



Departamento de Informática  
Universidad de Valladolid  
Campus de Segovia

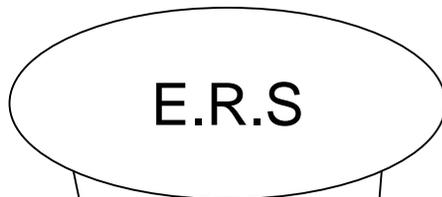
---

# TEMA 8: DISEÑO ESTRUCTURADO

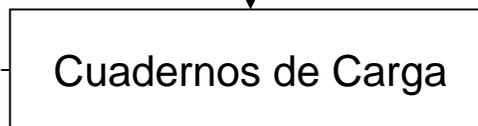
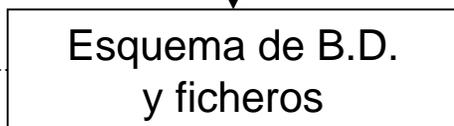
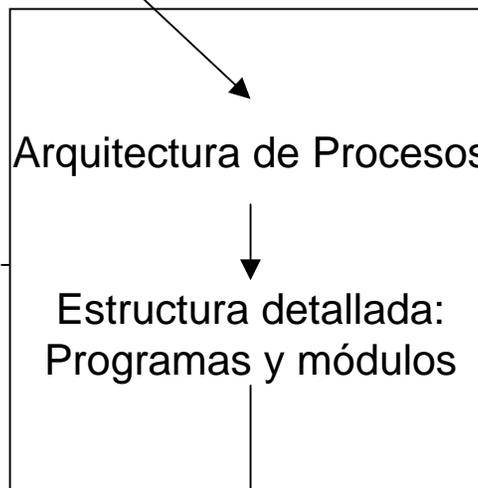
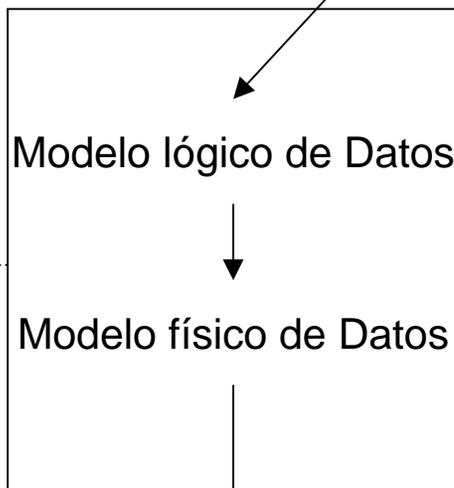
# DISEÑO ESTRUCTURADO

- Conceptos generales
  - Ideas básicas y notación
- Estrategias de diseño
  - Análisis de transacciones y de transformaciones

Análisis



Diseño



Implementación



# DISEÑO ESTRUCTURADO

- El diseño es el proceso por el cual se traducen las especificaciones de requerimientos en una representación del software
- El diseño estructurado (según P. Jones) es una aproximación disciplinada al diseño de sistemas.
- No hay que confundir Diseño estructurado con Programación estructurada.

# DISEÑO ESTRUCTURADO: CARACTERÍSTICAS

- Se caracteriza por:
  - **Usar la definición del problema** para guiar la definición de la solución.
  - Abordar la complejidad de sistemas grandes por medio del **particionamiento en módulos** y su organización en jerarquía adecuadas.
  - **Utilizar herramientas** para hacer el sistema más comprensible.
    - Diagrama de estructuras (DE): representación del particionamiento en módulos. Pseudocódigo: especificación de módulos....

# DISEÑO ESTRUCTURADO: CARACTERÍSTICAS

- Se caracteriza por:
  - Definir un **conjunto de estrategias** que permitan elaborar el diseño partiendo de una especificación del problema bien definida
    - Por ejemplo a partir del DFD se puede obtener el DE mediante el análisis de transacciones y transformaciones.
  - Ofrecer **criterios objetivos** para evaluar la **calidad del diseño** (cohesión y acoplamiento).

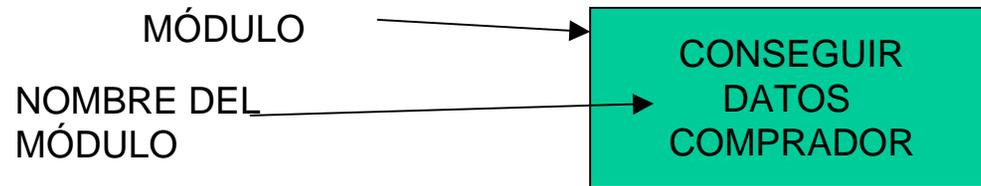
# VENTAJAS DE LOS SISTEMAS PARTICIONADOS EN MÓDULOS

- Ventajas:
  - Su construcción es sencilla
  - son probados más fácilmente
  - su corrección y modificación también es menos complicada
  - son más legibles.
- Principios que han de seguirse:
  - Cada módulo debe tener asignada una única función.
  - La función asignada debe ser fácilmente entendible.
  - Las conexiones entre los módulos han de ser las que existan entre las distintas partes del problema.
  - Las conexiones deben ser lo más simples posibles.

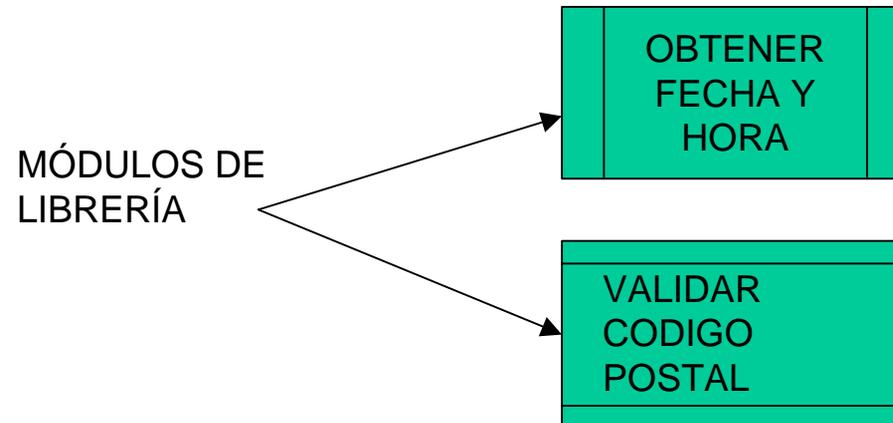
# LOS MODULOS: CARACTERÍSTICAS Y ATRIBUTOS

- Un módulo se comporta como una caja negra.
- Se define como un conjunto de sentencias de programa que poseen los siguientes atributos:
  - **Entrada:** La información que recibe cuando lo invocan.
  - **Salida:** Información que devuelve al módulo que lo invocó.
  - **Función:** Lo que hace para transformar los datos de entrada en los datos de salida.
  - **Mecánica:** La lógica mediante la cuál lleva a cabo su función.
  - **Datos internos:** Los datos a los que sólo el puede referirse (de ámbito local).
  - **Nombre:** por el cuál es referenciado.

# REPRESENTACIÓN GRÁFICA DE MÓDULOS



El nombre del módulo debe reflejar fielmente la función que cumple cuando el módulo es llamado.



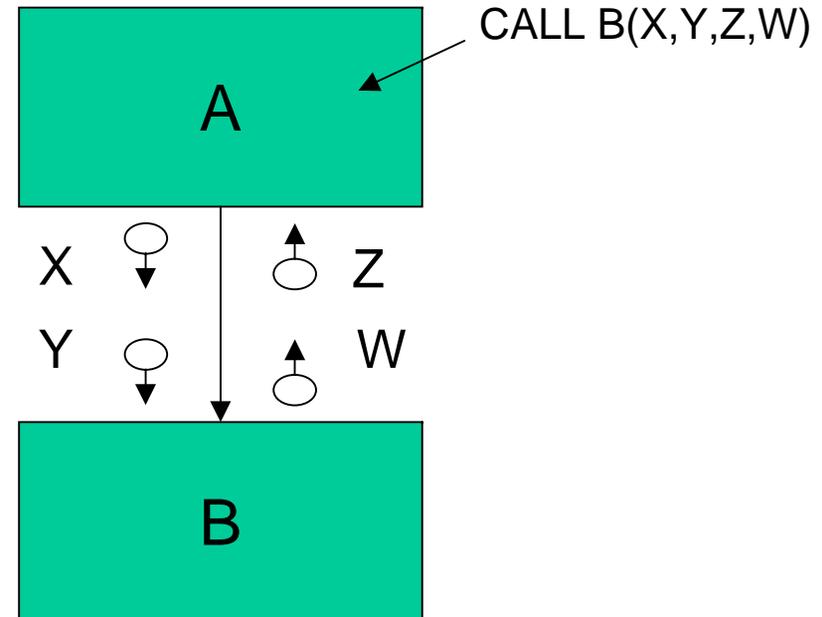
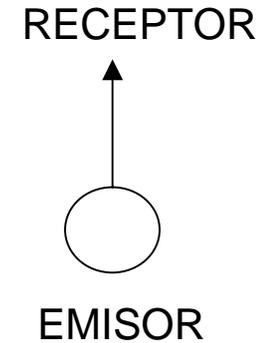
Los módulos predefinidos o de librería, son aquellos que ya han sido definidos anteriormente en algún sitio.

