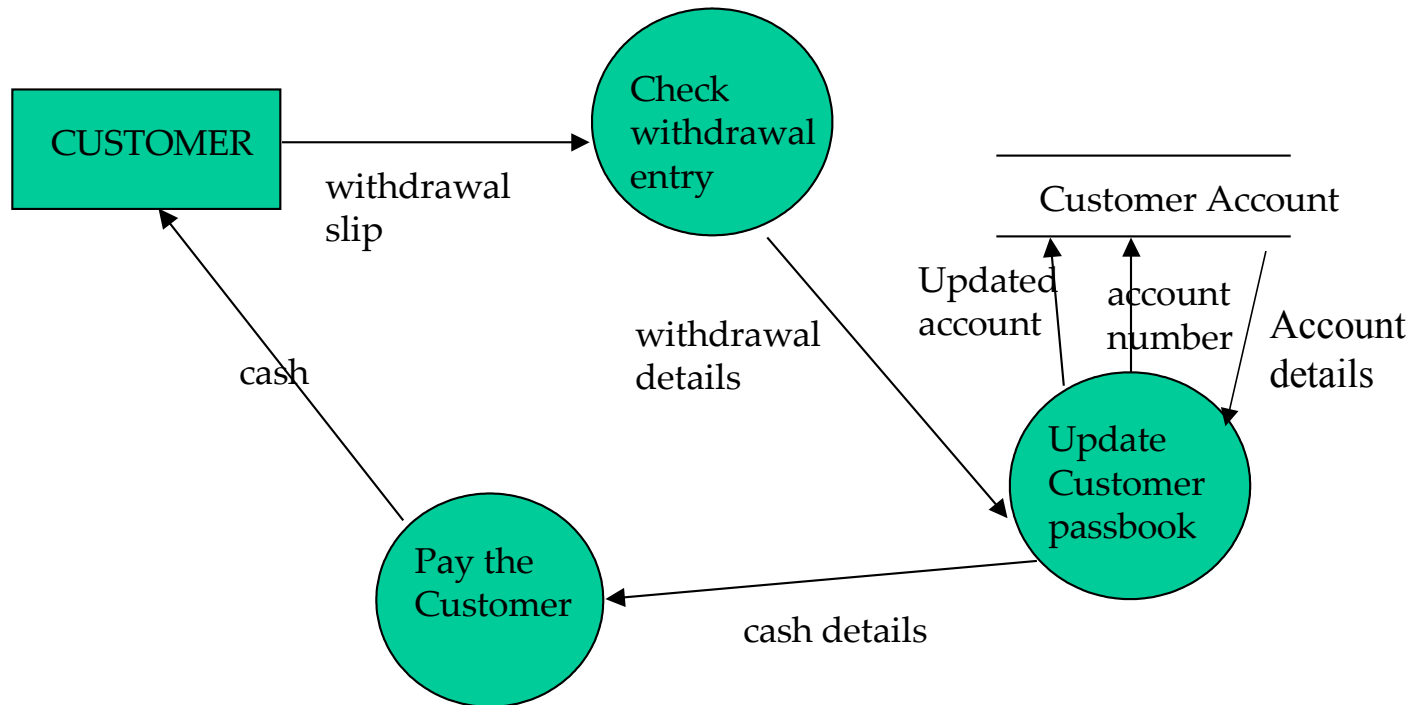


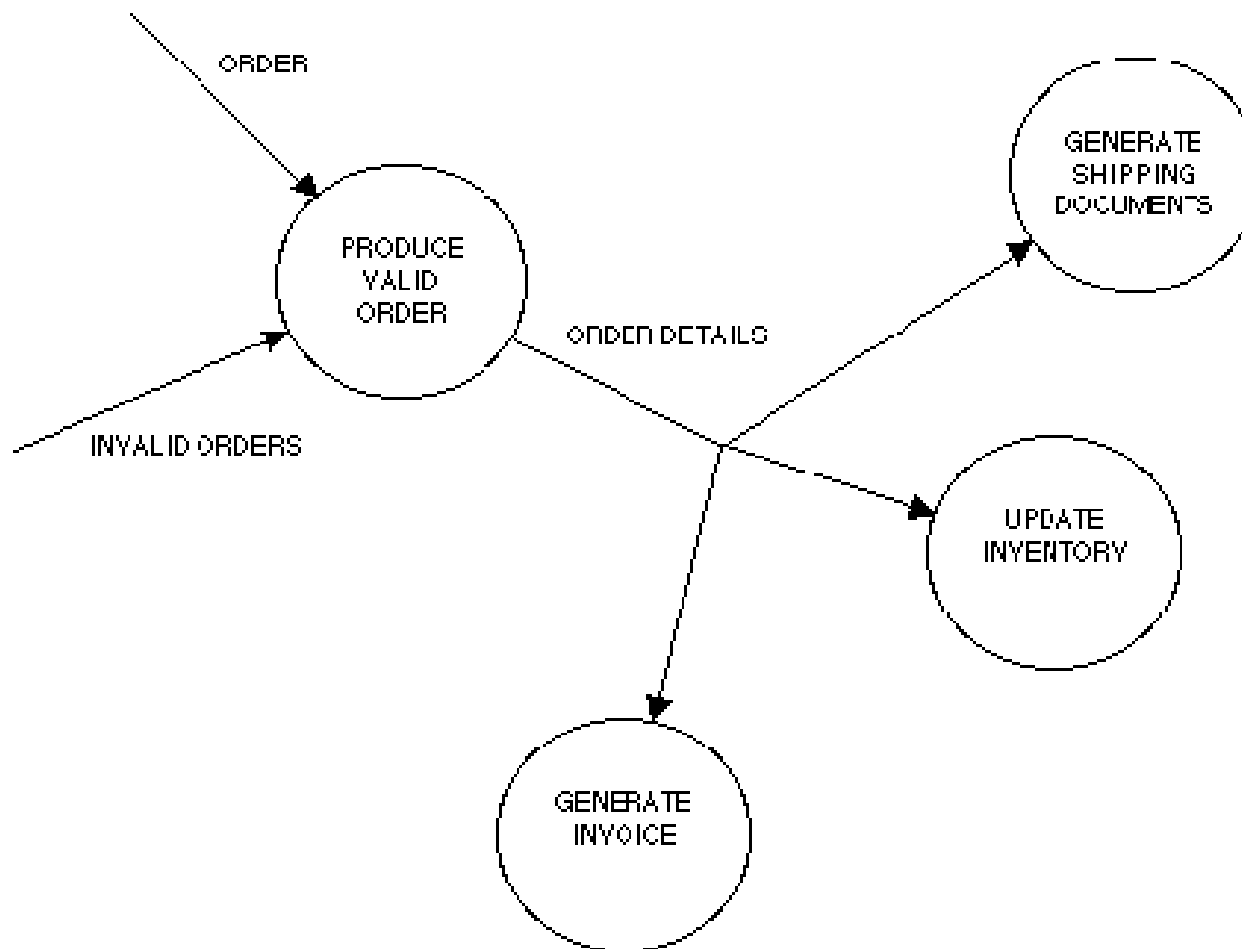
Figure 9.10: A necessary store



## Processes involved in withdrawing cash from savings account using the teller facilities

### Yourdon & DeMarco' Representation





*Figure 9.7(a): A diverging flow*

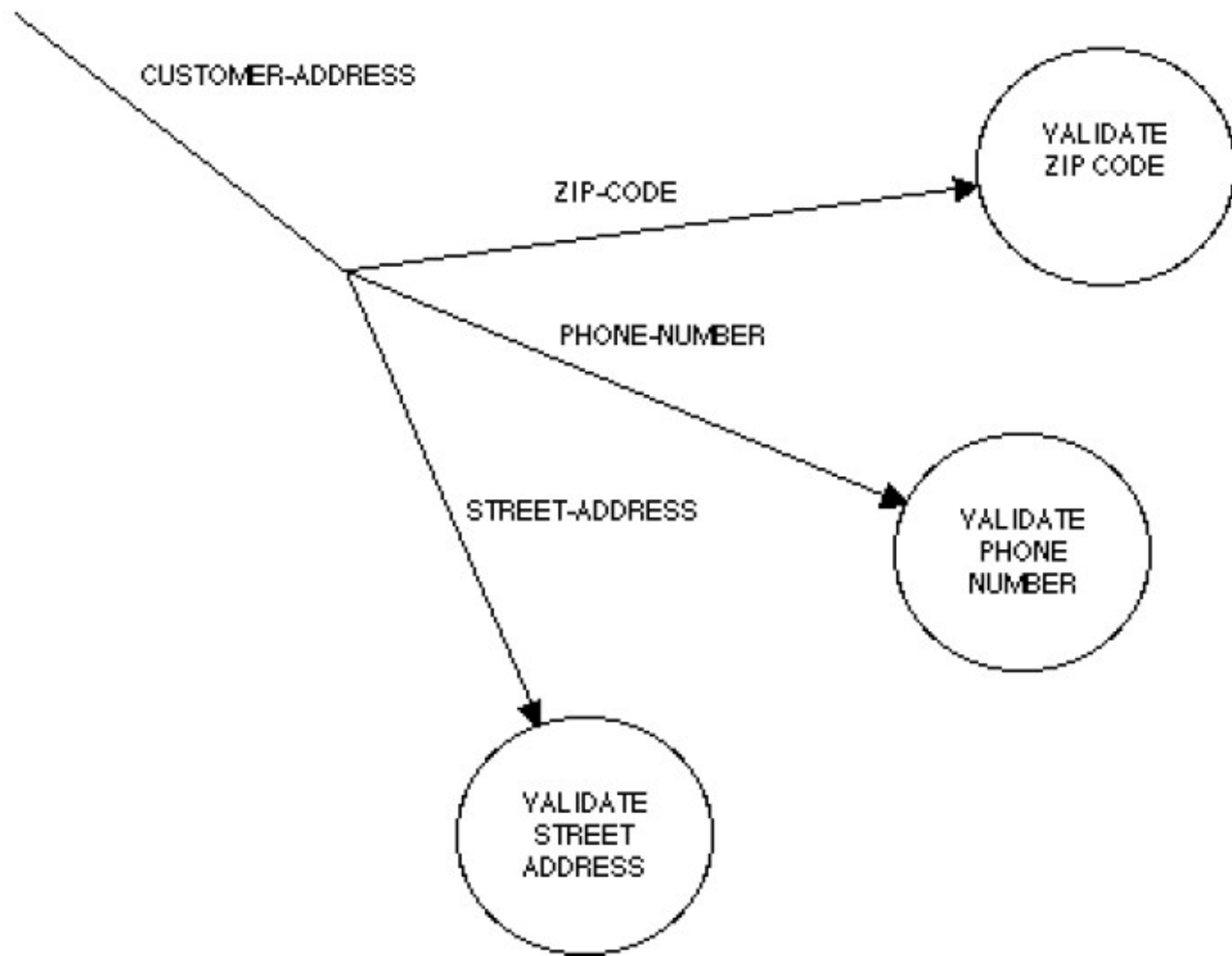


Figure 9.7(b): Another example of a diverging flow

# DFDs: consejos

Cómo elaborar un DFD:

- La mentalidad con que debemos abordar la realización de un **DFD** es la de "**interrogar a los datos**".
- Debemos preocuparnos sistemáticamente antes de los flujos de información que de los procesos.
- Es conveniente dar nombre primero a los flujos y después a los procesos.
- **EI** proceso a seguir podría ser ...

# DFDs: elaboración

El proceso a seguir podría ser el siguiente:

- Identificar todos los flujos netos, de entrada y de salida, del diagrama y dibújelos en su periferia.
- Intentar conectar las entradas con las salidas, y las salidas con las entradas, dejándose guiar por la información procedente del usuario, intentando reflejar el flujo de información tal como es más que preguntarse porqué es así.
- Revisar el diagrama siguiendo el camino desde las entradas a las salidas, desde las salidas a las entradas, y desde el centro a la periferia.
