

# Diseño Estructurado

Sistemas de Información II  
Juan Carlos Ramos @2016

# Diseño Modular

---

- ▶ Descomposición funcional sistematizada
- Acuerdo general en que:
  - Los sistemas más fáciles de cambiar están constituidos por módulos *manejablemente* pequeños,
  - cada uno de los cuales es *independiente*, hasta donde es posible
  - de manera que pueden sacarse del sistema, cambiarse, y reponerse sin afectar el resto del sistema.

# Diseño Modular

---

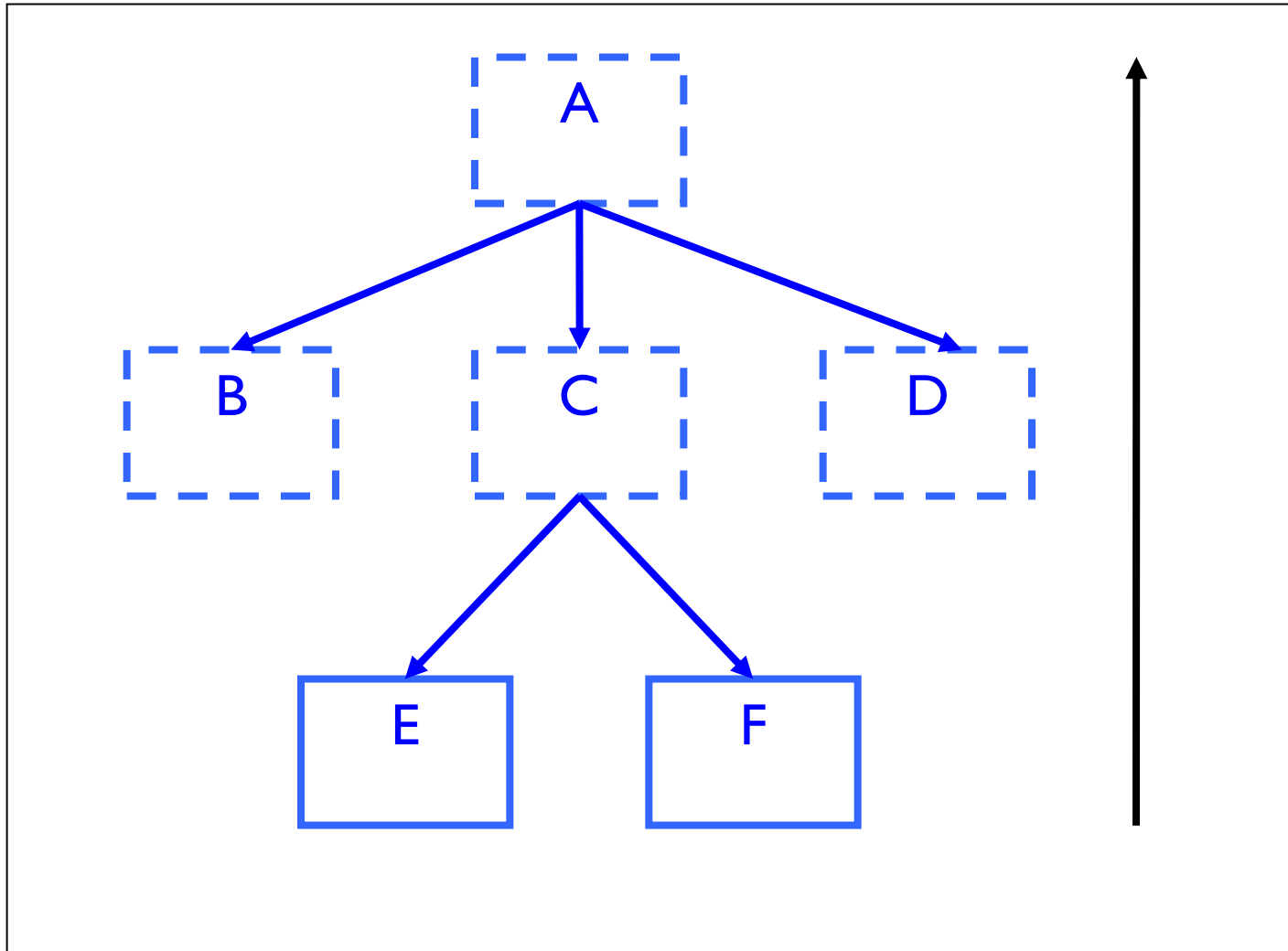
- ▶ La tarea del diseñador es formar los módulos y diseñar sus interconexiones para minimizar la interconexión entre estos, y a la vez reducir la posibilidad del efecto onda (un cambio en un módulo repercute en otros módulos, y así).
- Un diseño modular:
  - reduce la complejidad,
  - facilita los cambios, y
  - produce una implementación más sencilla,
  - permitiendo el desarrollo paralelo.

# Diseño Estructurado

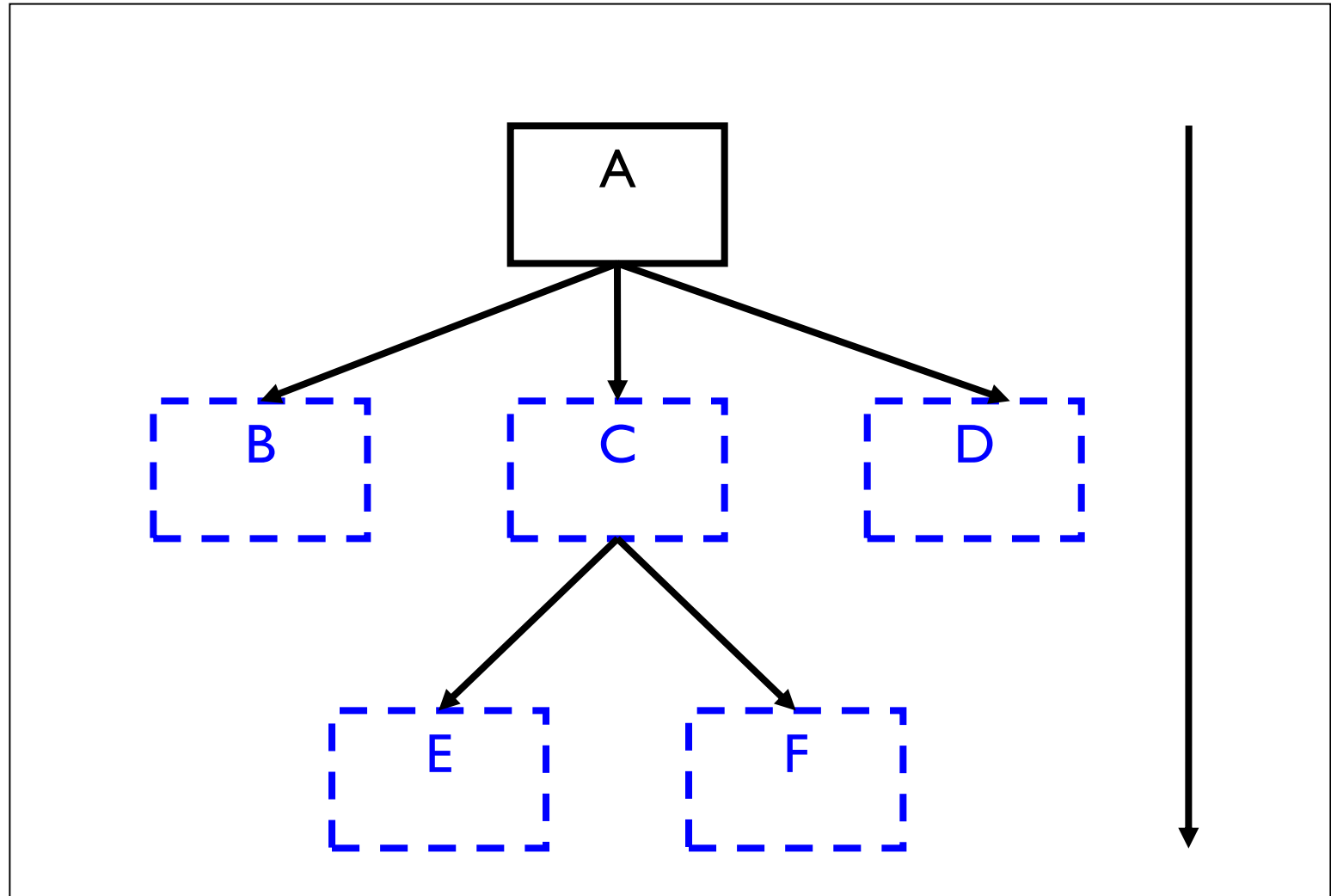
---

- Fases
  - Representar el sistema como un Diagrama de Flujos de Datos (DFD).
  - Estructurar el sistema como jerarquías de procesos (utilizando DE).
  - Realizar Análisis transformacional.
  - Realizar Análisis transaccional.
  - Verificar proyecto y reestructurar.
    - Cohesión, Acoplamiento
    - Tamaño de interfaces
    - Etc.
  - Descomposición de procesos
    - Refinamiento sucesivo
    - Factorización
  - Preparar para la implementación.