# REPRESENTACIÓN GRÁFICA DE LA CONEXIÓN ENTRE MÓDULOS

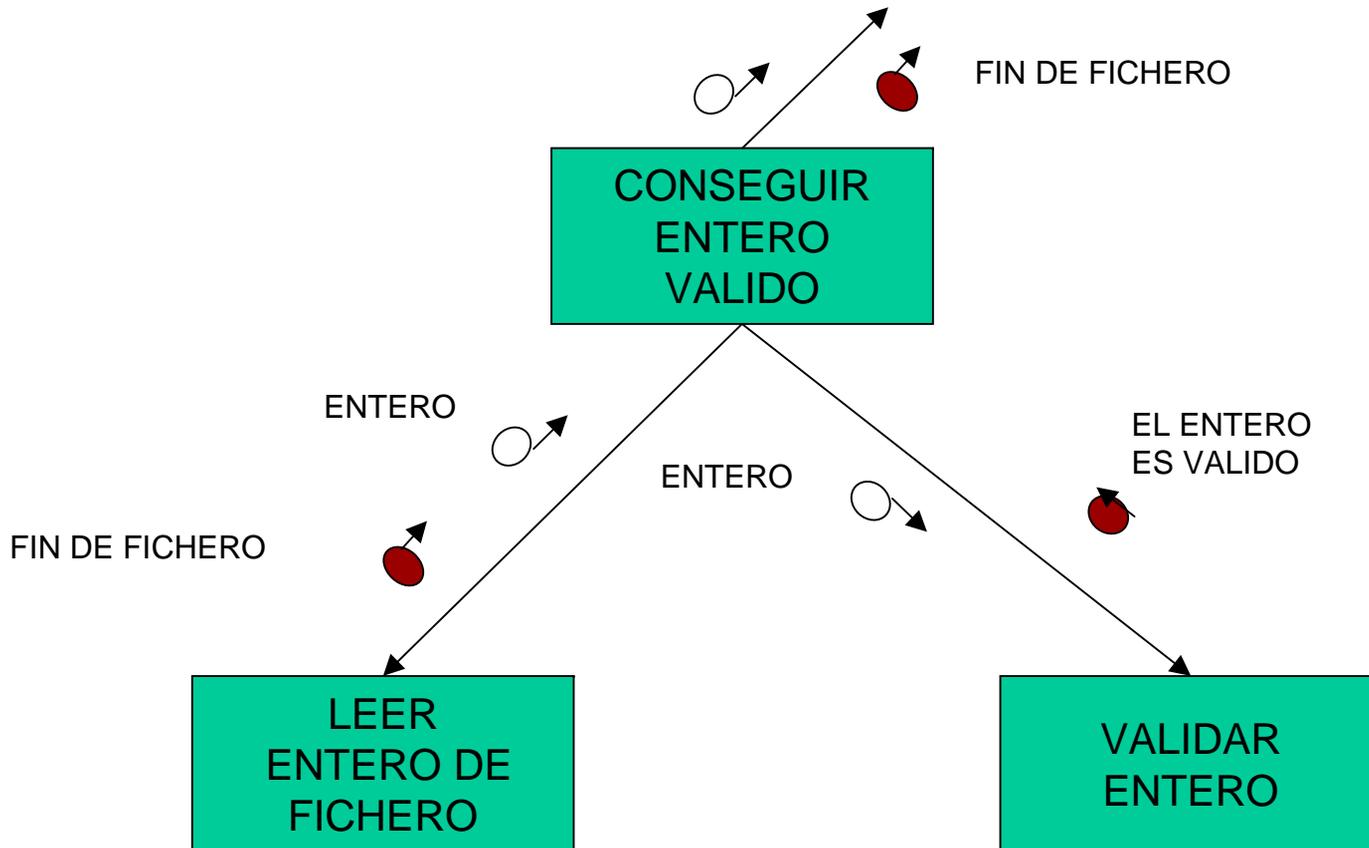
Cuando dos módulos se comunican intercambian información en forma de parámetros que aparecen en la llamada.



- Los **datos** son principalmente procesados, mientras que los **flags** son activados y chequeados.
- Los **datos** pertenecen al entorno del problema, mientras que los **flags** son entidades artificiales que se introducen para describir cualidades de los datos.



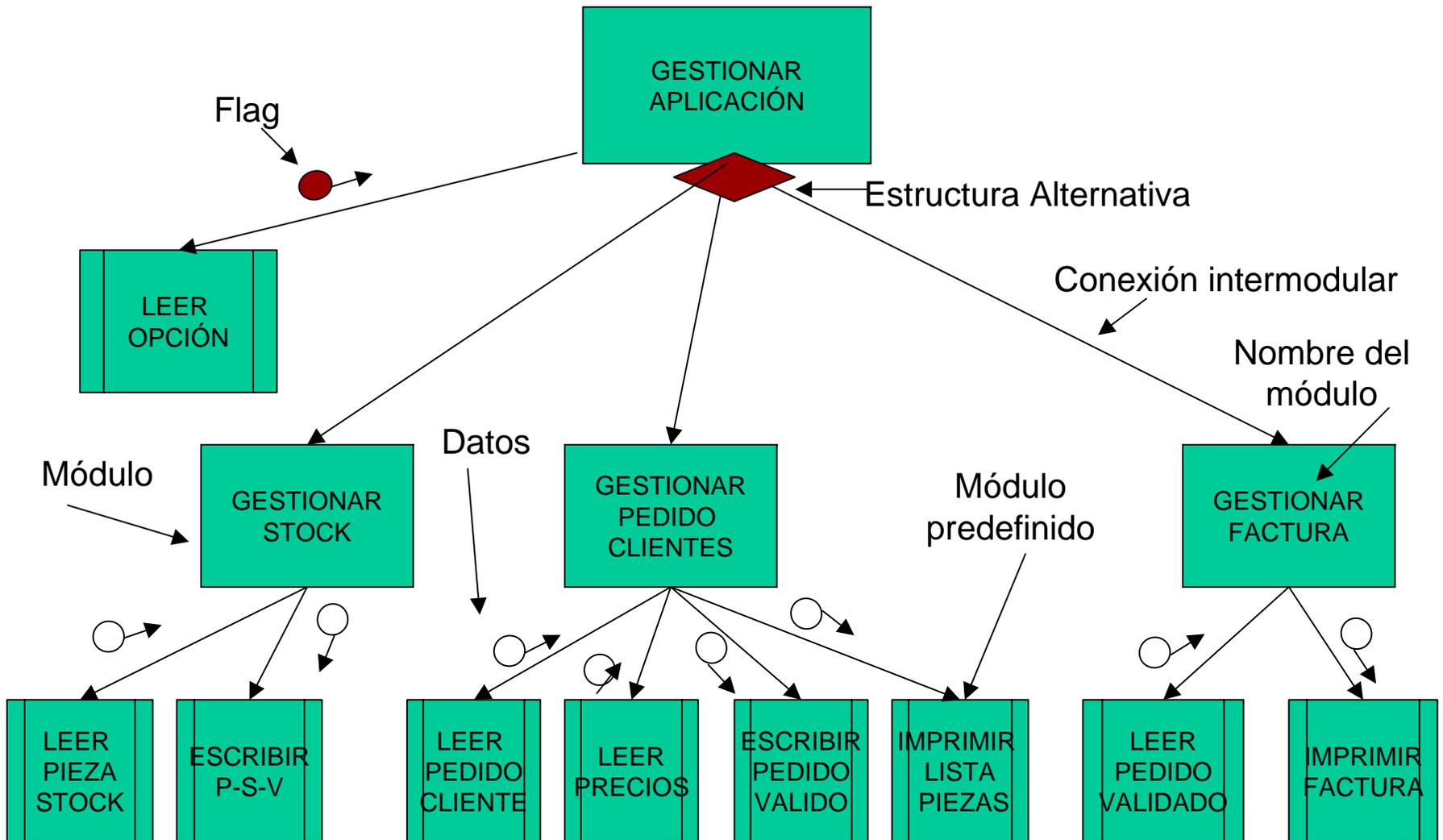
# EJEMPLO DE TRANSFERENCIA DE DATOS Y CONTROL



# EL DIAGRAMA DE ESTRUCTURA (DE)

- El diagrama de estructura (DE) es **una herramienta gráfica** que permite representar la descomposición de un sistema de módulos.
- Tiene forma de árbol
- Cada nodo se corresponde con un módulo del sistema
- Expresa la jerarquía de control que se establece entre los módulos
- Muestra los datos que componen la comunicación entre módulos (parámetros).
- No presenta aspectos procedurales.

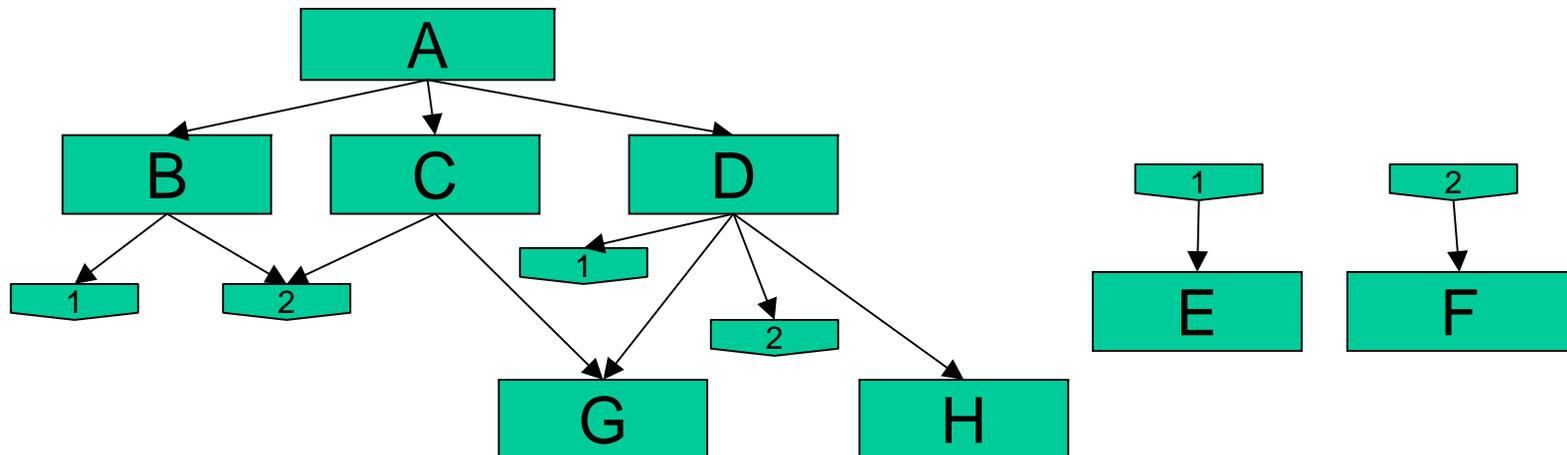
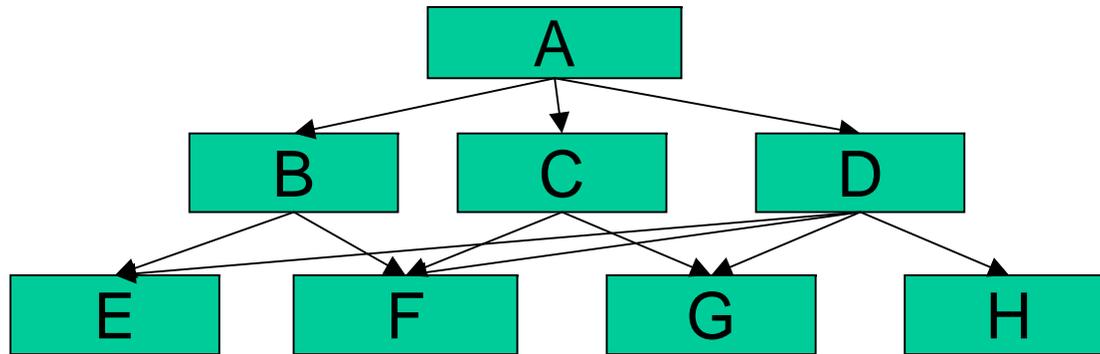
# EL DIAGRAMA DE ESTRUCTURA (DE)



# ALGUNAS OBSERVACIONES SOBRE EL D.E.

- En la medida de lo posible, es conveniente representar los módulos en el orden en el que serán invocados.
- Si los módulos predefinidos tienen subordinados, estos no se incluirán en el DE.
- El DE no debe indicar nada sobre los datos internos de los módulos, ni el nombre de los parámetros de entrada y salida.
- El nombre del módulo debe resumir no sólo la función que desempeña, sino también la de sus subordinados.