- Prepararse para volver a empezar.

# DFDs: elaboración

**Al intentar conectar las entradas con las salidas, iremos:**

- reflejando los flujos de información que aparezcan en el sistema,
- situando procesos (sin darles nombre en un principio) donde sea necesario procesar información para conectar unos flujos con otros,
- introduciendo los ficheros (almacenes temporales de información) que existan en el sistema.



# DFDs: elaboración

Al revisar el diagrama, iremos:

- Examinando los flujos, su contenido, y preguntándonos:
  - ¿Qué necesito para elaborar este contenido?
  - ¿De dónde proceden sus componentes?
  - ¿Puedo elaborar (mediante un proceso) la información de algún otro flujo para obtener la de este?
- Dando nombre a las cajas blancas, que comienzan a proliferar,
- preguntándonos si es necesaria la información de algún otro flujo para obtener las salidas del proceso a partir de las entradas.

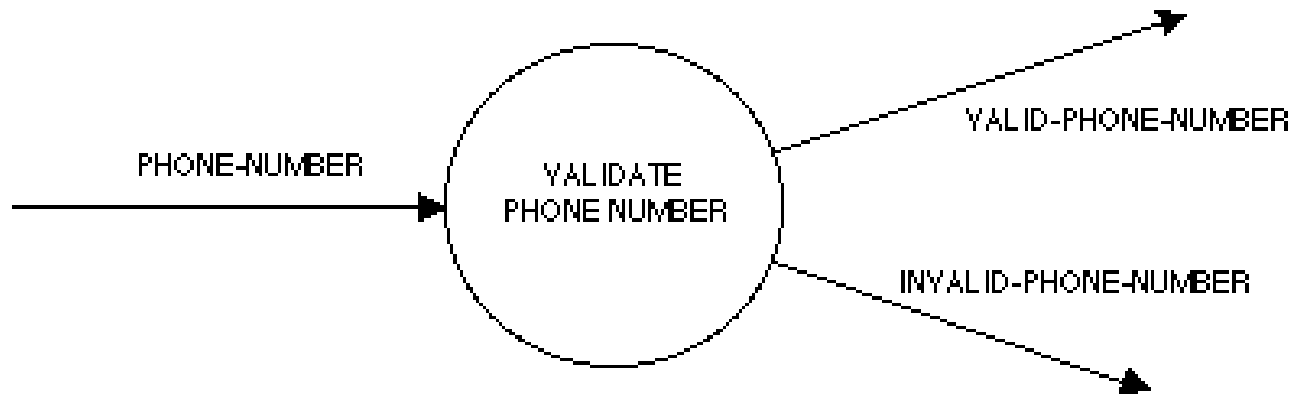
# DFDs: elaboración

**Prepararse para volver a empezar**, no nos extrañe encontrar

- algún flujo de entrada desconectado del resto del diagrama (eliminarlo sin más),
- alguna zona dentro del diagrama desconectada del resto del mismo (si no es una zona de interés la eliminaríamos)