# Estrategia Ascendente



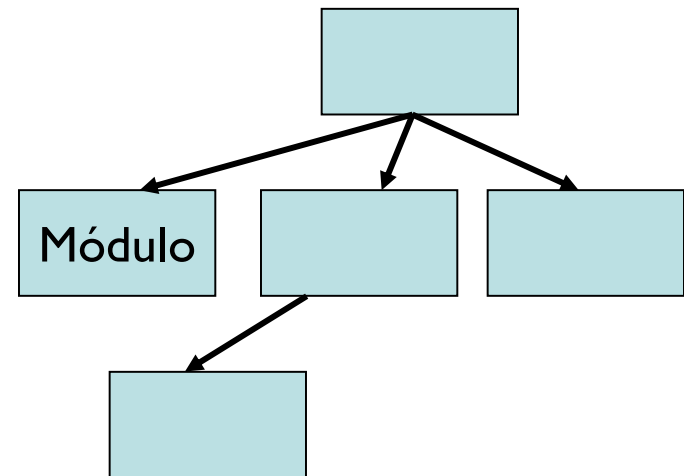
# Estrategia Descendente



# Diagramas Estructurado



---

- **Diagramas Estructurados (DE)**
  - Describen una arquitectura de programas a través de una jerarquía de llamadas a módulos.
- **Módulo**
  - Representación de una unidad de software, que ejecutará alguna función/proceso.
  - Nombre del módulo: escrito dentro del rectángulo. Descriptivo de la tarea que realiza el módulo.



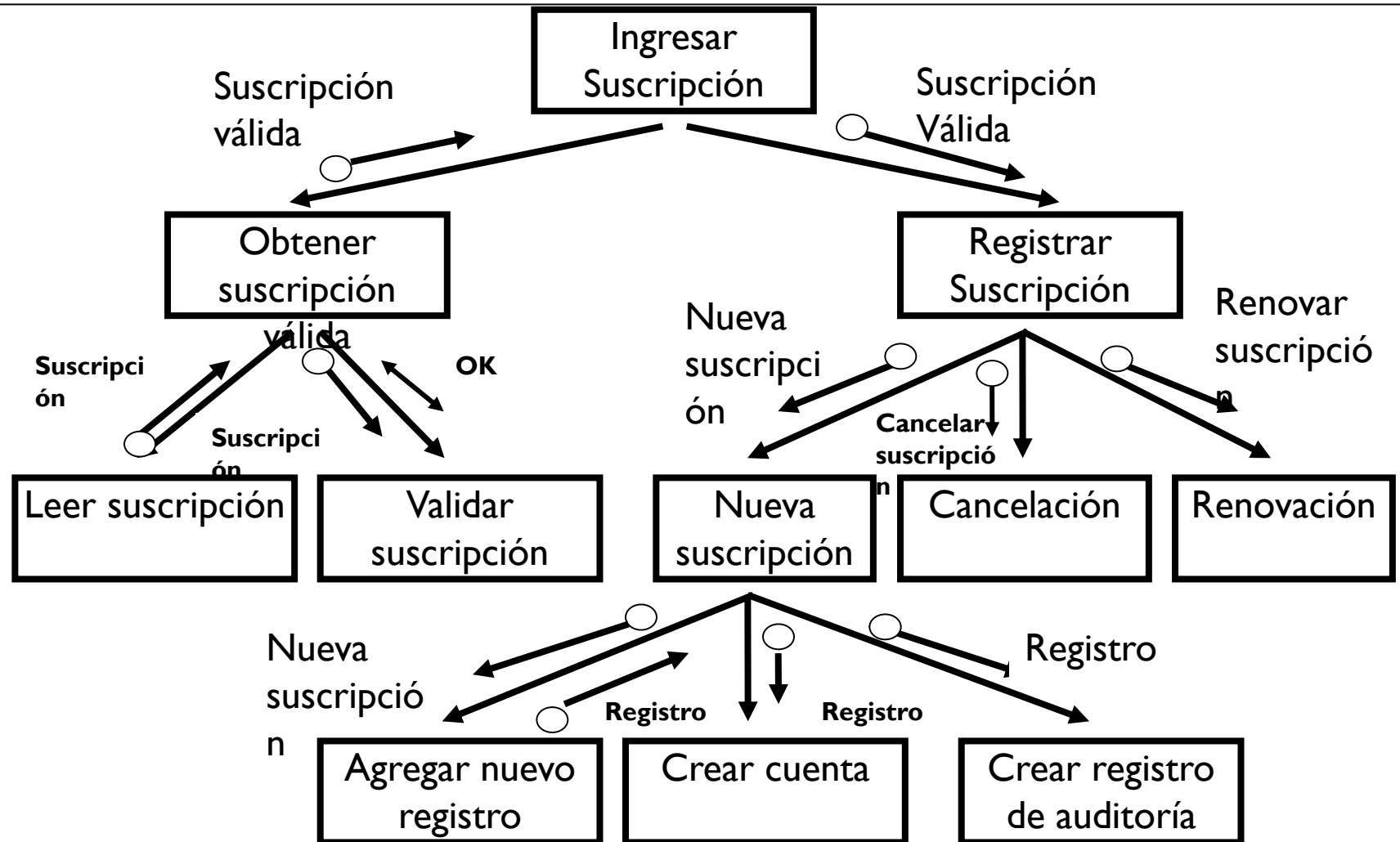
# Diagramas Estructurados

---

- Comunicación entre módulos
  - La estructura es de “llamada” o “invocación” de módulos.
- Los módulos pueden intercambiar:
  - **Datos:** representado por una flecha con un círculo vacío en un extremo.
  - **Señales:** representado por una flecha con un círculo lleno en un extremo.



# Diagramas Estructurados



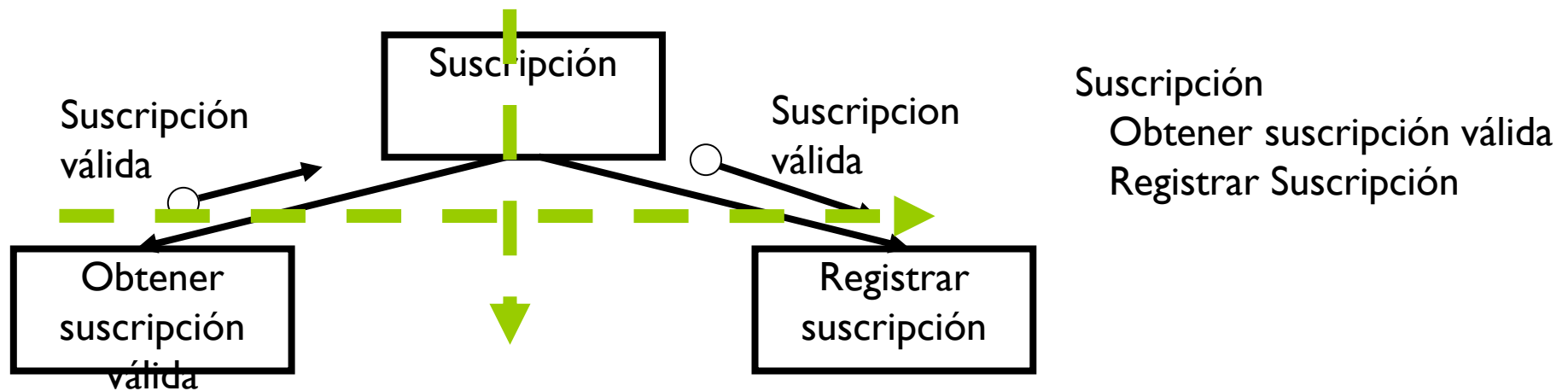
# Diagramas Estructurados

- Reglas de control para un diagrama estructurado
  - Hay uno y sólo un módulo al tope de la jerarquía (nivel I) del DE.
  - Este módulo es el llamado *raíz* (root).
  - Desde la raíz el control es pasado hacia *abajo* nivel por nivel a los otros módulos.
  - El control *siempre es devuelto* al módulo invocante. Por esto, cuando la ejecución del programa finaliza, el control regresa al root.
  - Hay a lo sumo una relación de control entre dos módulos cualesquiera. Si el módulo A invoca a B, B no puede invocar a A.