# OTRAS NOTACIONES DENTRO DEL D.E. (CONECTORES)



# METODO PARA LA ESPECIFICACIÓN DE MÓDULOS

- El propósito es transmitir al programador información complementaria que le permita llevar a cabo la generación de código.
- La especificación de un módulo debe ser **completa** en cuanto a las funciones, pero no tan restrictiva como para no dejar libertad al programador.
- Los métodos que existen son:
  - Interfaz-función
  - pseudocódigo
  - Otros métodos complementarios:
    - Diagramas de flujo
    - Diagramas de Nassi-Sneiderman
    - Tablas de decisión
    - Árboles de decisión

# INTERFAZ-FUNCIÓN

- Se indican cuales son las entradas, las salidas y la función que se espera realice el módulo.
- La función se presenta como una sentencia simple que establece la relación entre las entradas y salidas.
- Incluir detalles de funcionalidad

# EJEMPLO DE INTERFAZ-FUNCIÓN

**MÓDULO:** CALCULAR IMPORTE ARTÍCULO

**ENTRADA:**

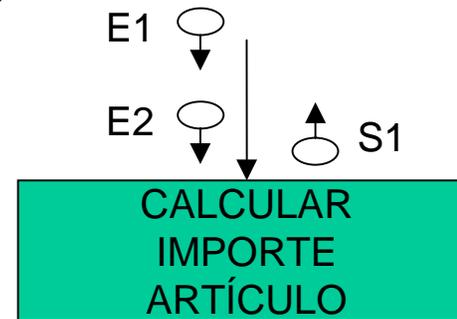
CÓDIGO-ARTÍCULO  
CANTIDAD-ARTÍCULO

**SALIDA:**

IMPORTE -ARTÍCULO

Detalles Funcionales:

- 1.- Obtiene a partir de base de datos externa el precio del artículo.
- 2.- A partir del precio y de la cantidad obtiene el valor total.
- 3.- A partir del precio total evalúa el IVA.
- 4.- Con el IVA y el valor total obtiene el importe total.



E1.- Código artículo  
E2.- Cantidad artículo  
S1.- Importe artículo

# ESPECIFICACIÓN POR PSEUDOCÓDIGO

- Es un modo más detallado que el método de la interfaz-función.
- El pseudocódigo empleado en diseño se suele denominar Lenguaje de Diseño de Programas (LDP).
- Se caracteriza por tener:
  - Una conjunto de palabras claves para representar todos los constructores estructurados.
  - Una sintaxis libre para describir las características del procesamiento.
  - Facilidades para la declaración de datos.
  - Un mecanismo de definición de subprogramas y de invocación.

# PSEUDOCÓDIGO. ITERACIÓN

- Iteración:
  - Bucle **Mientras-Hacer**  
MIENTRAS <condición> HACER  
    <Bloque de sentencias>  
FIN-MIENTRAS
  - Bucle **Repetir-hasta**  
REPETIR  
    <Bloque de sentencias>  
HASTA <condición>
  - Bucle **Para-cada**  
PARA CADA i=<valor-inicial> HASTA <valor-final>  
    <Bloque de sentencias>  
FIN-PARA

# PSEUDOCÓDIGO. SELECCIÓN

- Selección:

- **Simple**

SI <condición> ENTONCES  
    <Bloque de sentencias>  
FIN-SI

- **si-entonces-sino**

SI <condición> ENTONCES  
    <Bloque de sentencias>  
SINO  
    <Bloque de sentencias>  
FIN-SI

- **caso**

EN CASO <variable>  
variable=<valor1>  
EJECUTAR  
    <Bloque de sentencias>  
variable=<valor2>  
EJECUTAR  
    <Bloque de sentencias>  
.....  
FIN-CASO

# TABLAS DE DECISIÓN

- Es una notación que traduce las acciones y condiciones a una forma tabular.
- Los pasos a seguir para elaborar una tabla de decisiones son:
  - Listar todas las acciones involucradas en el procedimiento
  - Listar todas las condiciones que se consideren durante la ejecución del procedimiento.
  - Asociar conjuntos específicos de condiciones con acciones, eliminando combinaciones imposibles.
  - Definir reglas indicando que acciones ocurren para un conjunto de condiciones

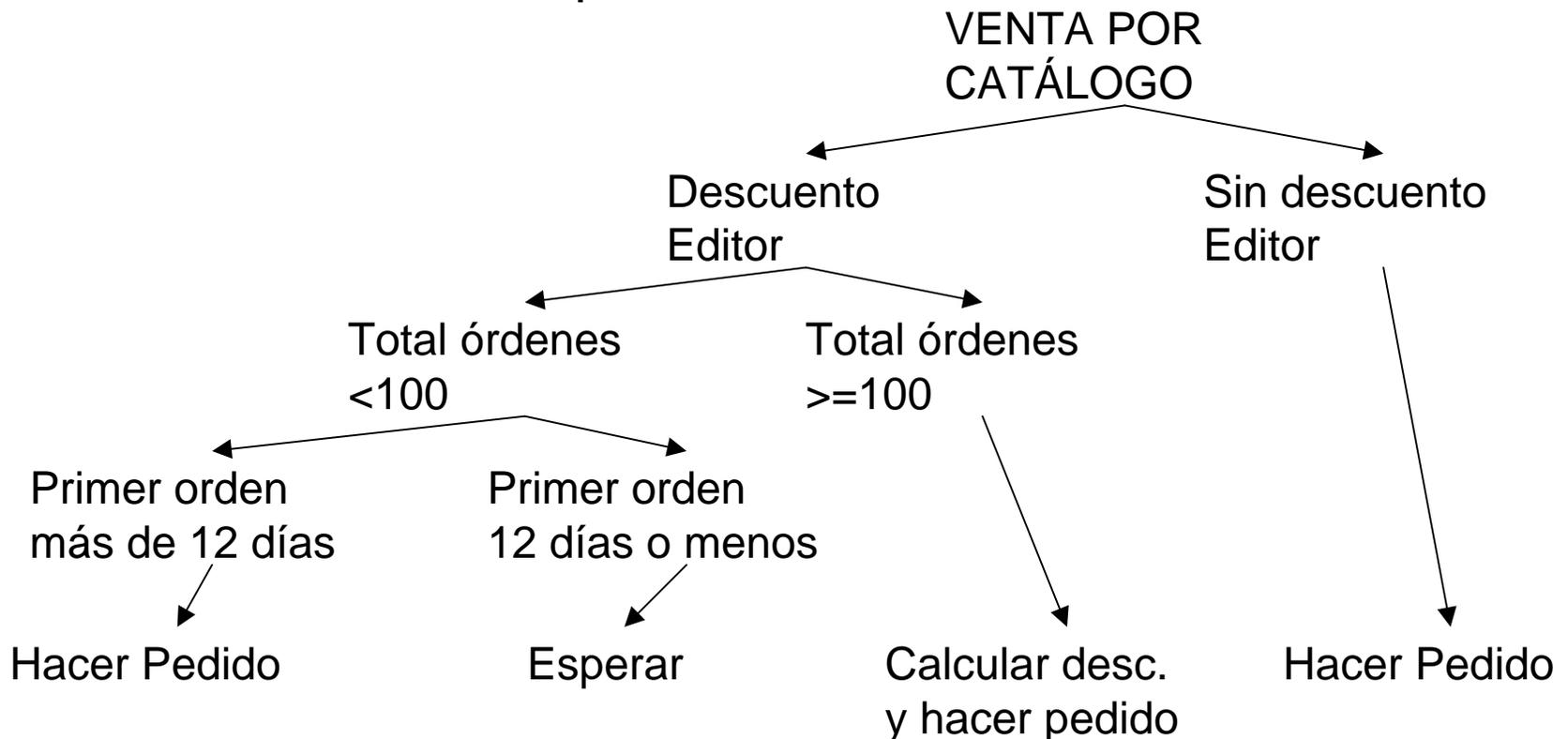
# EJEMPLO DE TABLA DE DECISIÓN

Si la cuenta del cliente se factura usando un método de tarificación fijo, se establece una carga mensual mínima para consumos menores de 100 Kwh. En los demás casos, la facturación por computadora aplica la tarifa “A”. Sin embargo, si la cuenta se factura usando un método de facturación variable, se aplicará la tarifa A a los consumos menores de 100Kwh, en otro caso, se factura de acuerdo a la tarifa “B”

		1	2	3	4
CONDICIONES	Tarifa fija	V	V	F	F
	Tarifa variable	F	F	V	V
	Consumo <100 Kwh	V	F	V	F
	Consumo >=100Kwh	F	V	F	V
ACCIONES	Cargo mensual mínimo	X			
	Tarifa A		X	X	
	Tarifa B				X

# ÁRBOLES DE DECISIÓN

- Esta representación es menos exhaustiva que la tabla de decisión respecto a la combinación de condiciones.
- Sin embargo es preferible en el caso de que el número de acciones sea reducido y no todas las combinaciones de condiciones sean posibles.