# DFDs consejos: denominaciones

- Dar a los flujos nombres significativos, que reflejen su contenido en información y, si es relevante, el estado de la misma.
- Habrá que poner especial esmero en la denominación de los flujos de la interface del sistema.
- Un proceso bien concebido puede describirse fácilmente en términos de sus entradas y salidas



*Figure 9.5: A typical DFD*

# DFDs: + consejos

Sobre los flujos de los ficheros:

- A veces, asociadas a las entradas/salidas de los ficheros, existen salidas/entradas triviales, que no es necesario reflejar. (Por ejemplo, una modificación de un item de un fichero requiere una lectura previa, que no siempre reflejaremos).
- A veces, en ficheros simples, sólo existe un flujo de entrada, y otro de salida, con el mismo contenido que los item del fichero. En este caso bastará con dar nombre al fichero y dejar sin nombrar los dos flujos, a los que se pondrá el mismo nombre.

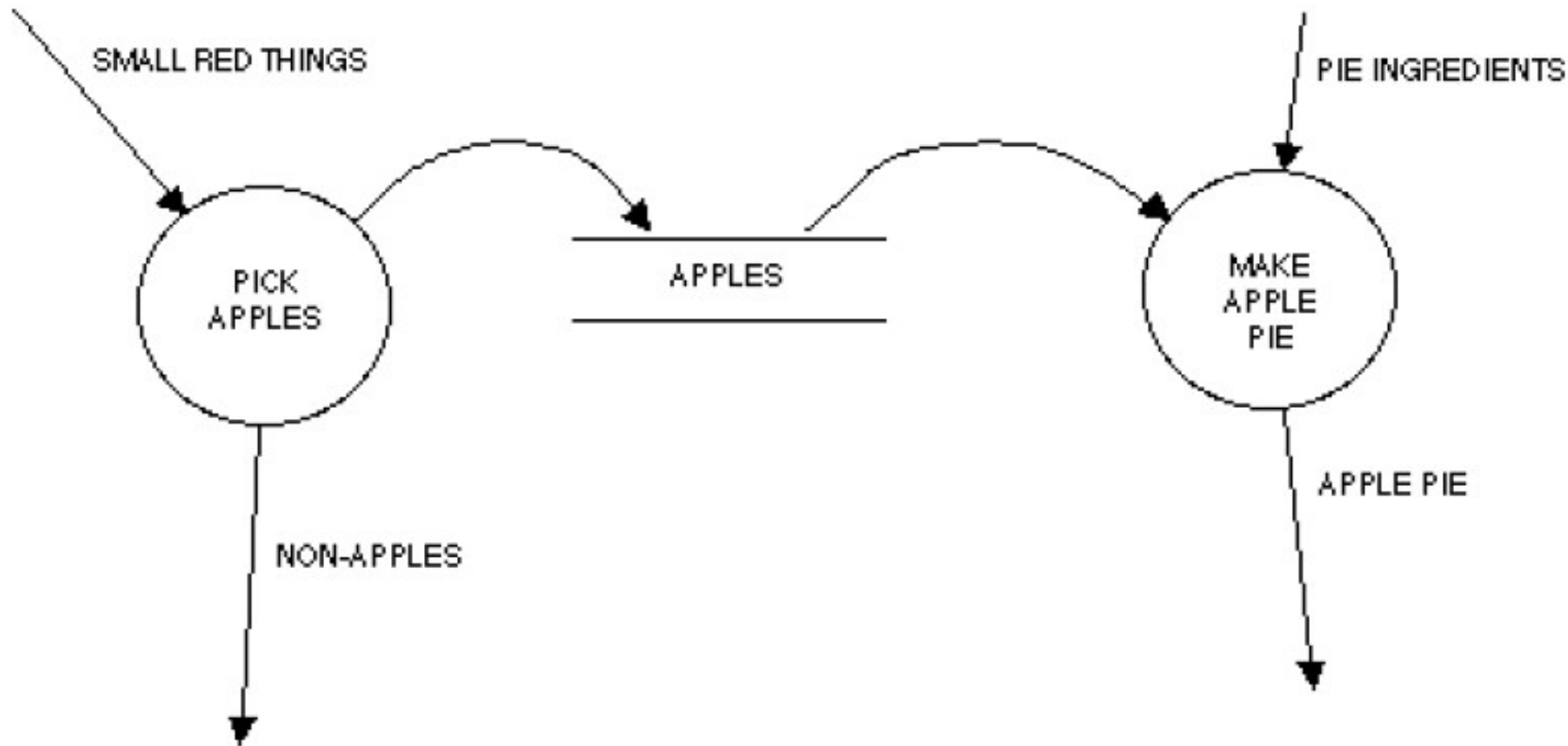


Figure 9.12: Stores with unlabeled flows

# DFDs: + consejos

Otras recomendaciones:

- Basta con que los **DFD** reflejen el comportamiento de los sistemas cuando están en un régimen estacionario
- No es necesario reflejar exhaustivamente los caminos de flujo asociados al tratamiento de errores.
- **¡Ojo con los flujos de control!**

# DFDs jerarquizados

## DFD jerarquizados

No es eficiente trabajar con diagramas muy extensos, en los que aparezcan muchos procesos.

La solución: organizar los diagramas por niveles, que vayan ampliando progresivamente el detalle de la descripción.