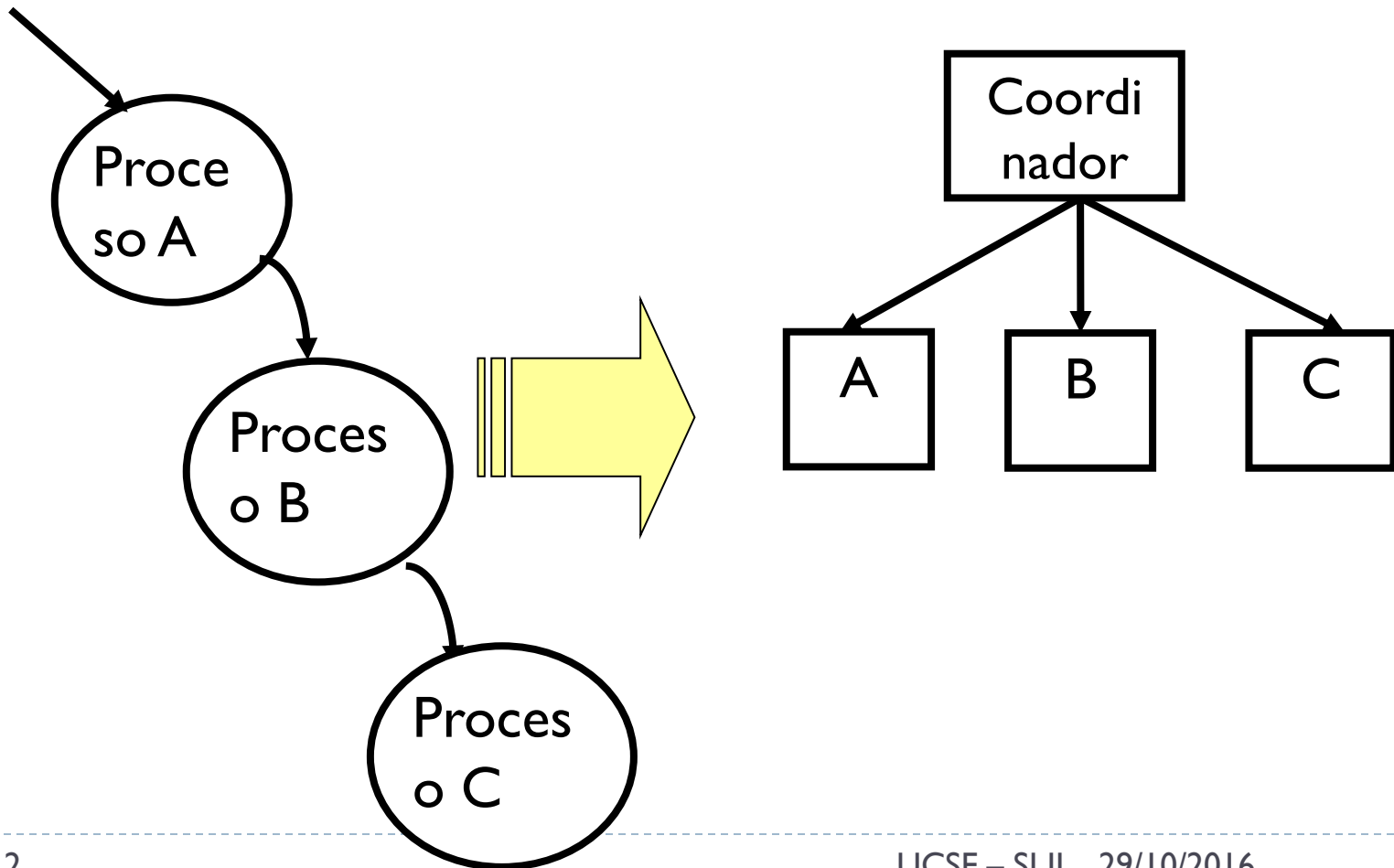
# Diagramas Estructurados

- La secuencia de ejecución de los módulos: de arriba hacia abajo y de izquierda a derecha.
- Los DE no muestran ni iteración ni selección por defecto, pero pueden incorporarse simbología para estos casos.