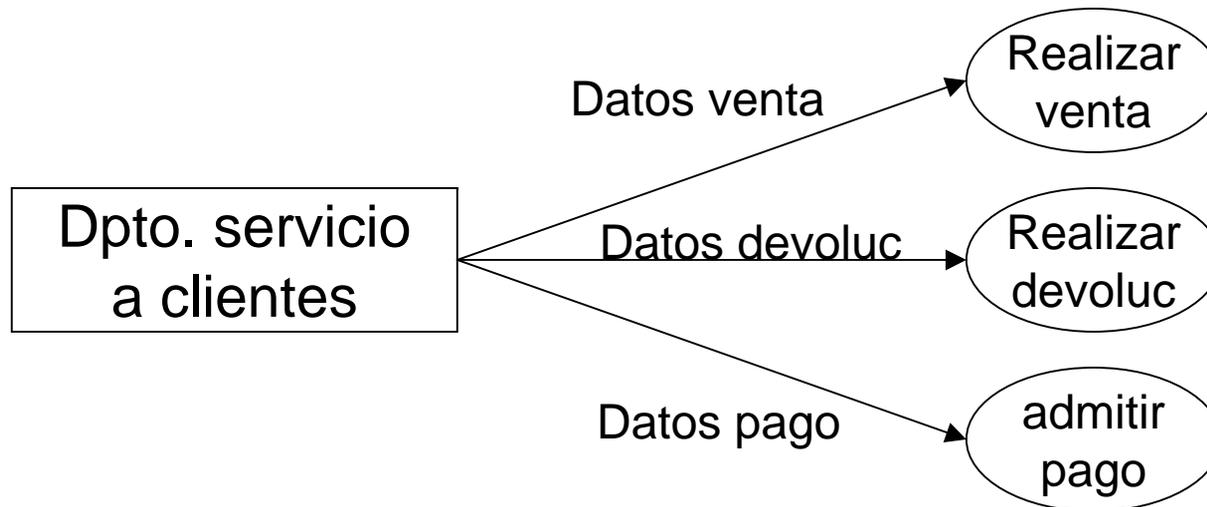


# ESTRATEGIAS PARA DERIVAR EL DIAGRAMA DE ESTRUCTURAS

- El diseño estructurado provee de un procedimiento general de transformación que permite pasar del diagrama de Flujo de Datos al Diagrama de estructura del sistema.
- El procedimiento consta de dos estrategias: análisis de transacciones y análisis de transformaciones.
- Estas dos estrategias se aplican según el siguiente orden:
  - **Análisis de transacciones:** con el objeto de determinar el número de transacciones que posee el DFD.
  - **Análisis de transformaciones:** Es el procedimiento de transformación en sí. Toma el DFD inicial y genera de forma casi mecánica un D.E.
  - **Análisis de transacciones:** Para componer los D.E en un único diagrama empleando un centro de transacciones.

# ANÁLISIS DE TRANSACCIONES

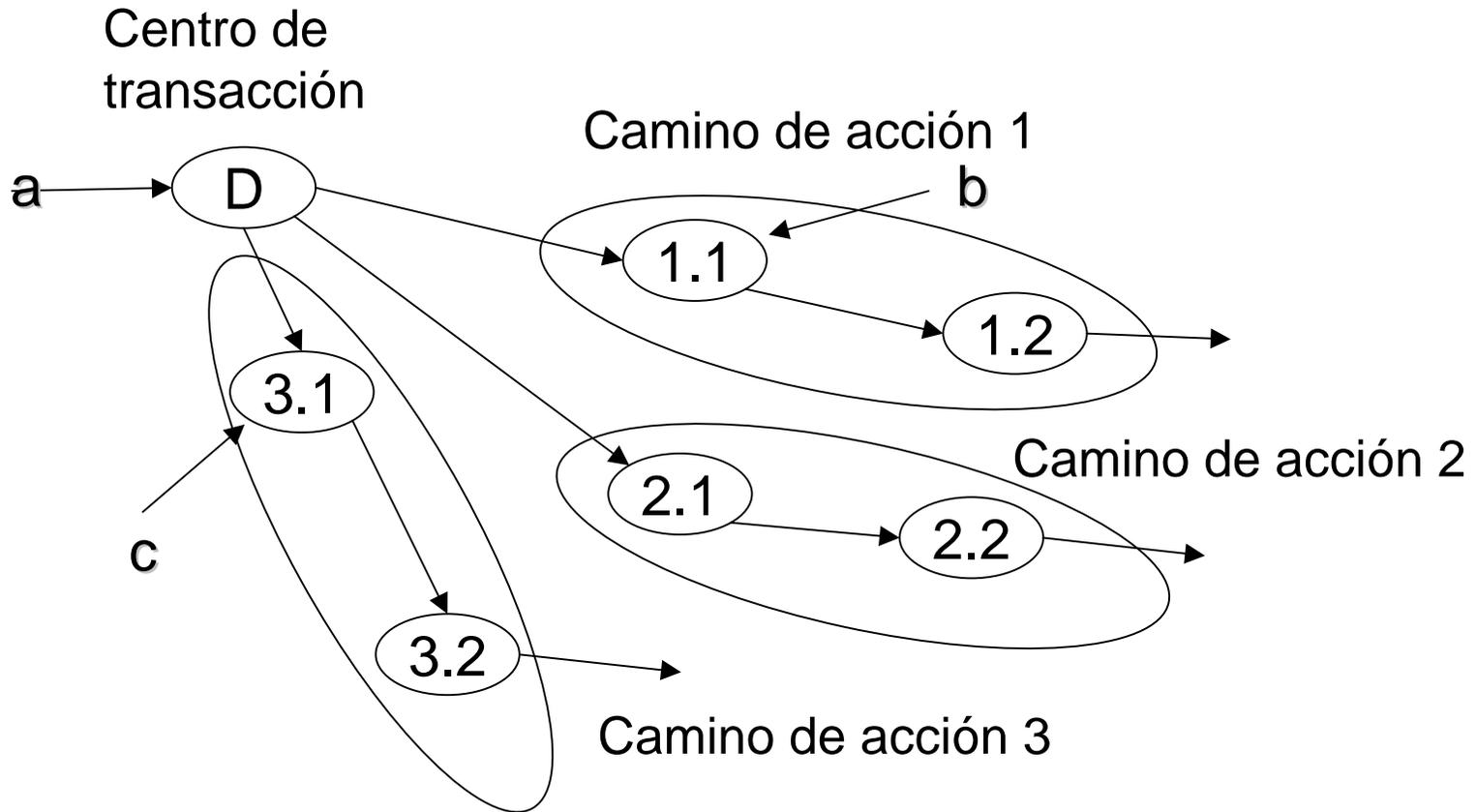
- Consiste en revisar los DFD generados en la etapa de análisis para determinar las distintas transacciones que posee el sistema.
- Una **transacción es un estímulo** (dato) que dispara o activa en un sistema un conjunto de actividades dentro del sistema.
- Cada **transacción** determina un camino de acción por donde puede discurrir el flujo de información de **forma exclusiva** respecto del resto de caminos.



# ANÁLISIS DE TRANSACCIONES: PASOS A SEGUIR

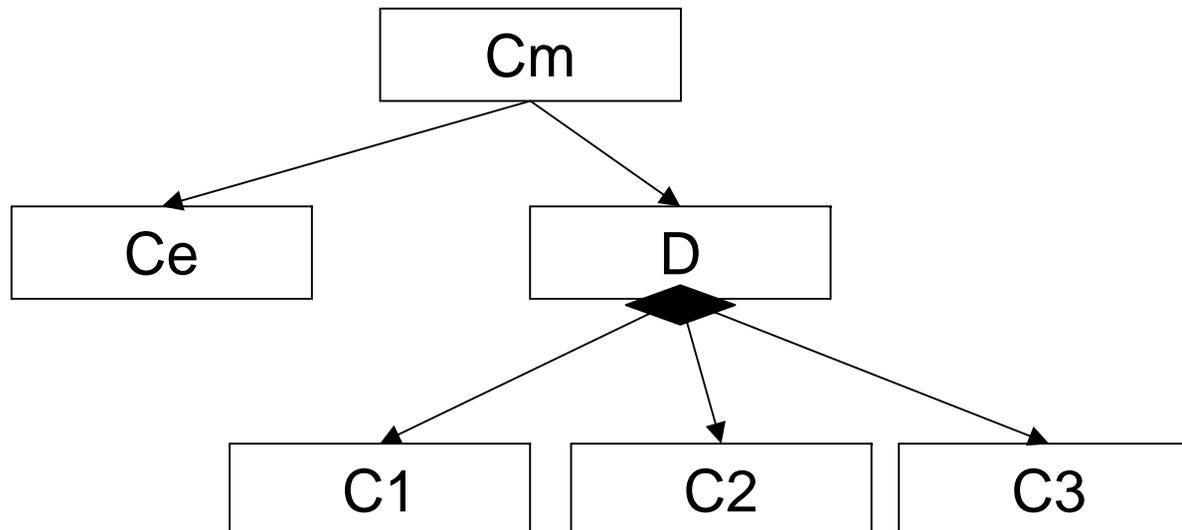
- Identificación del centro de transacciones y de los caminos de acción exclusivos.
  - El centro de transacciones está ligado al origen de varios caminos de información que fluyen radialmente de él.
  - El centro de transacciones puede aparecer, o no, de forma explícita en el DFD.
  - Un DFD puede no tener centros de transacción.
- Primer corte del Diagrama de Estructura