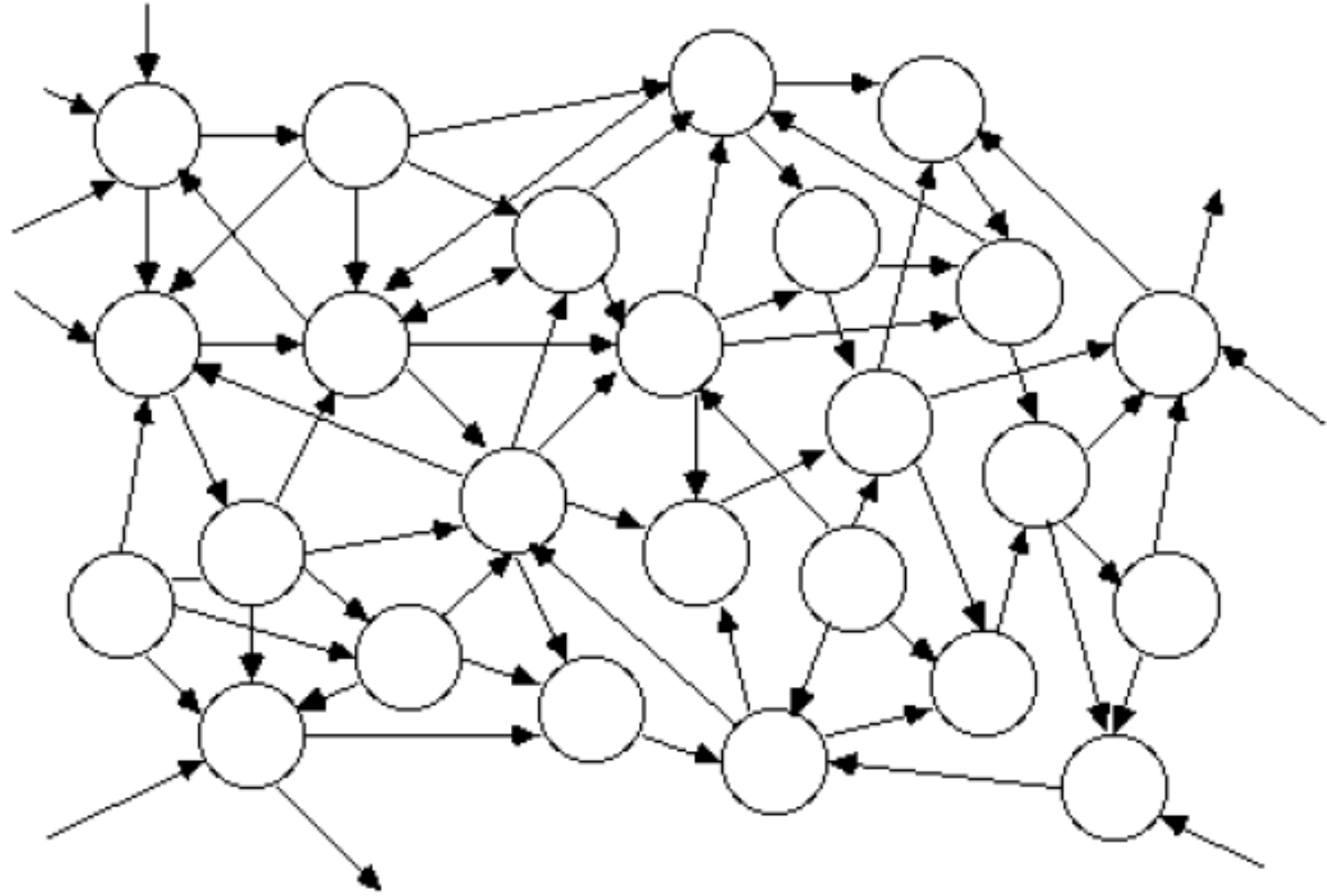


Figure 9.20: A complex DFD

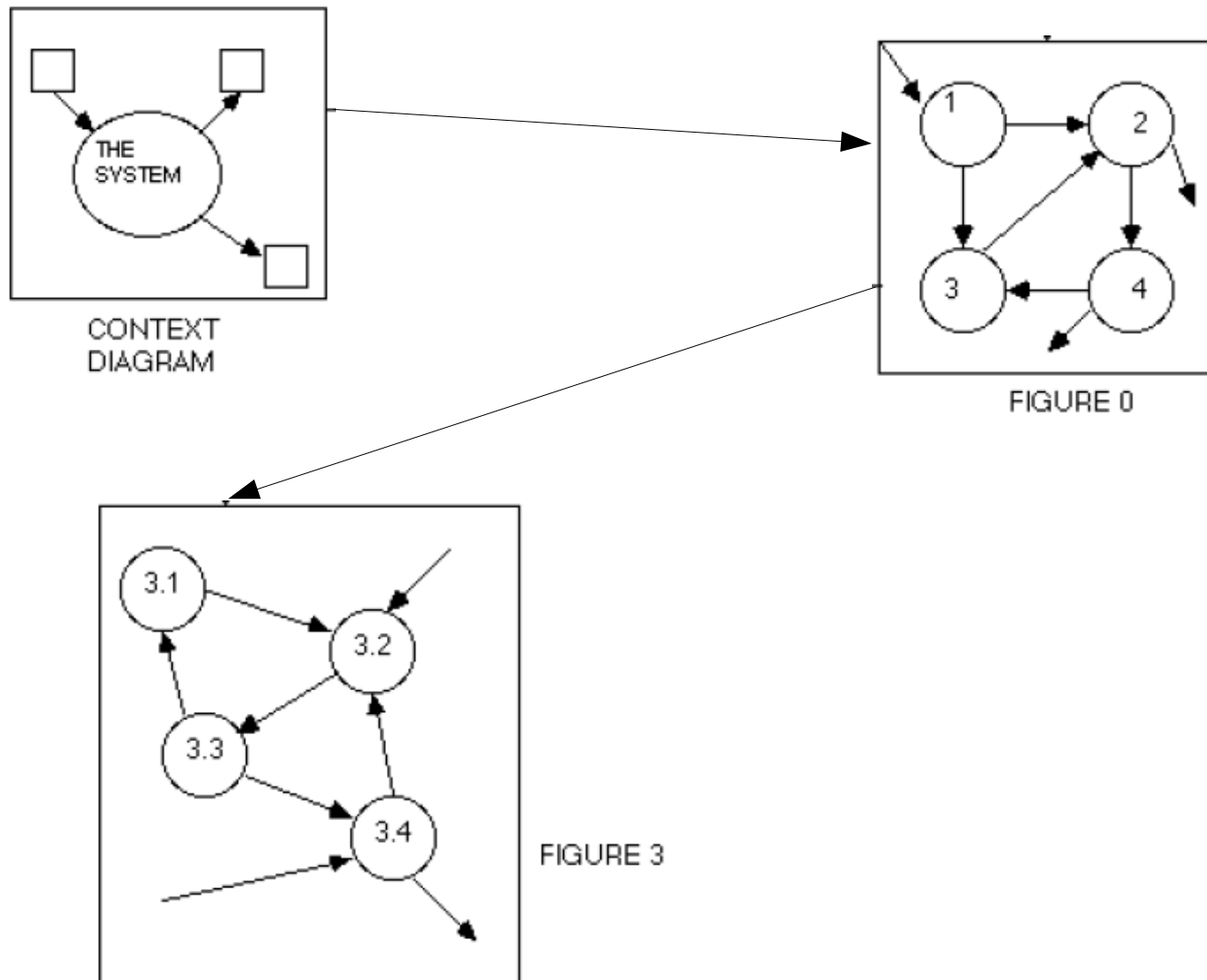


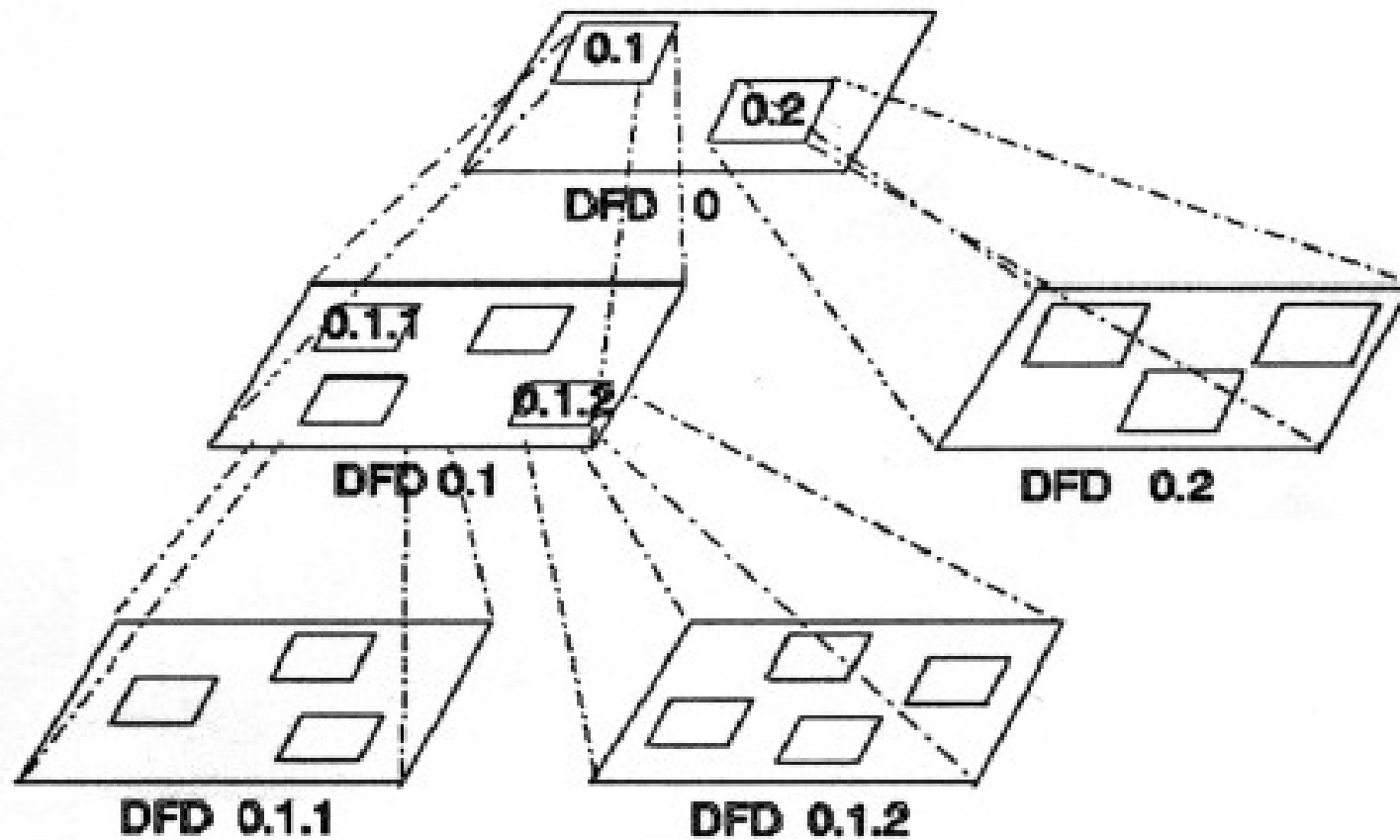
Figure 9.21: Leveled dataflow diagrams

# DFDs jerarquizados

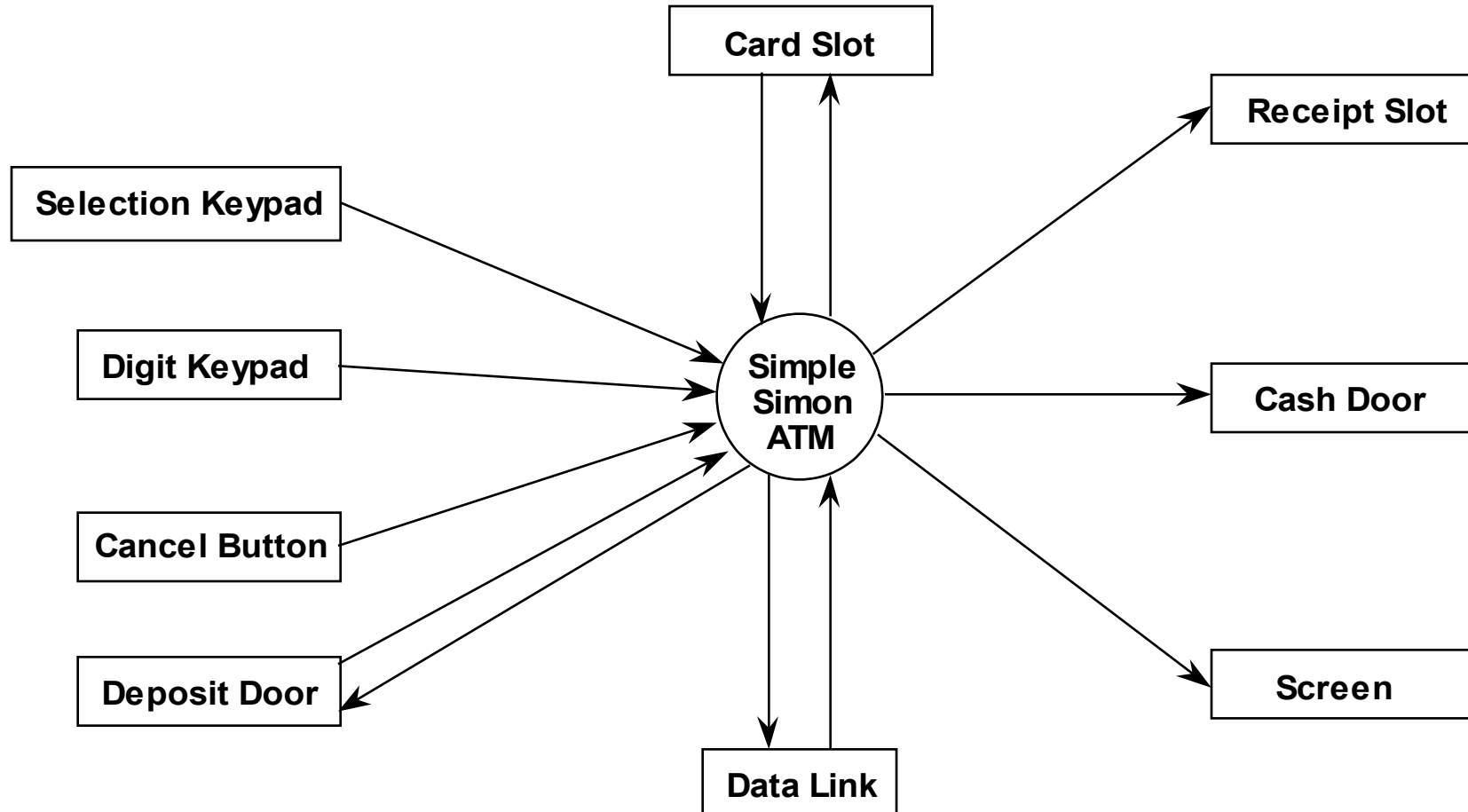
En una jerarquía de DFDs se distinguen:

- de **contexto**: es el diagrama situado en la raíz del árbol, muestra su interacción con el entorno.
- **intermedio**: como su nombre indica son nodos del árbol no terminales.
- **primitivas funcionales**: son los procesos a los que no corresponde ningún **DFD** en un nivel inferior.