# DFD a DE

---



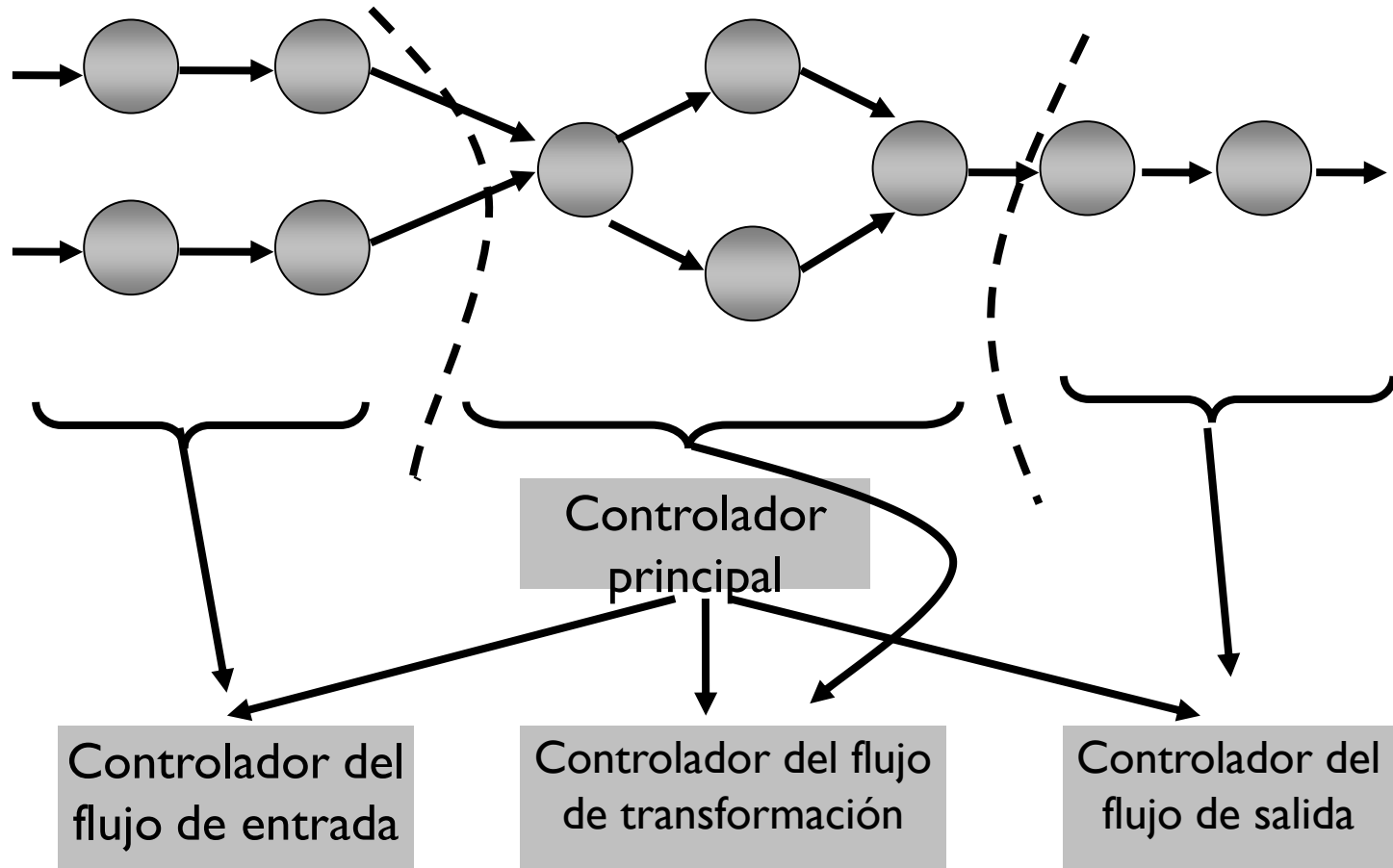
# Derivación de DFD a DE

---

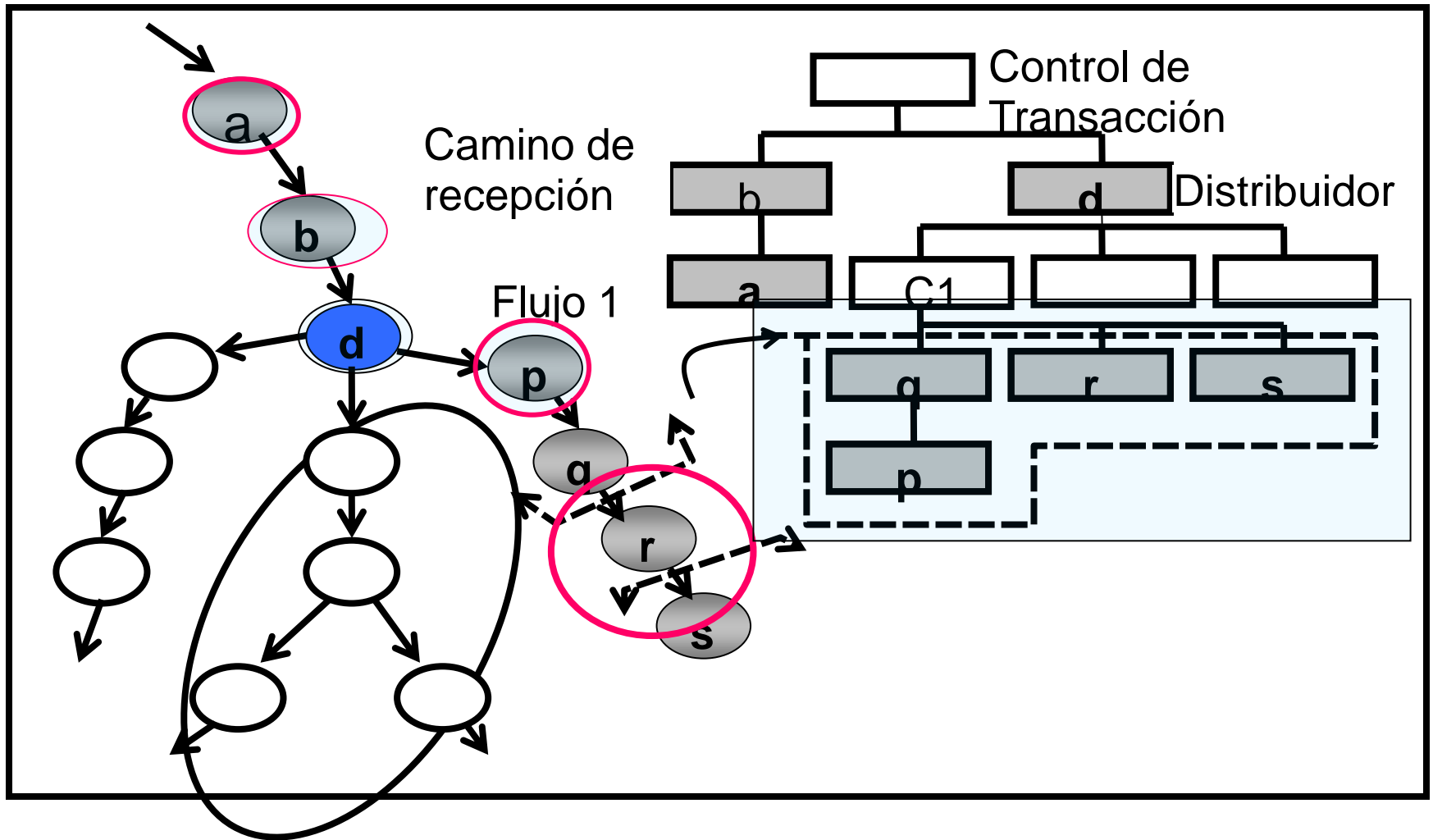
- **Proceso**

1. Establecer el tipo de flujo de información;
2. Determinar los límites del flujo;
3. Convertir el DFD en la estructura del programa;
4. Definir la jerarquía de control mediante factorización;
5. Refinar la estructura resultante usando medidas y heurísticas de diseño (acoplamiento y cohesión).