# ANÁLISIS DE TRANSACCIONES: IDENTIFICACIÓN DE LOS CENTROS DE TRANSACCIONES



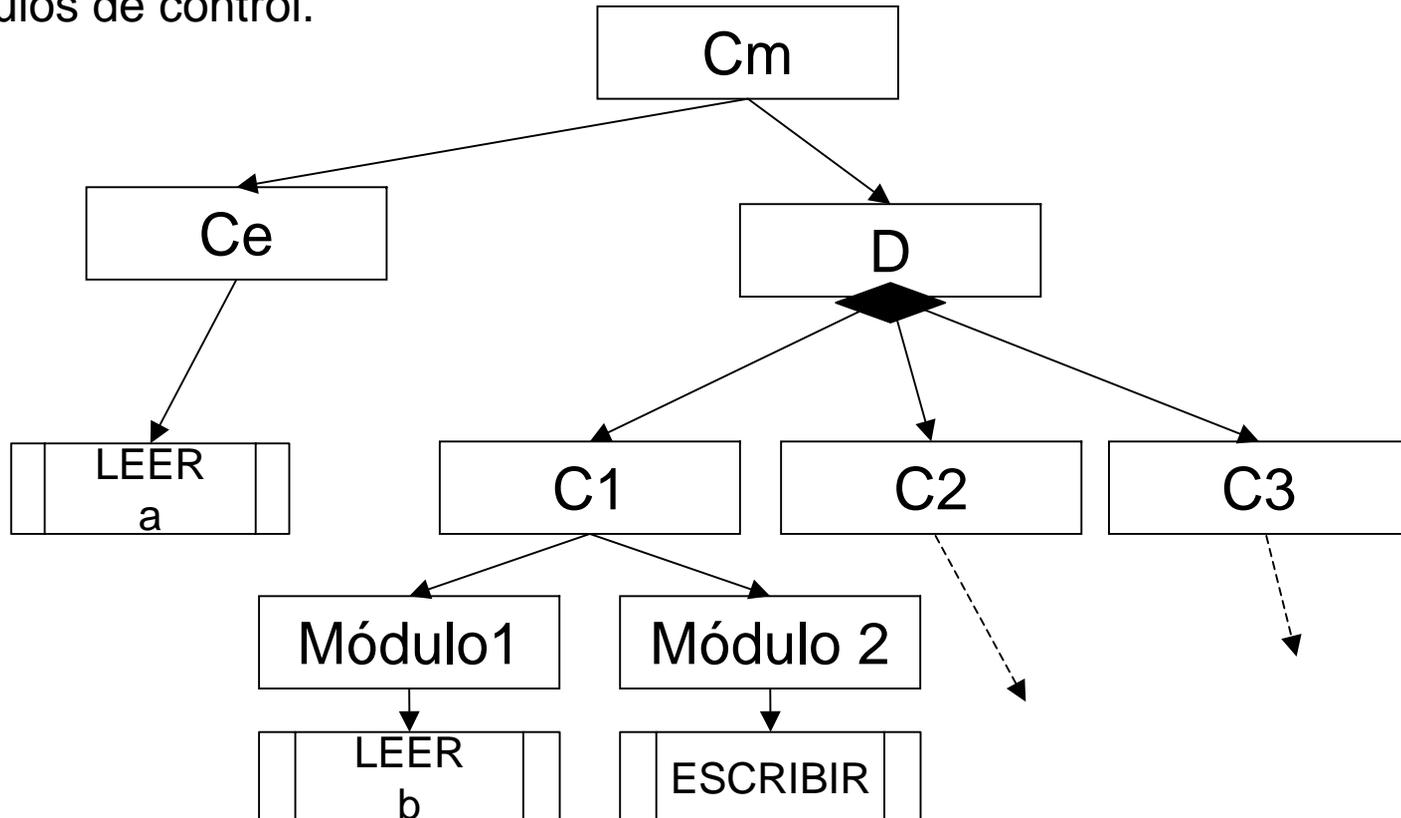
# ANÁLISIS DE TRANSACCIONES: PRIMER CORTE DEL D.E.

- El primer corte del D.E. consta de los siguientes módulos:
  - **Cm** :Módulo coordinador
  - **Ce**: Módulo controlador del procesamiento de la Información de llegada
  - **D**: Módulo Centro de transformacion.
  - **C1,2,3**: Módulos controladores de cada camino de acción
- Todos estos módulos deben tener un nombre significativo que refleje todo lo que hay por debajo de ellos



# PRIMER CORTE DEL D.E.

- Incluir los módulos que dependen del módulo de control de entrada
- Aplicar análisis de transformación para cada camino de acción susceptible de poseer un centro de transformación con objeto de definir el D.E, de cada uno de los caminos.
- Si no hubiera ningún centro de transformación, finalizar el análisis de transacción situando los módulos correspondientes bajo cada uno de los módulos de control.



# ANÁLISIS DE TRANSFORMACIONES

- Consiste en revisar los DFD generados en la etapa de análisis para determinar los centros de transformación que posee el sistema.
- Un centro de transformación es la parte de un DFD que contiene las funciones esenciales del mismo y es independiente de una implementación particular de la entrada/salida.
- Puede consistir en uno o varios módulos.

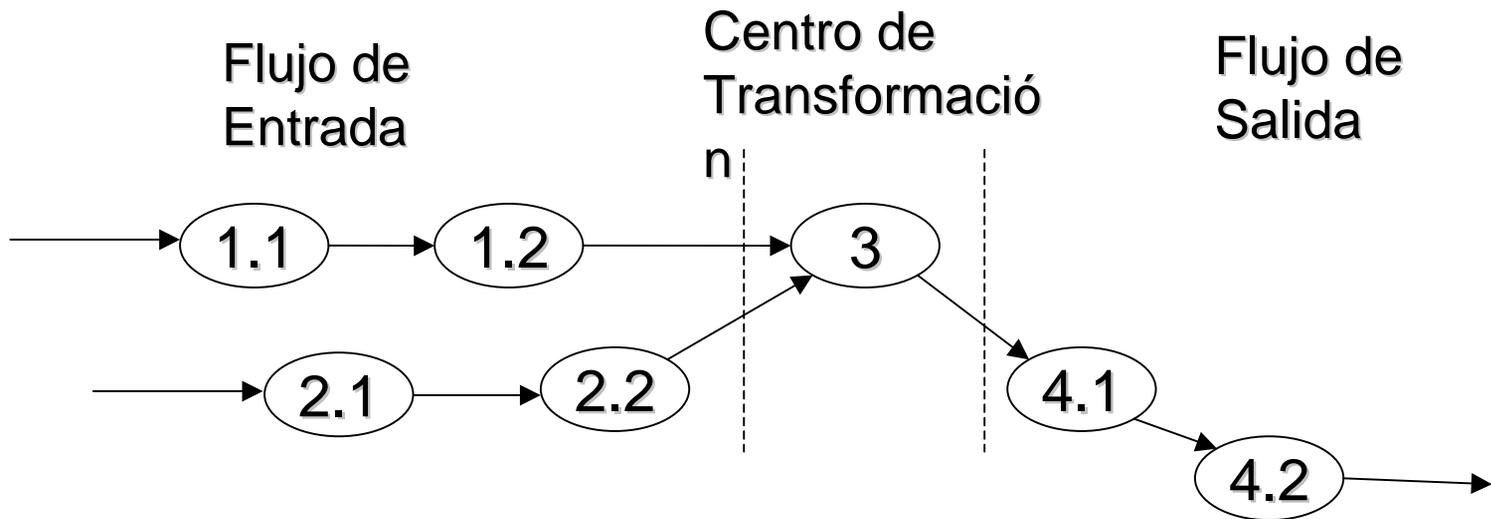
# ANÁLISIS DE TRANSFORMACIONES

- Los pasos a dar son los siguientes:
  - Identificar las funciones centrales del DFD o la transformación central.
  - Convertir el DFD en una primera aproximación o corte al D.E.
    - Primer nivel de factorización
    - Segundo nivel de factorización
  - Refinar el D.E. Mediante los criterios de diseño, etc..
  - Comprobar que el D.E. Final verifica los requerimientos del DFD inicial

# IDENTIFICACIÓN DE LA TRANSFORMACIÓN CENTRAL

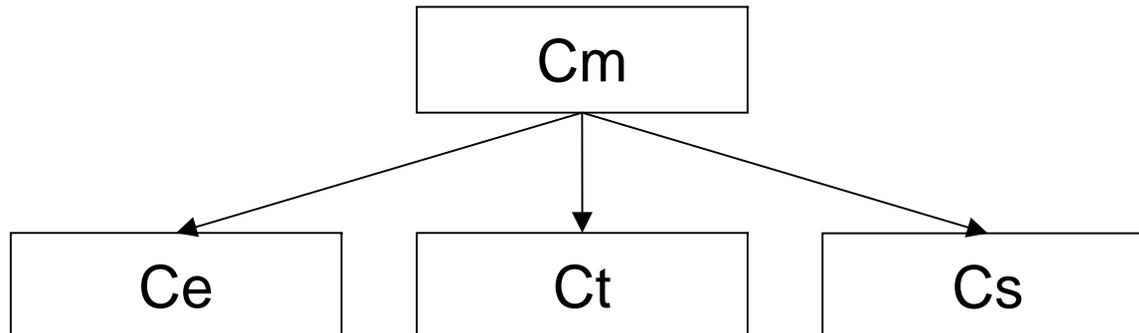
- Para detectar la transformación central se recorren las ramas de entrada y salida de la siguiente forma:
  - Cada rama de entrada se recorre desde el exterior hacia el interior del DFD buscando el proceso donde empieza a tener lugar la transformación. Se marca este lugar.
  - Cada rama de salida se recorre desde el exterior hacia el interior del DFD buscando el lugar donde los datos ya han sido procesados. Se marca el lugar .
  - Si se unen los puntos marcados, los procesos encerrados dentro de esta línea de demarcación componen la transformación central.