## DENOMINACION DE PROCESOS Y DIAGRAMAS NOTACION



# Context Diagram of the Simple ATM System



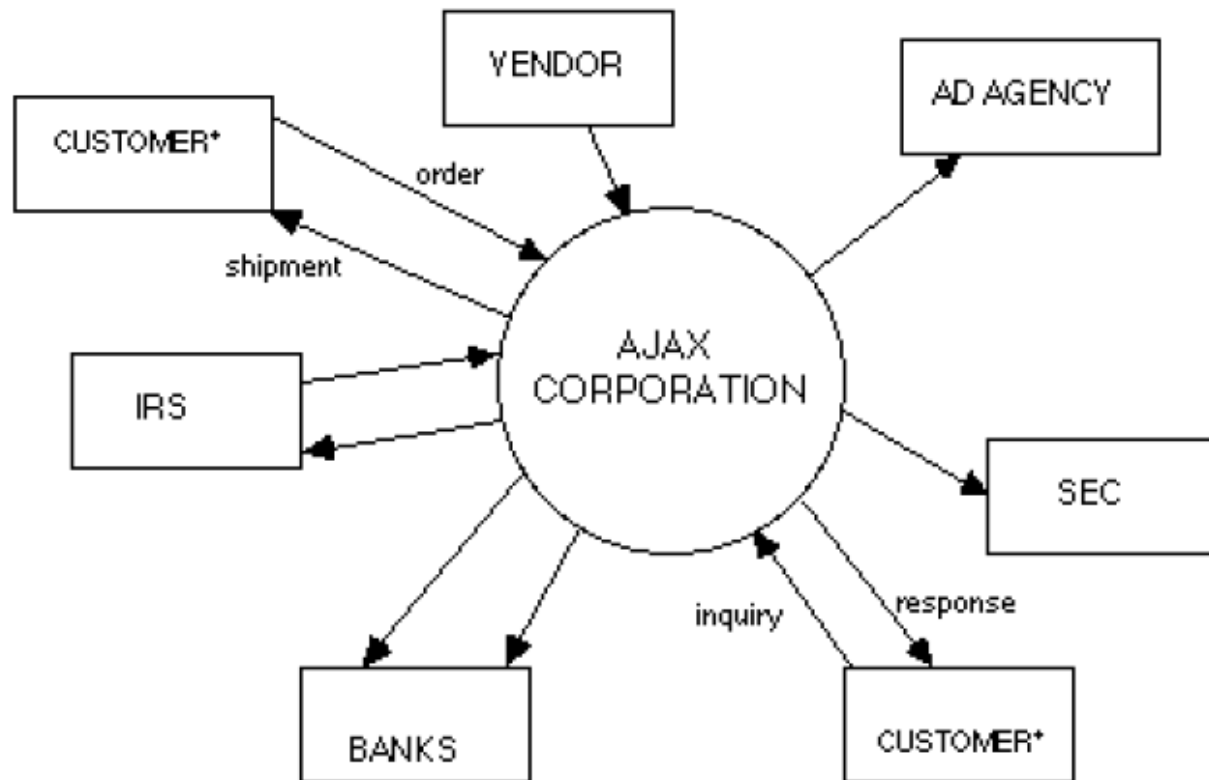
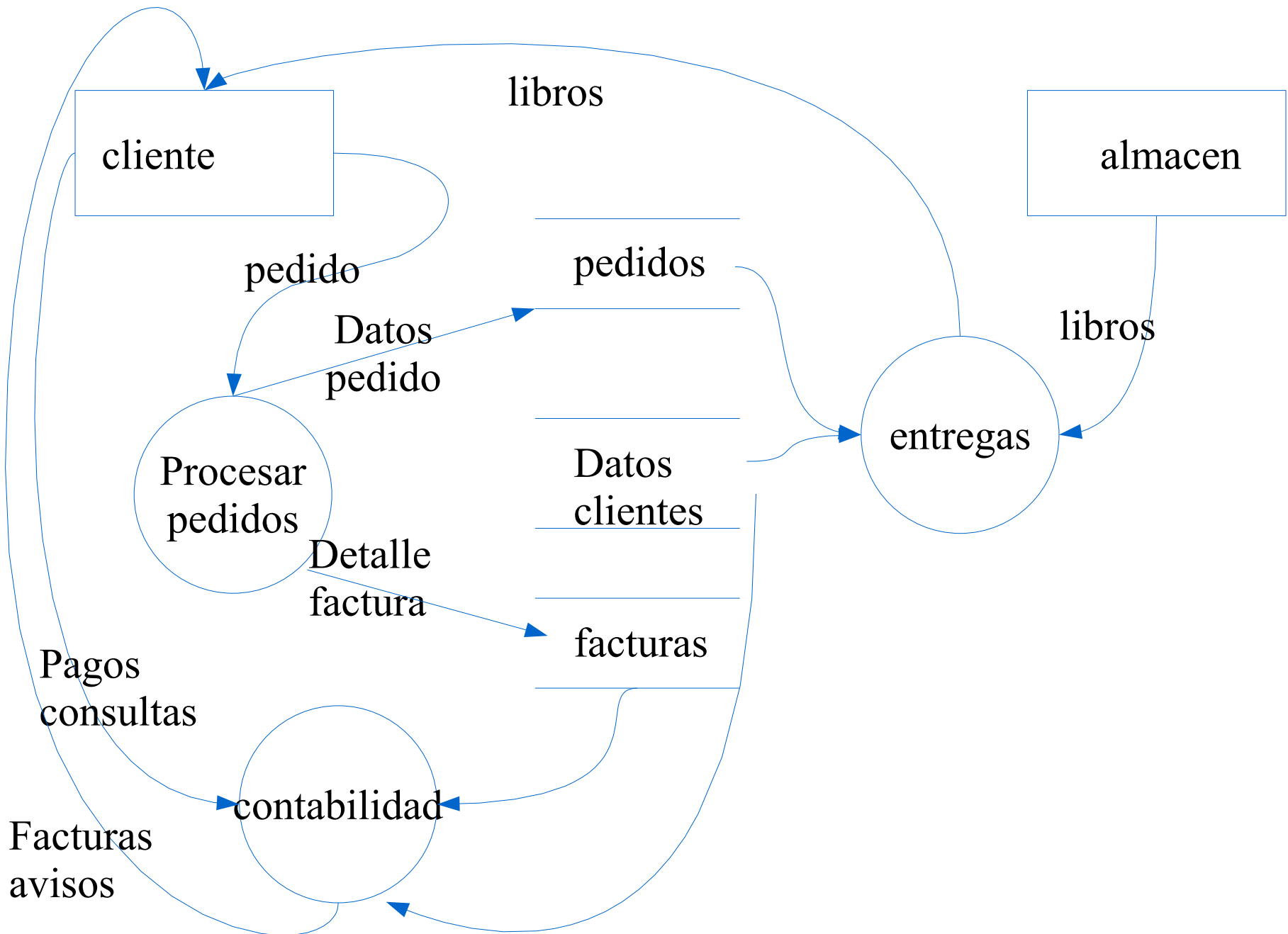
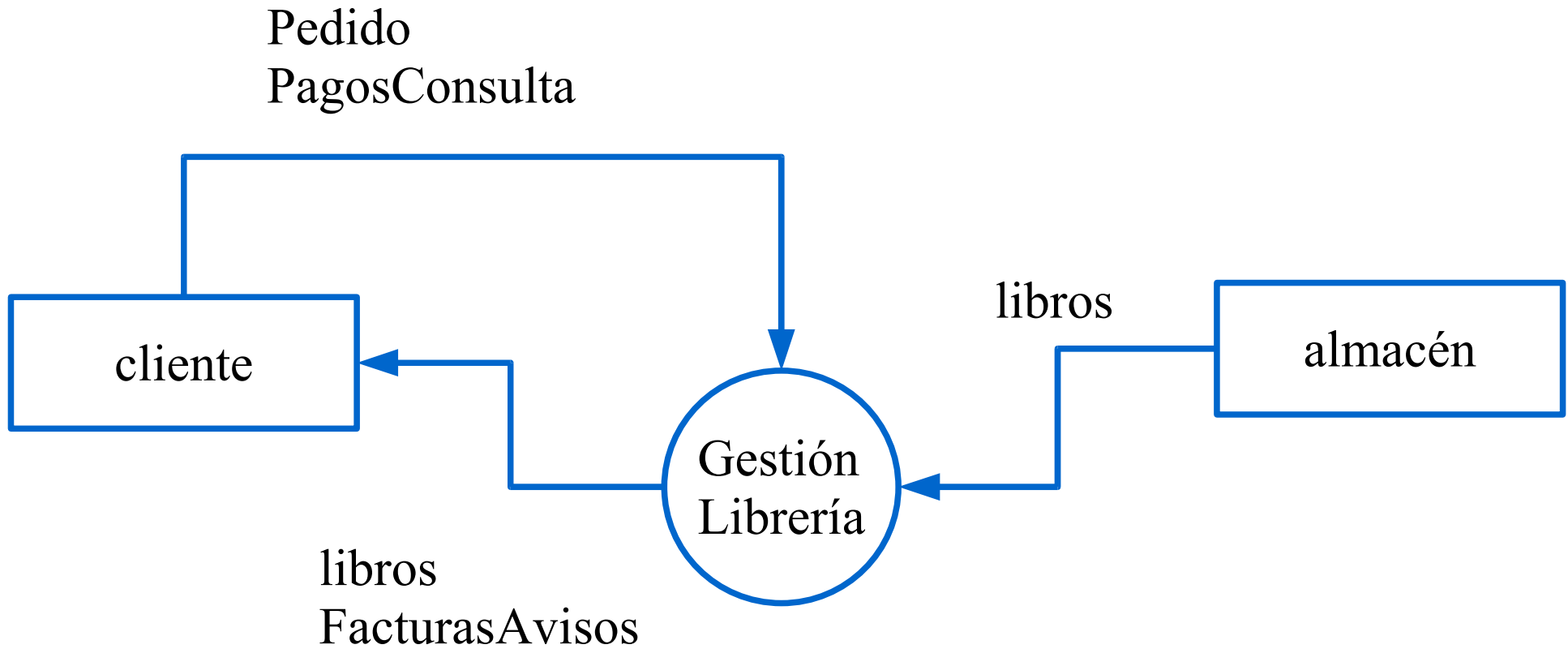