# Transformación

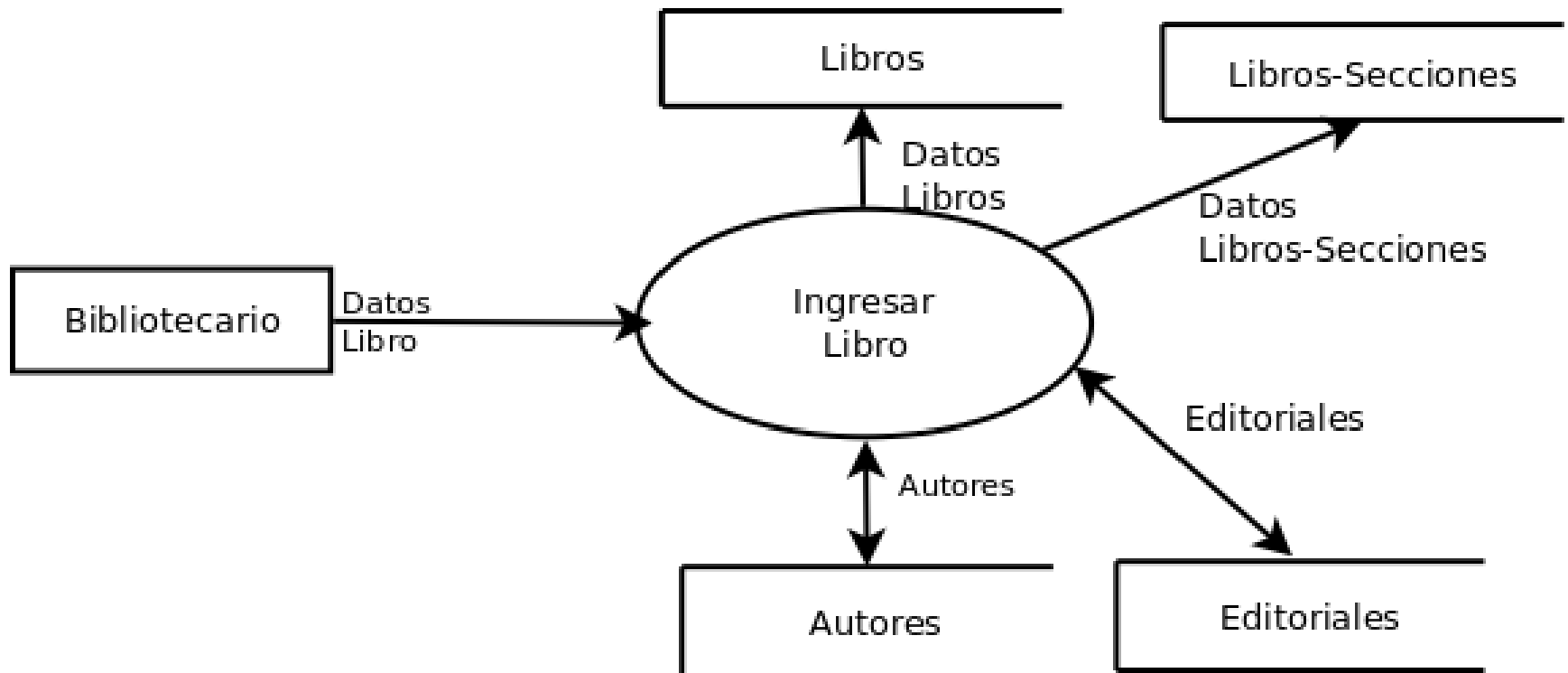


# Transacción



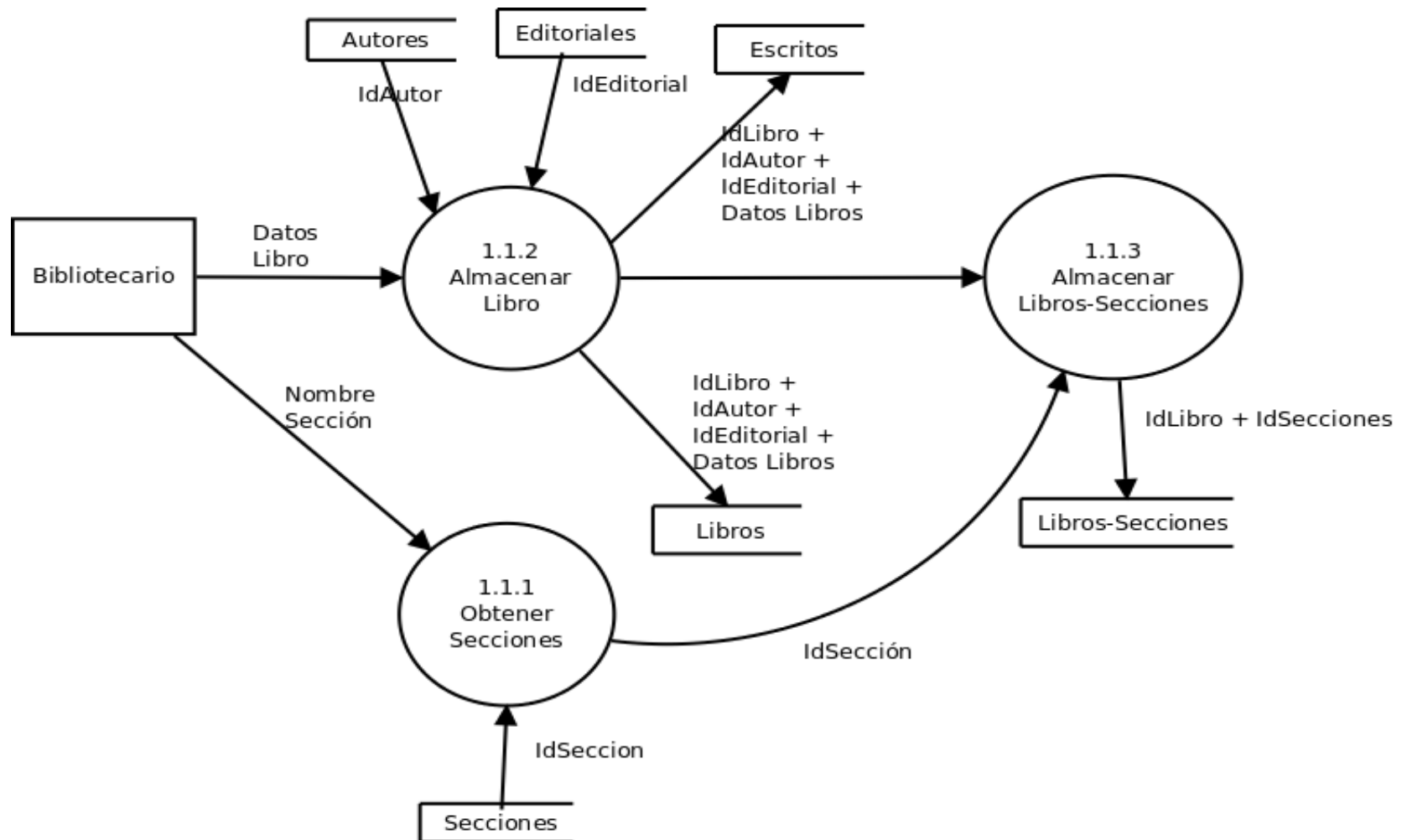
# Ejemplo

---

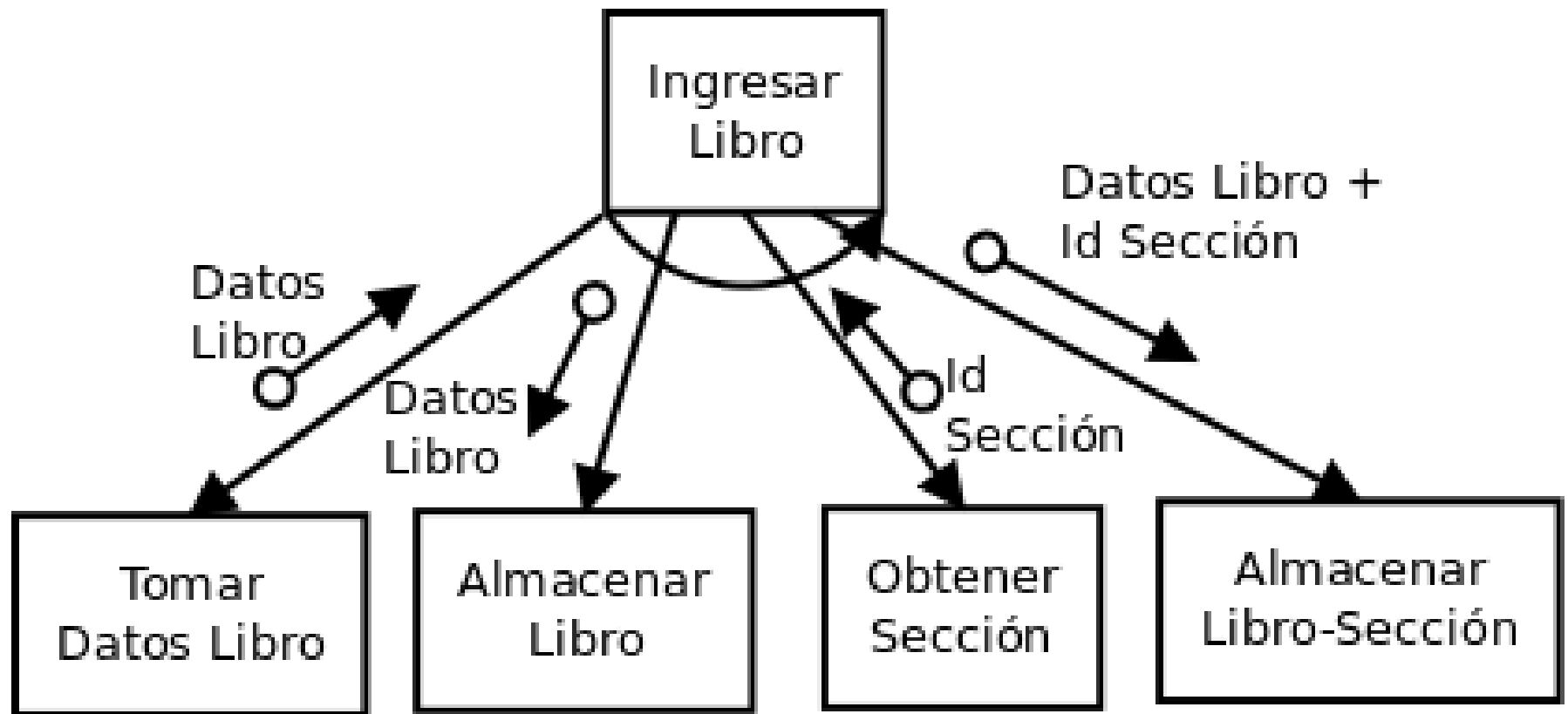




# Ejemplo



# Ejemplo



# Diagramas Estructurados

# Diagramas Estructurados

---

## ► Elementos Básicos

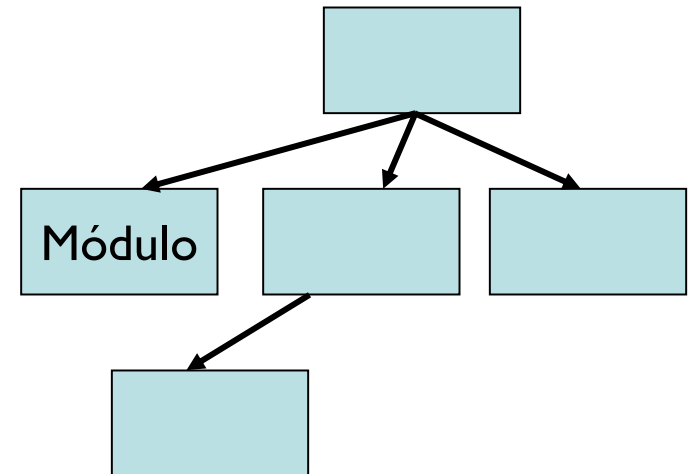
### ► Módulo

## ► Llamadas intercambiando

### ► Datos



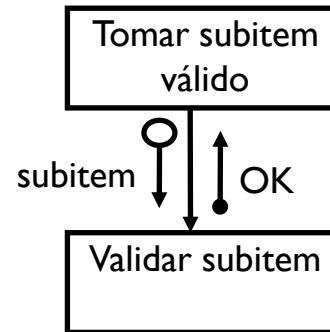
### ► Señales



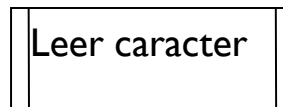
# Diagramas Estructurados

---

- Transferencia de datos



- Extensiones
  - Módulos librería



# Diagramas Estructurados

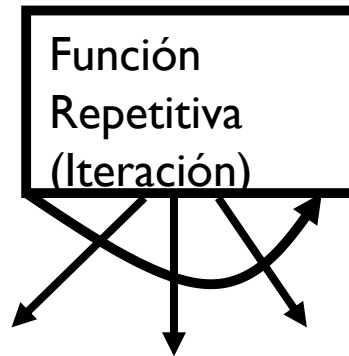
---

- Secuencia, selección e iteración
  - *Secuencia*: orden en el cual los bloques son ejecutados.
  - *Selección*: uso de condiciones para controlar si o no un módulo es ejecutado o cuál de varios bloques será ejecutado.
  - *Iteración*: control de ciclos.