# ANÁLISIS DE TRANSFORMACIONES: IDENTIFICACIÓN DEL CENTRO DE TRANSFORMACIÓN

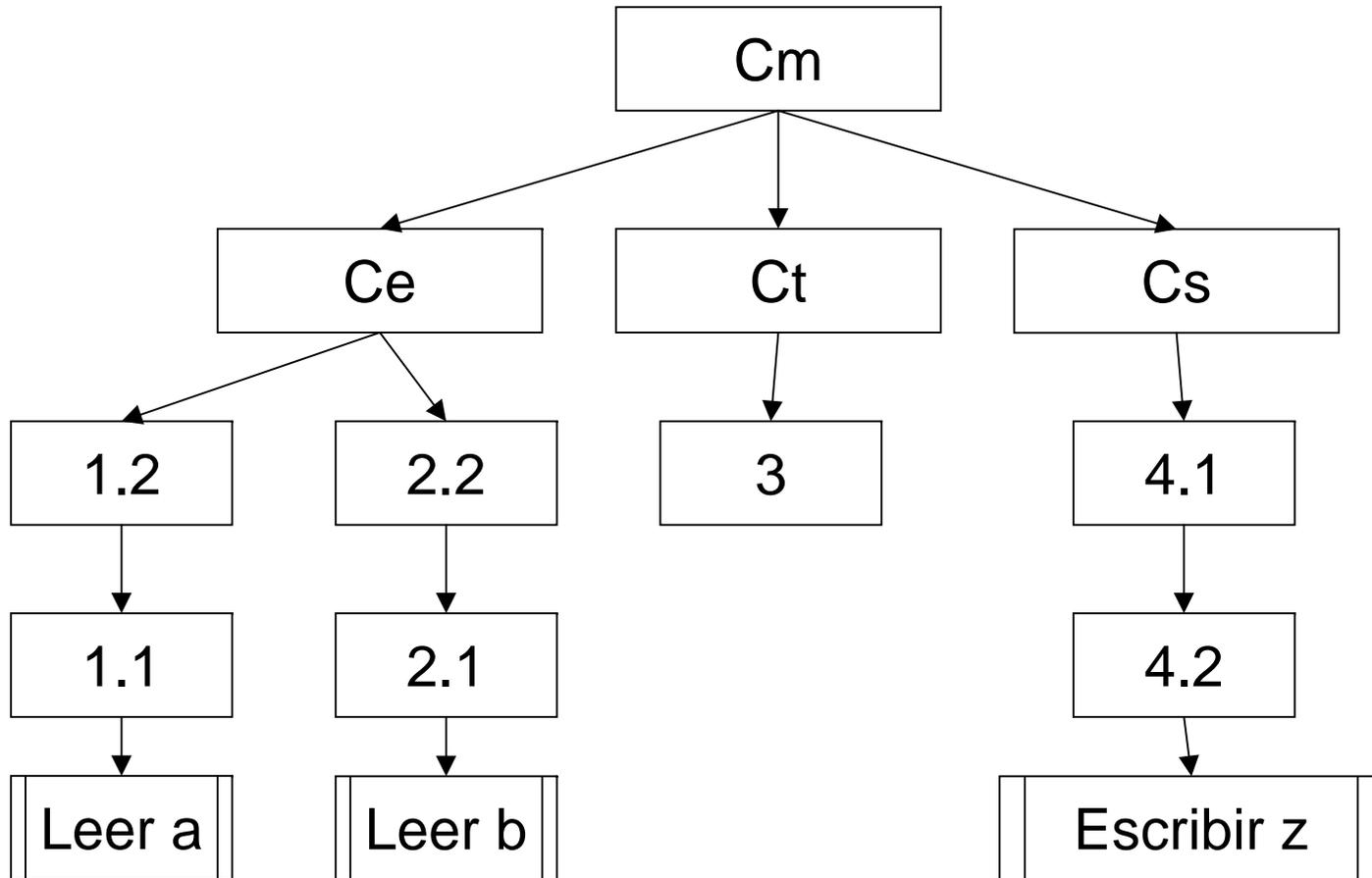


# ANÁLISIS DE TRANSFORMACIONES: PRIMER NIVEL DE FACTORIZACIÓN

- El primer nivel de factorización consta de los siguientes módulos:
  - **Cm** :Módulo cordinador
  - **Ce**: Módulo controlador del procesamiento de la Información de llegada
  - **Ct**: Módulo controlador del Centro de transformacion.
  - **Cs**: Módulo controlador del procesamiento de la Información de llegada
- Todos estos módulos deben tener un nombre significativo que refleje todo lo que hay por debajo de ellos

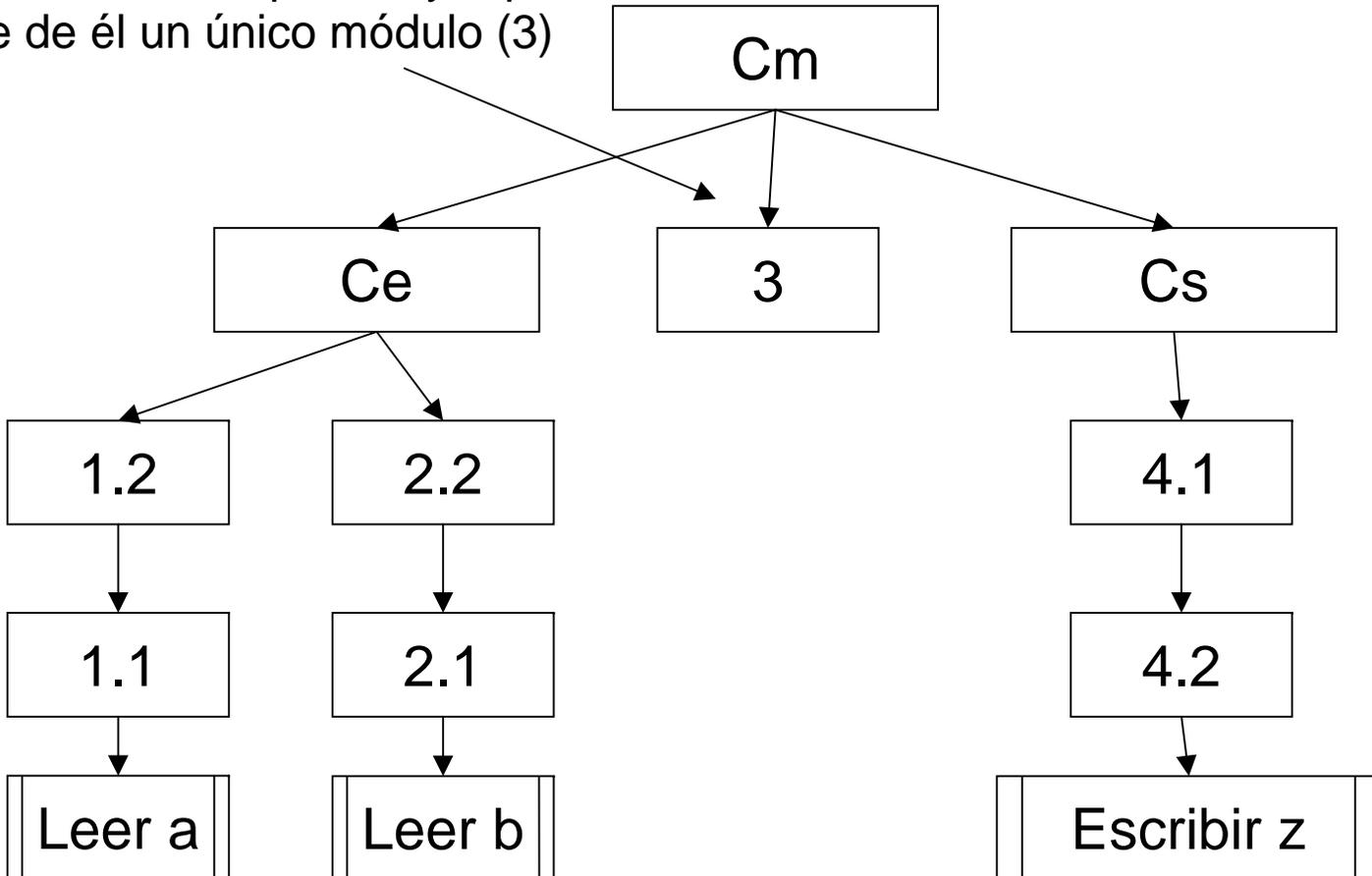


# ANÁLISIS DE TRANSFORMACIONES: SEGUNDO NIVEL DE FACTORIZACIÓN



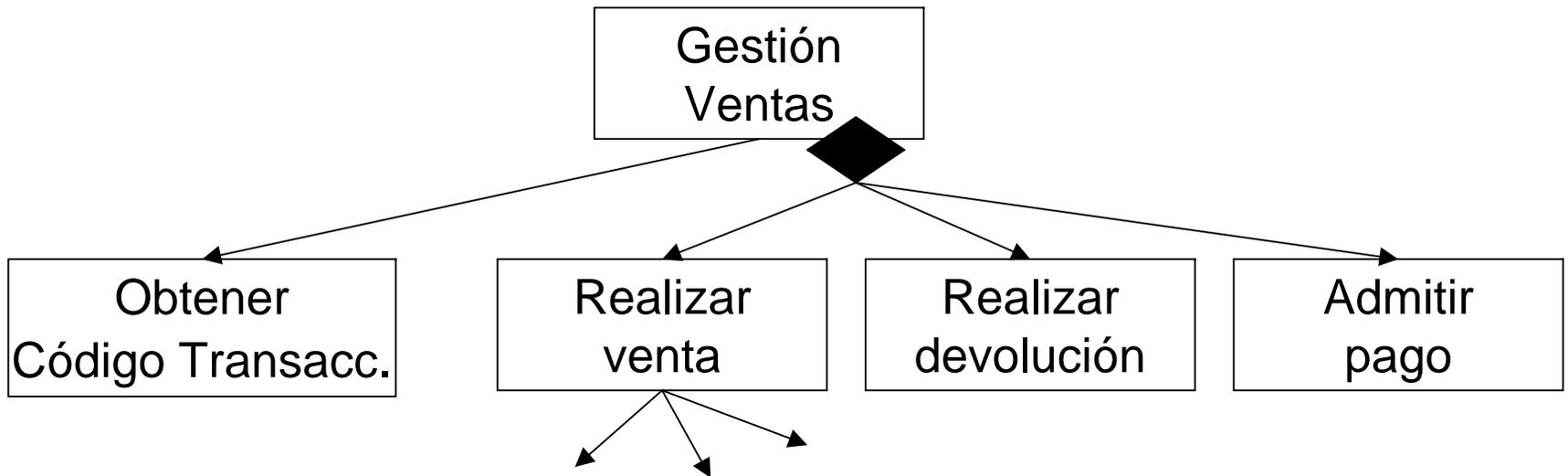
# ANÁLISIS DE TRANSFORMACIONES: REFINAMIENTOS

El módulo de control del centro de Transformación desaparece ya que sólo depende de él un único módulo (3)