Figure 9.15: A typical context diagram

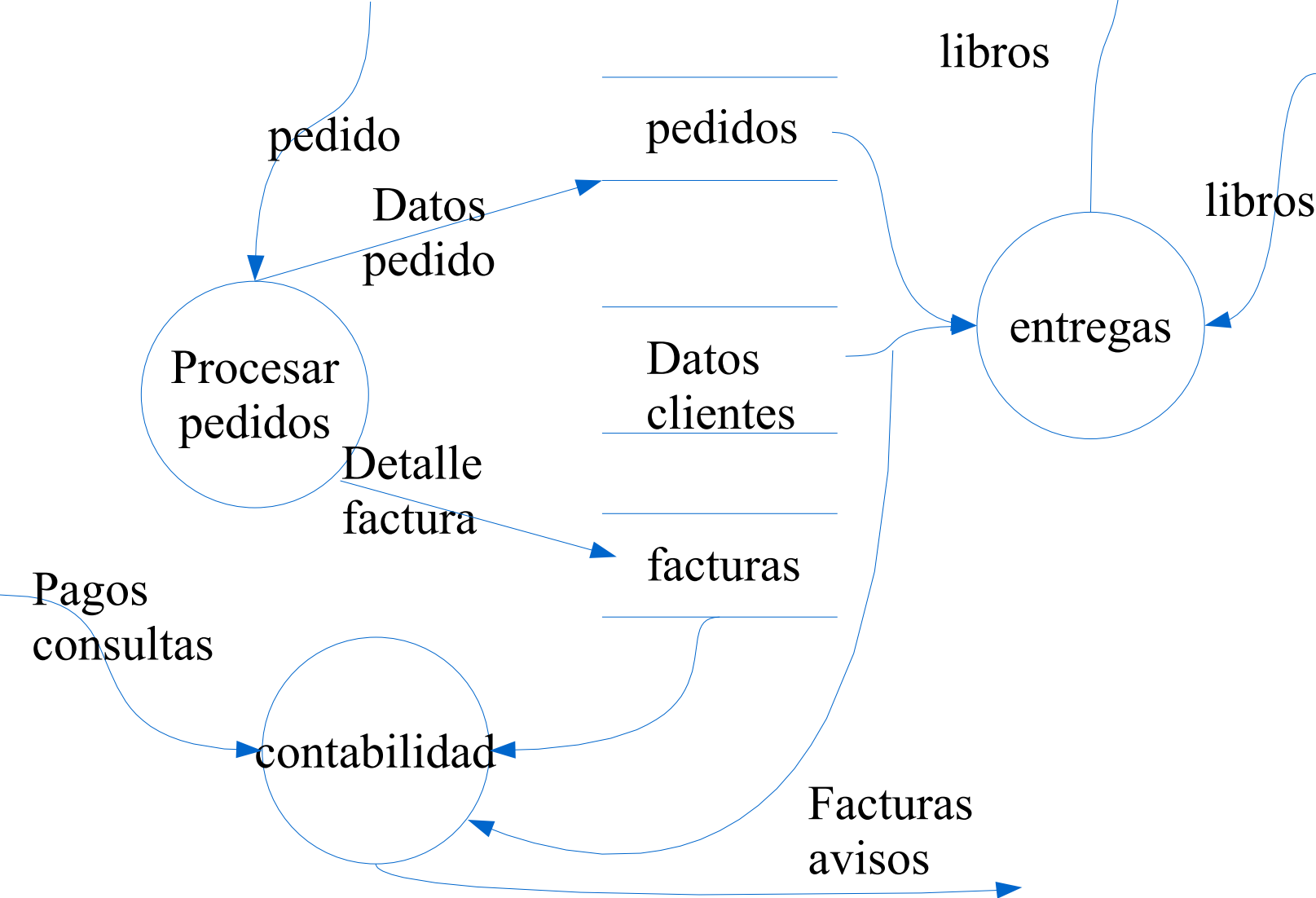


# Diagrama de contexto





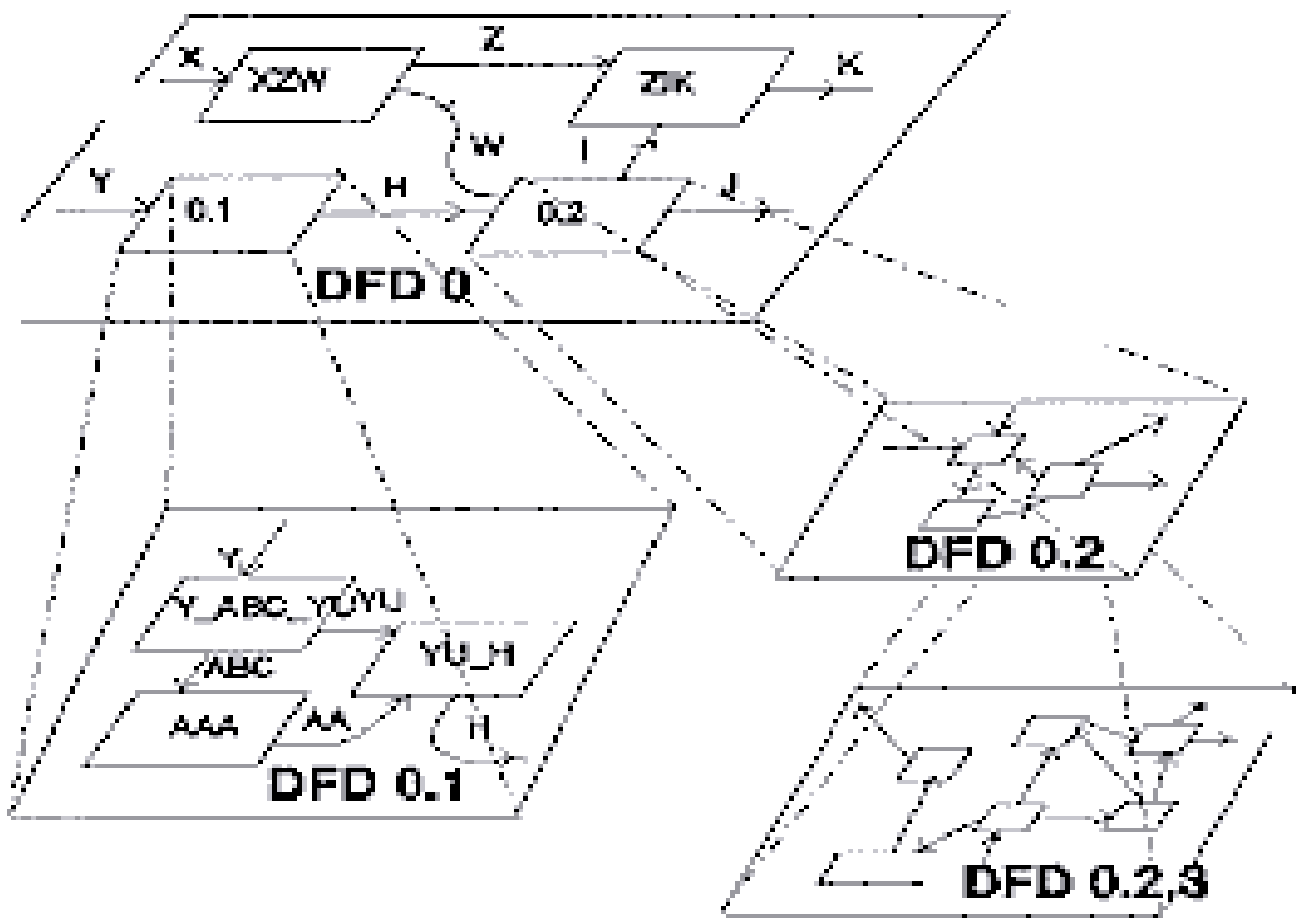
# Proceso Gestión Librería



# DFDs jerarquizados: reglas

## reglas:

- balanceo de los diagramas: las entradas y salidas netas de un diagrama tienen que coincidir con los flujos de entrada y salida del proceso a que corresponde, en el nivel superior.
- ubicación de los ficheros: un fichero sólo puede aparecer en un diagrama, y ¿en qué nivel?: en el primero en que interaccione con más de un proceso.
- No debe indicarse, sobre los diagramas, la procedencia o destino de los flujos exteriores; esto puede averiguarse mirando en el correspondiente diagrama del nivel superior.



# DFDs jerarquizados: consejos

- En los diagramas, el número de procesos no debe ser ni muy bajo, lo que multiplicaría el número de diagramas, ni muy alto (un número mayor de siete comienza a ser problemático).
- Las interfaces, como consecuencia de una división en sub-diagramas apropiada, no deben ser muy complejas, y deben guardar una cierta relación con el problema.
- Los diagramas deben ser legibles,
- ¿A partir de qué nivel debe dejar de descomponerse un diagrama?: cuando los procesos que contengan se puedan describir con facilidad; hay quien entiende que esto requiere que sólo presenten un flujo de entrada y otro de salida.