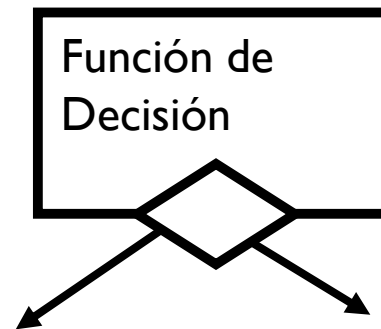
# Diagramas Estructurados

---

- **Iteración**

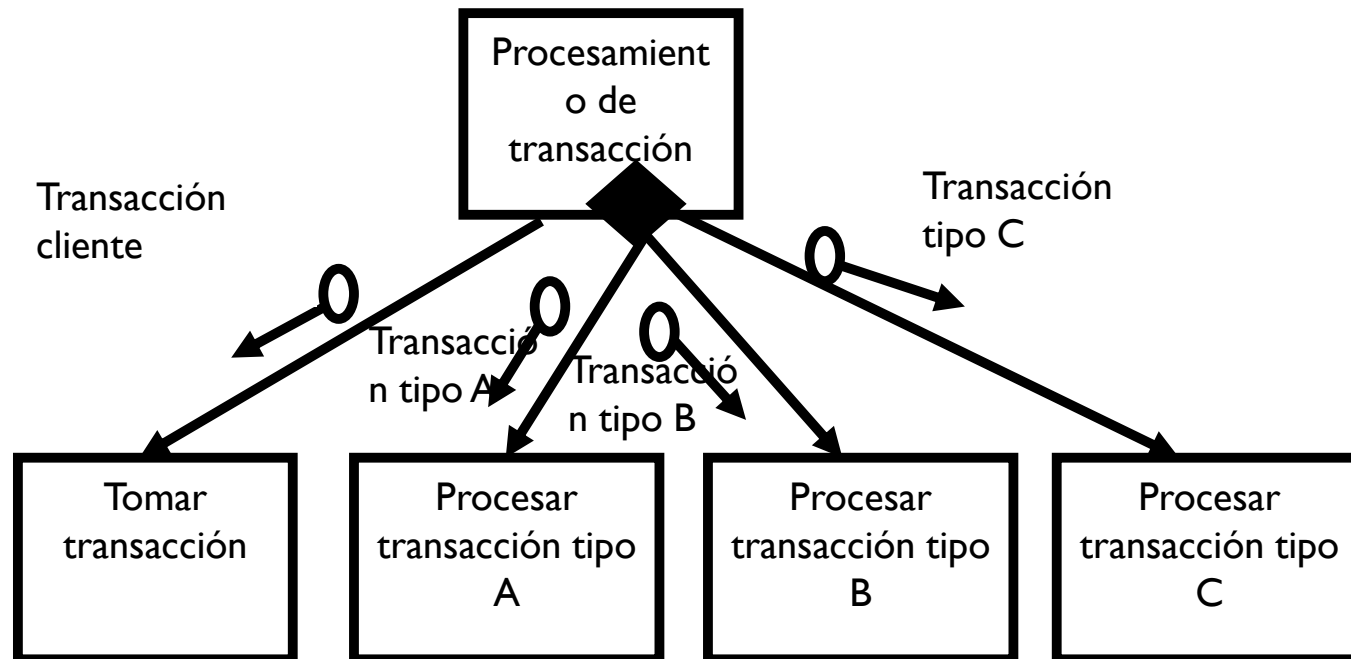


- **Selección / Decisión:**



# Diagramas Estructurados

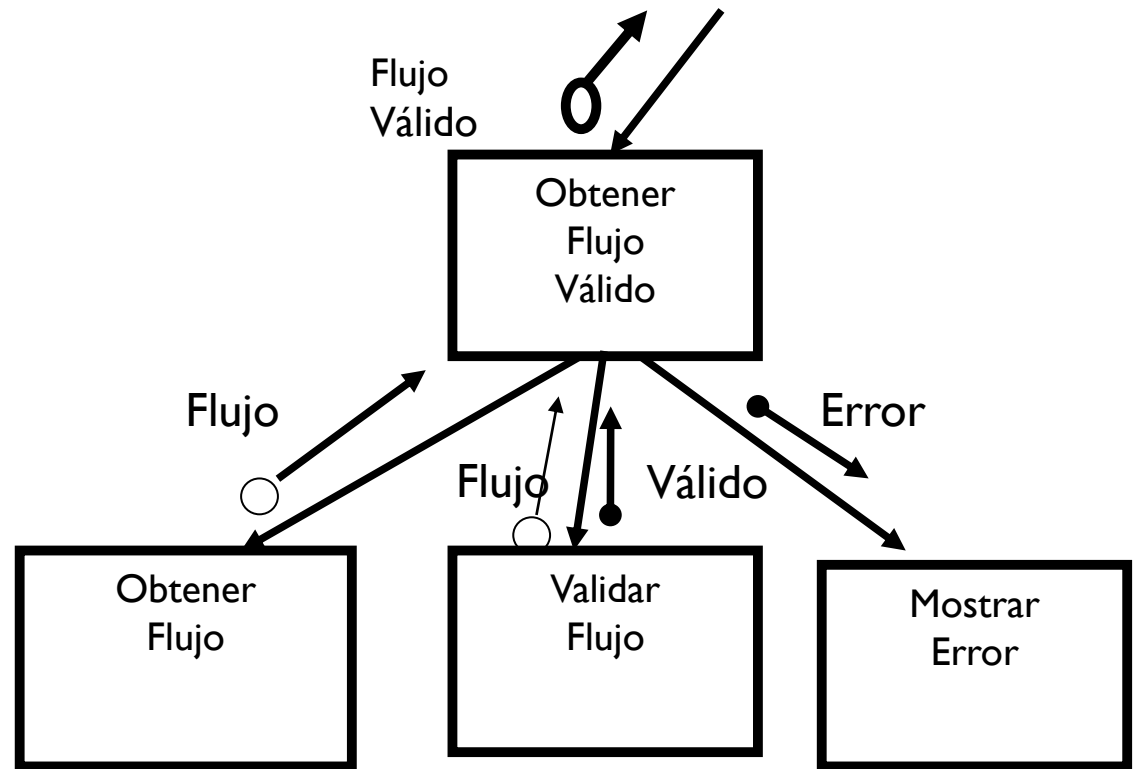
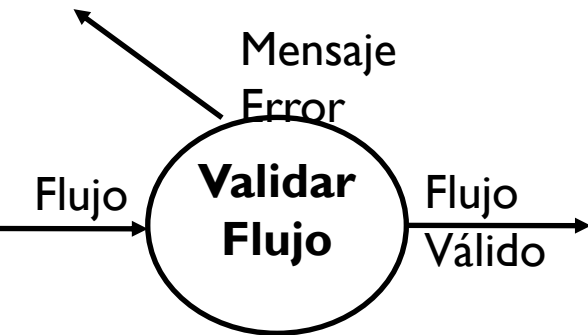
## ► Centro de Transacción