# RECONSTRUCCIÓN DEL SISTEMA

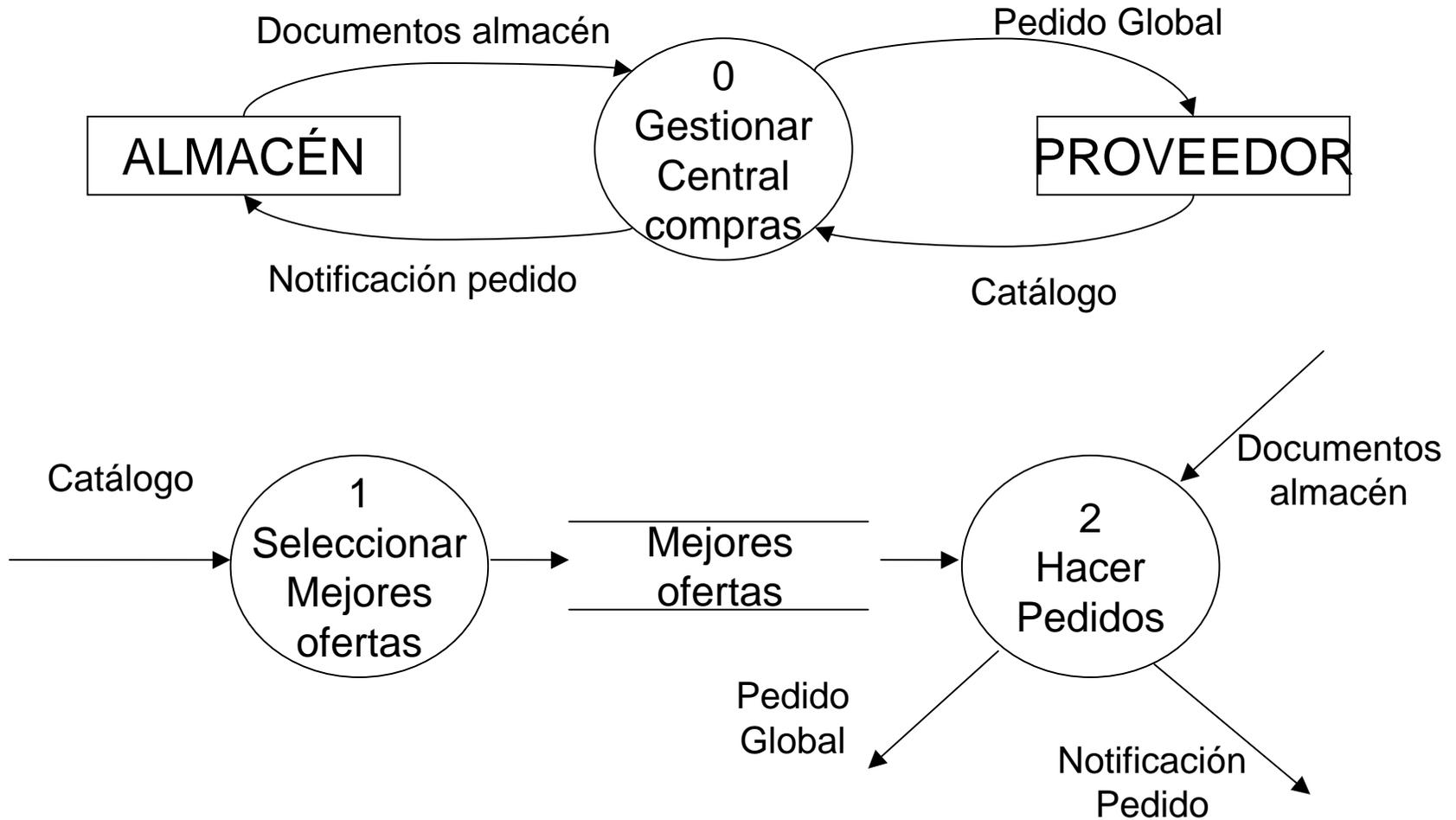
- Se juntan todos los D.E. individuales en un único D.E.



# EJEMPLO 1

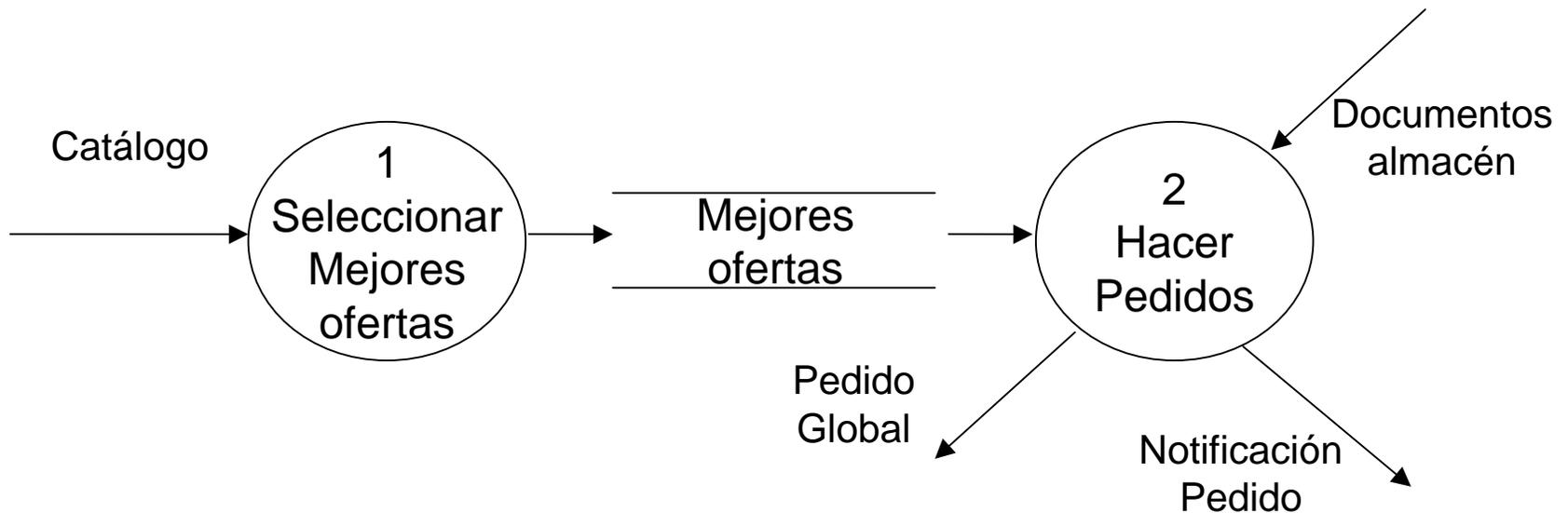
- Desarrollo de un sistema de información que apoye la gestión de una central de compras que permita realizar pedidos globales por temporada.
- Vamos a considerar que las dos operaciones básicas, selección de mejores ofertas y la realización del pedido según ofertas, no son excluyentes sino que se realizan de forma consecutiva, es decir, primero se evalúan las ofertas y a continuación se realiza el pedido global acompañado de información adicional (histórico). Esto implica que una vez se le ha proporcionado al sistema el catálogo y los documentos de almacén el sistema proporciona la documentación para realizar el pedido global.
- El DFD del sistema es el que sigue:

# EJEMPLO 1 (DFD expandido)

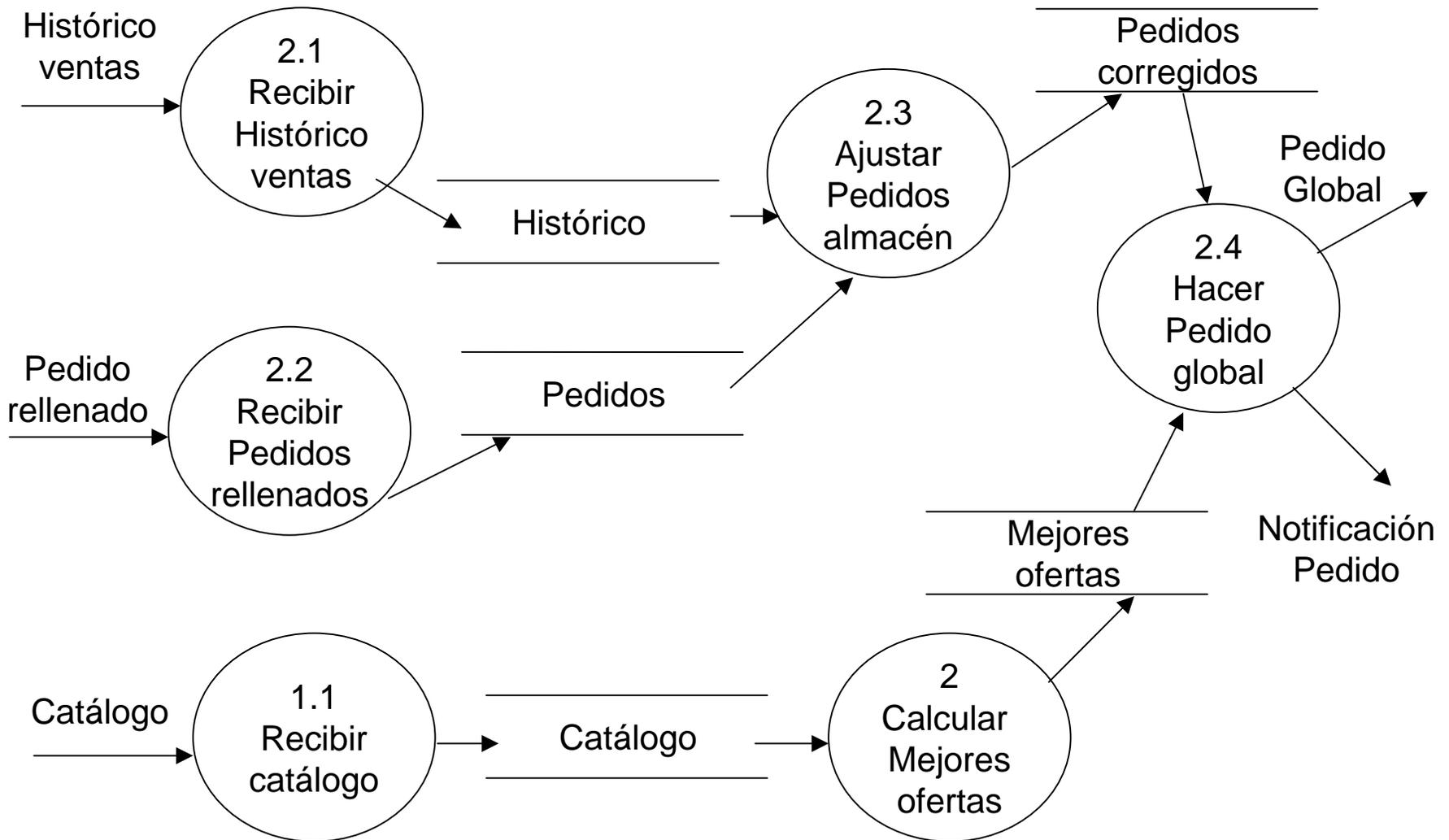


# EJEMPLO 1 (DFD expandido)

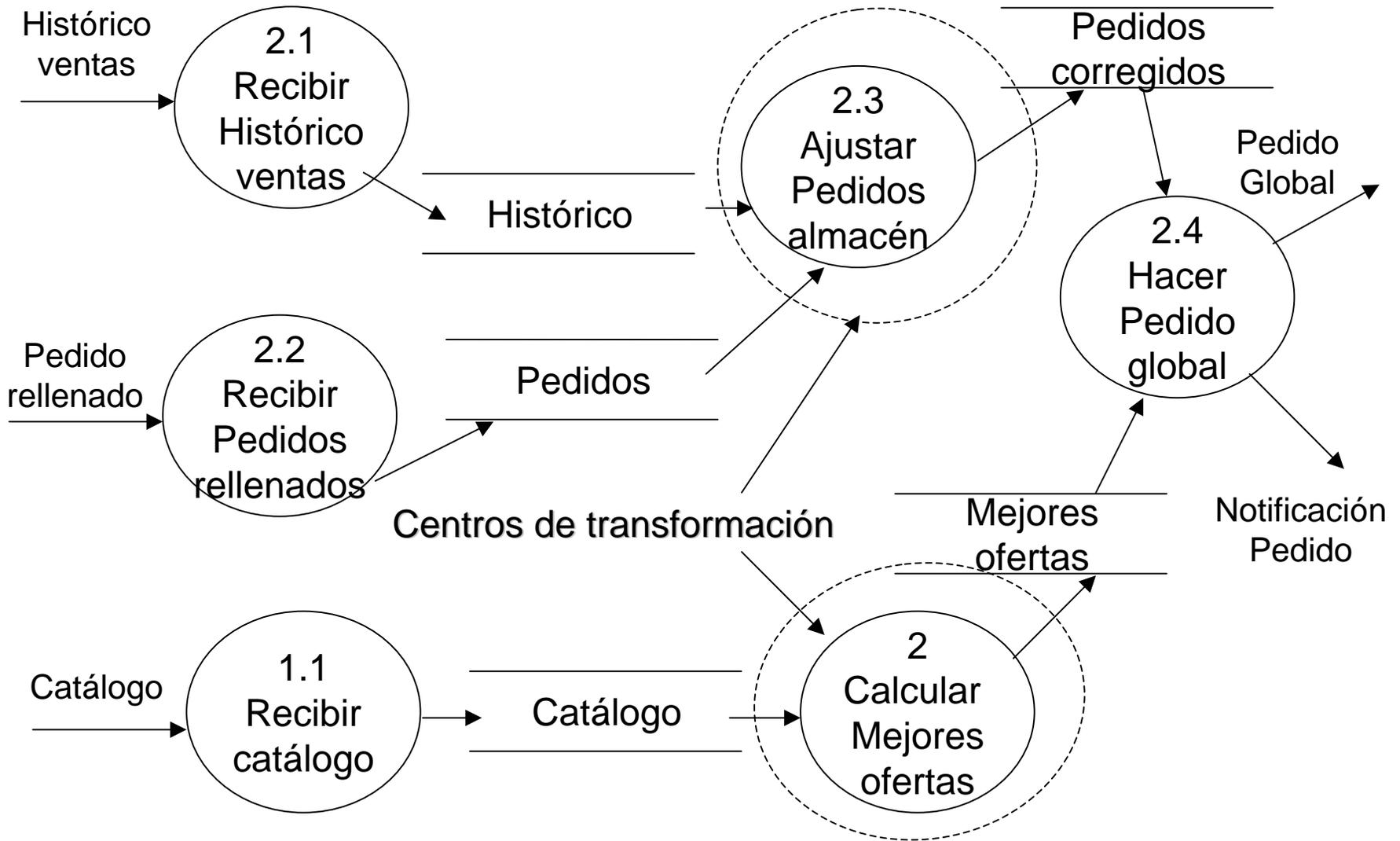
- No hay centro de transacciones



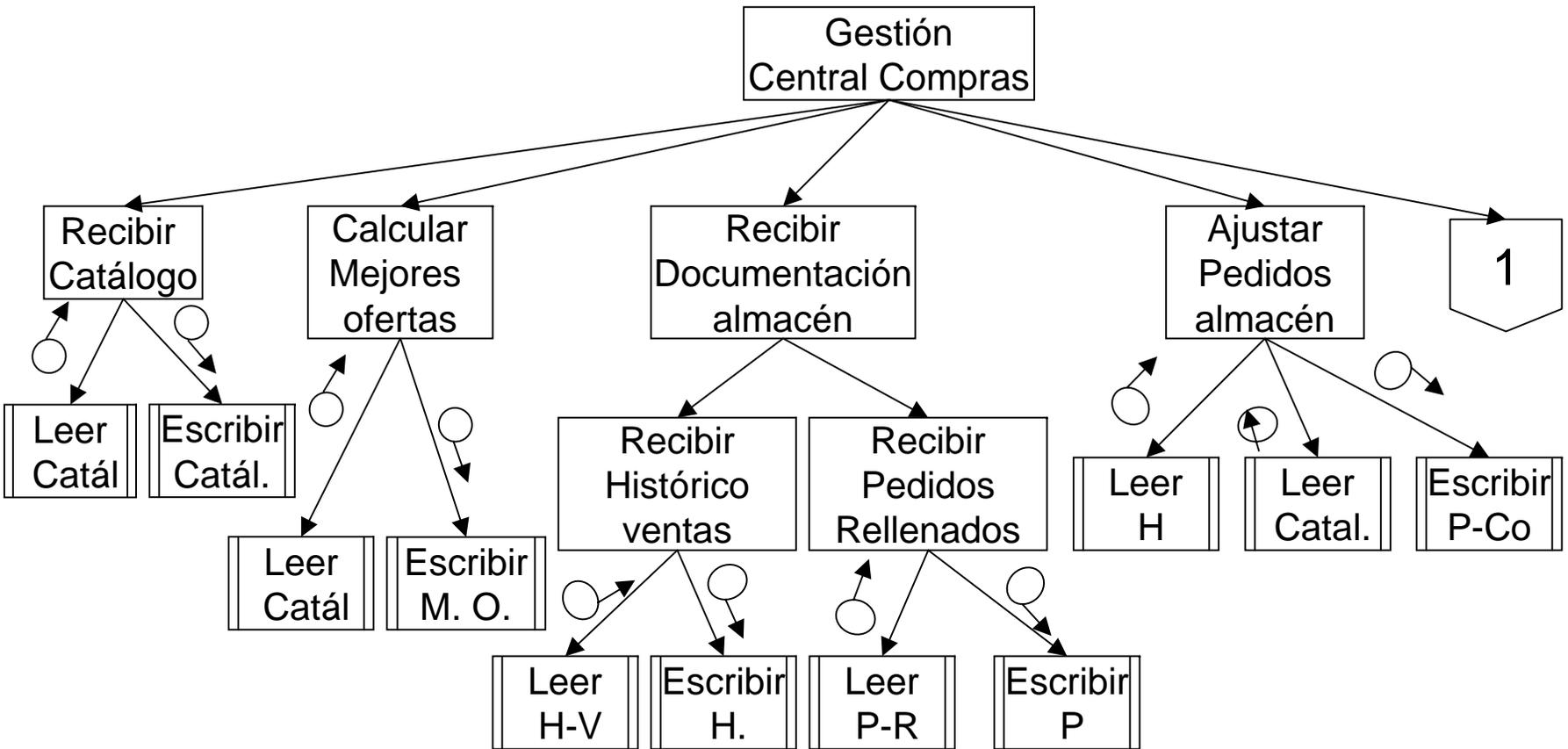
# EJEMPLO 1 (DFD expandido)



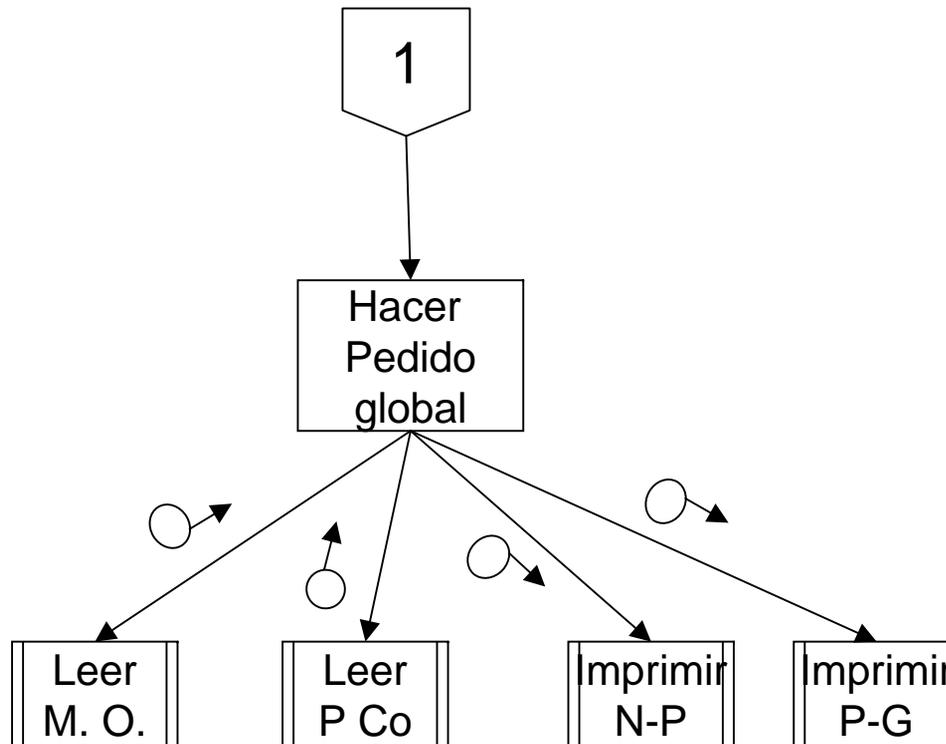
# EJEMPLO 1 (Centros de transformación)



# DIAGRAMA DE ESTRUCTURA



# DIAGRAMA DE ESTRUCTURA



# EJEMPLO 1

- Si se hubiera considerado en el ejemplo que los dos procesos que aparecen en el diagrama de sistema son exclusivos entre sí (menú) el diagrama de estructura del sistema cambia sustancialmente ya que aparece un centro de transacciones.

# DIAGRAMA DE ESTRUCTURA