# DFDs: verificación

## Denominaciones:

- que todos los flujos tengan un nombre asignado,
- que todos los flujos tengan un contenido conocido,
- que los nombres de los procesos hagan referencia a sus entradas y salidas, y guarden relación con el contenido de los sub-diagramas que dependan de ellos,

# DFDs: verificación

## Consistencia:

- el balanceo de los flujos,
- que un mismo proceso no aparezca más de una vez,
- Que cada proceso pueda elaborar los flujos de salida contando sólo con la información de los flujos de entrada (conservación de los datos).

# DFDs: verificación

## ficheros

- que no aparezcan ficheros que sólo reciben información (sumideros).

## Legibilidad

- que las interfaces sean sencillas,
- que los nombres de los procesos sean significativos (evitar nombres como PROCESAR\_SALIDA),
- que la descomposición funcional sea uniforme (esto es, que la jerarquía de diagramas esté equilibrada).

# ejemplos de nombres de procesos

nombre	¿Qué tal?
calcular trayectoria misil	
hacer resumen de inventario	
procesar entrada	
validar numero telefono	
funciones varias	
atender clientes	
hacer algo	
asignar grupo	
funciones específicas	



# ejemplos de nombres de procesos

nombre	¿Qué tal?
calcular trayectoria misil	bien
hacer resumen de inventario	bien
procesar entrada	
validar numero telefono	bien
funciones varias	
atender clientes	
hacer algo	
asignar grupo	bien
funciones específicas	