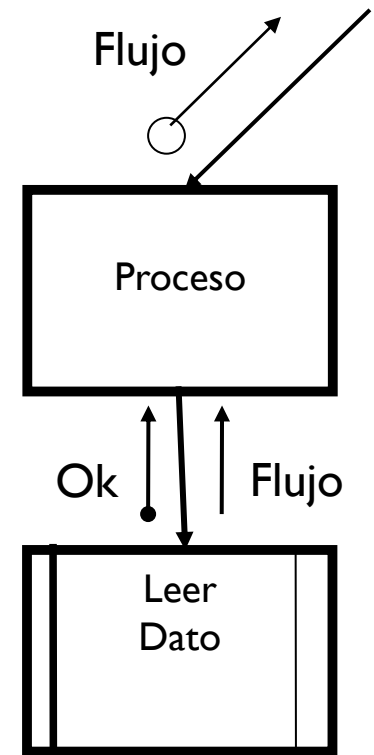
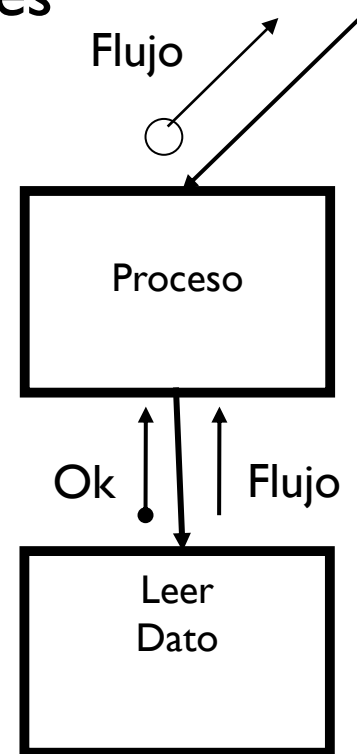
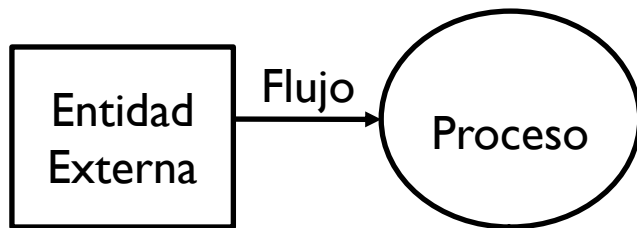
# Diagramas Estructurados

- Validación



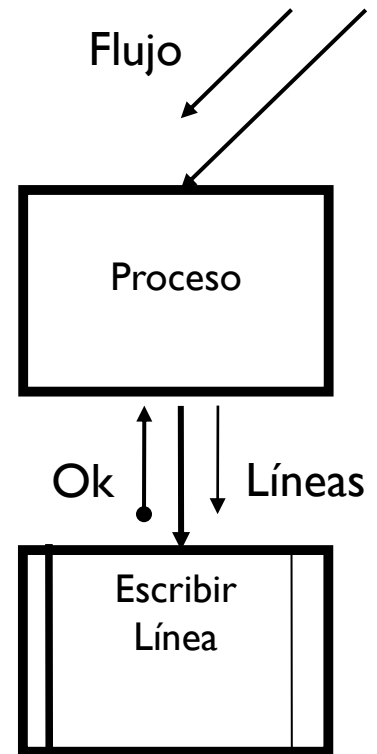
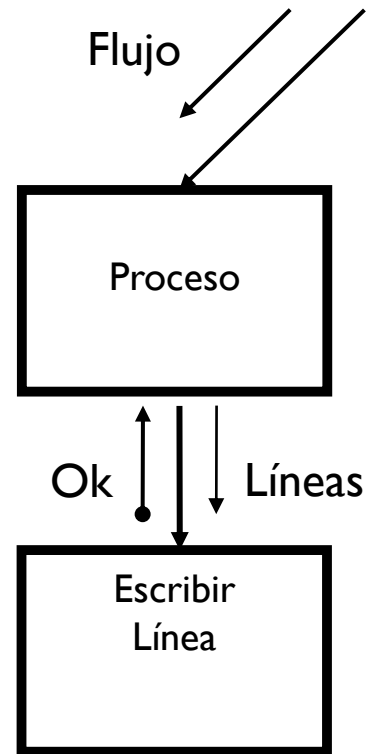
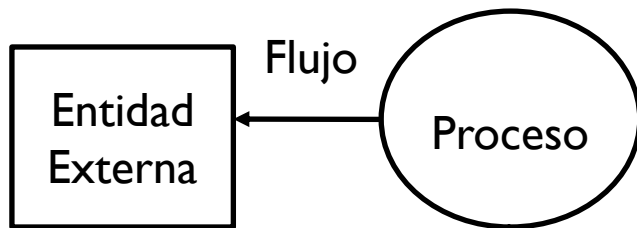
# Diagramas Estructurados

## ► Lectura de datos desde entidades



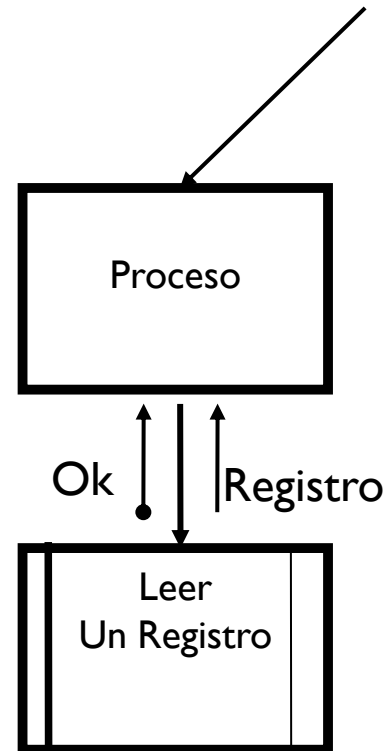
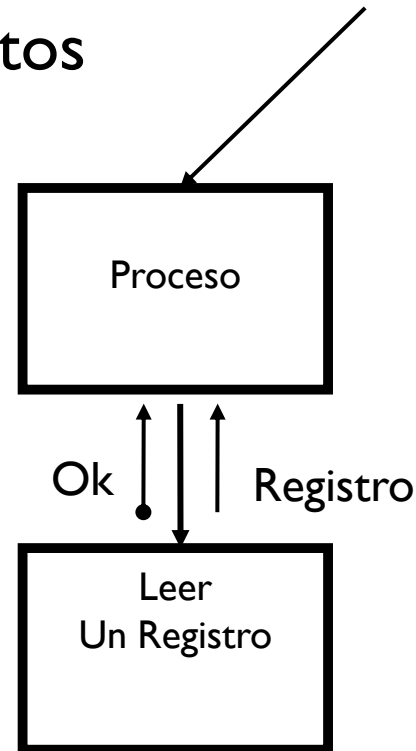
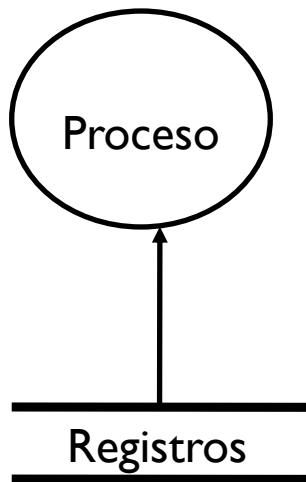
# Diagramas Estructurados

## ► Emisión de datos a Entidades Externas



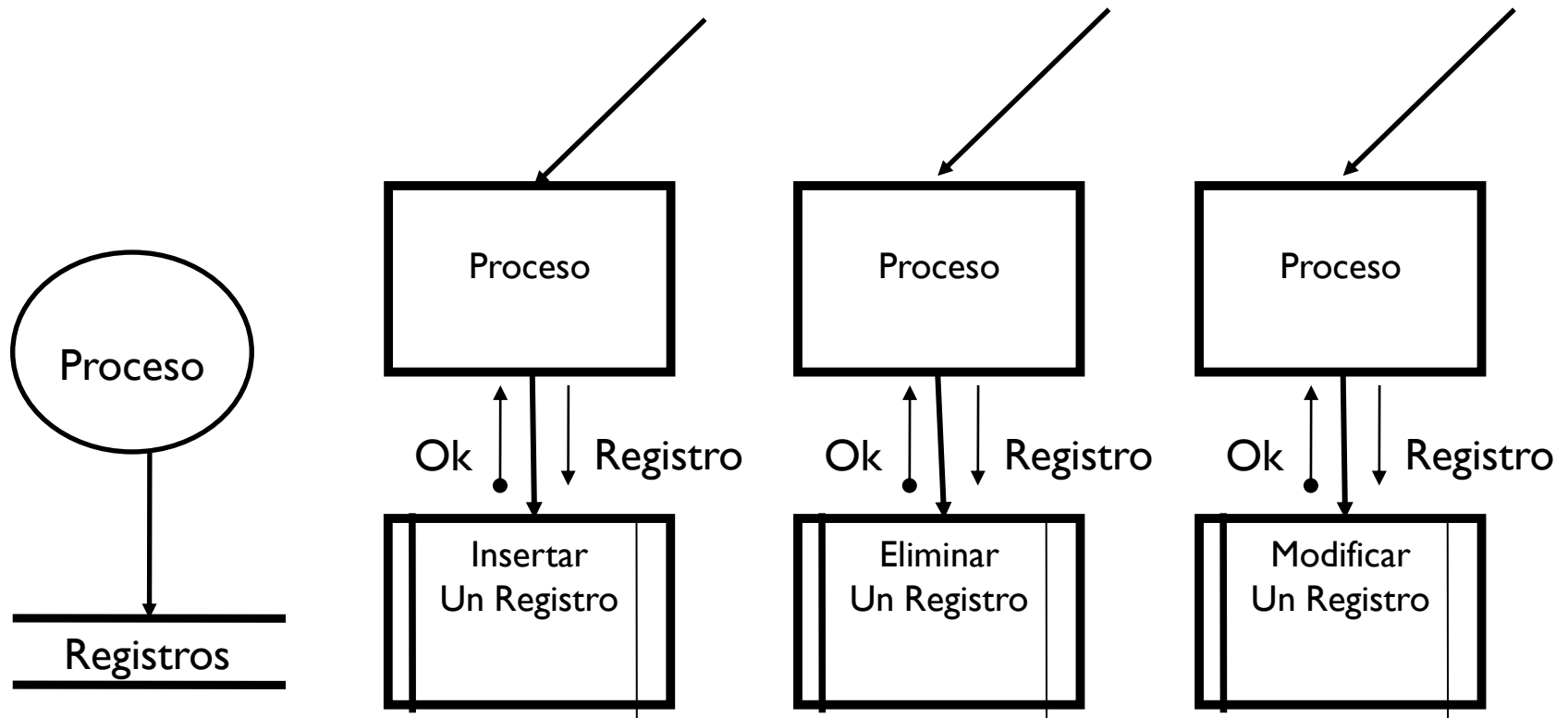
# Diagramas Estructurados

## ► Lectura desde almacenes de datos



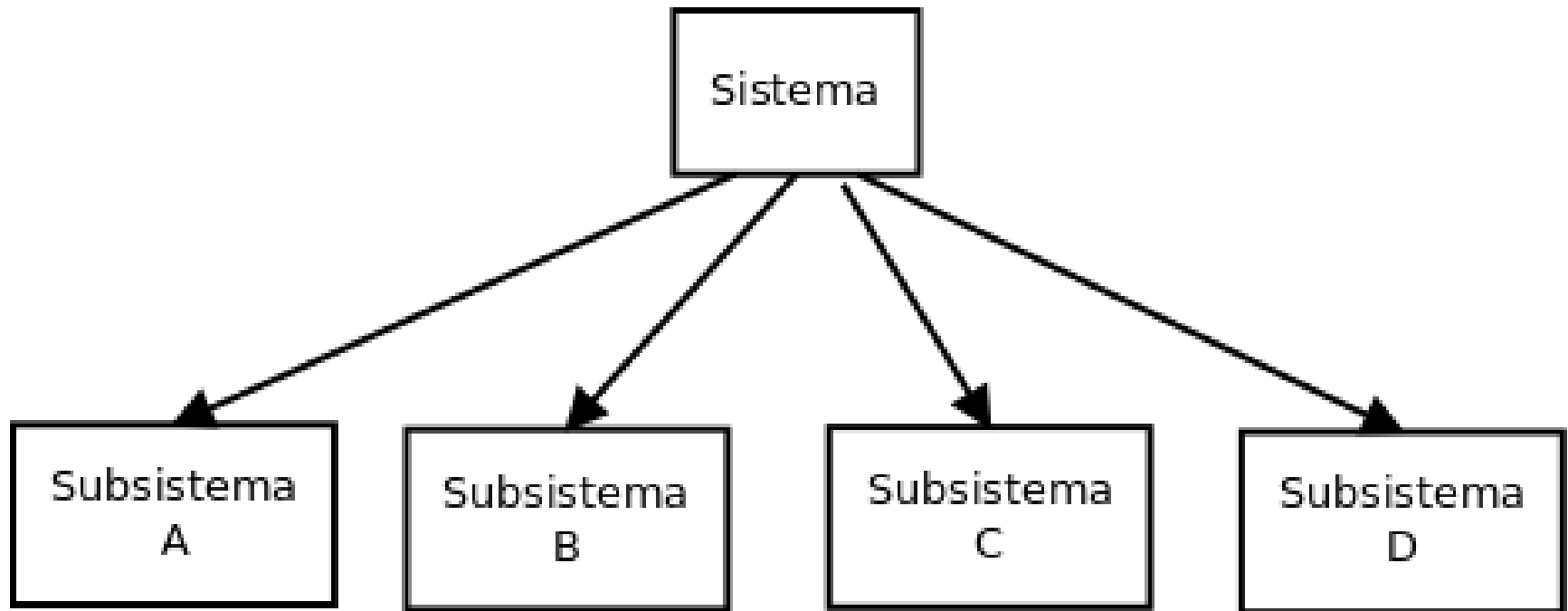
# Diagramas Estructurados

## ► Escritura en almacenes de datos



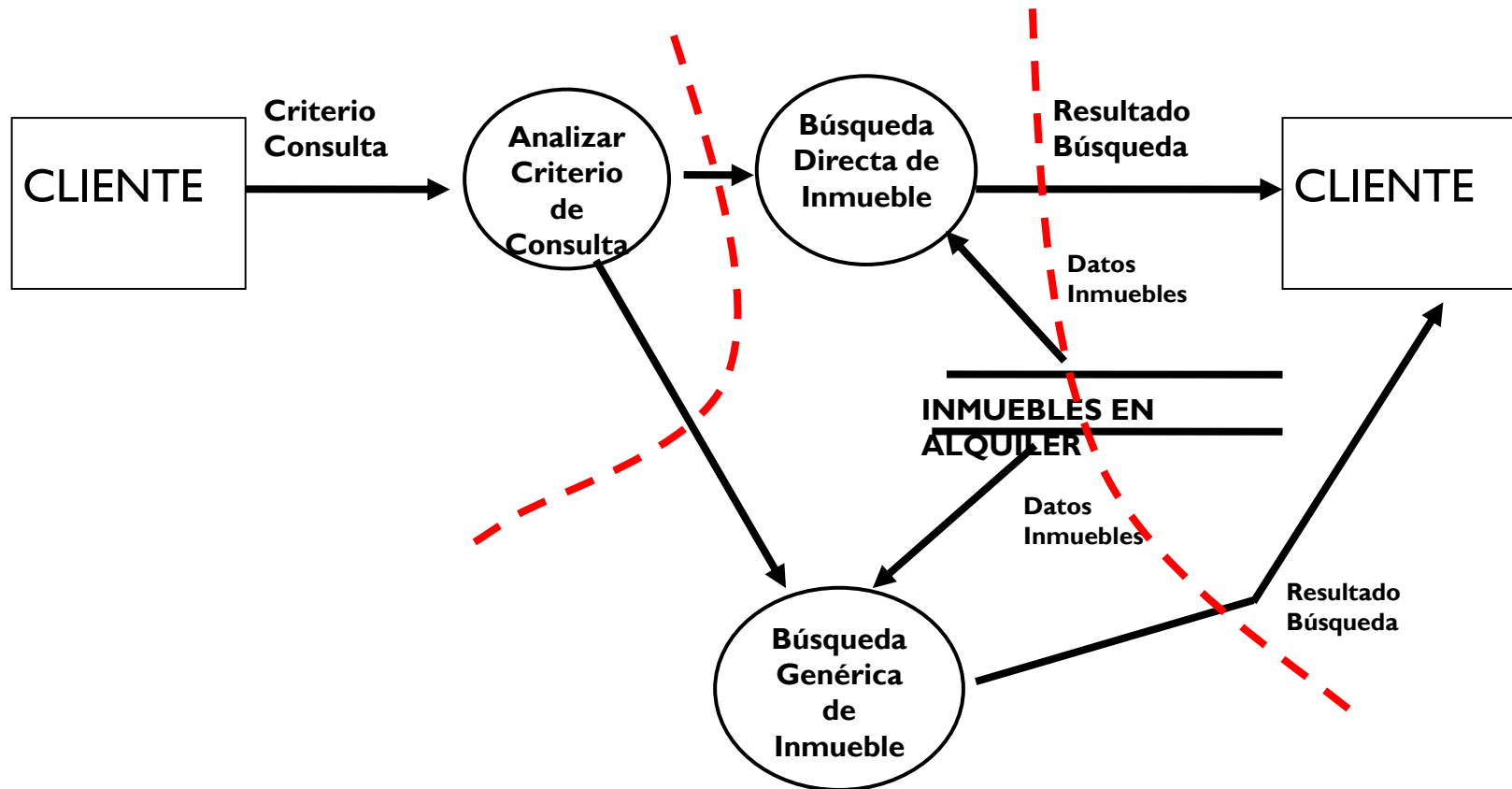
# DE – Caso Especial

---



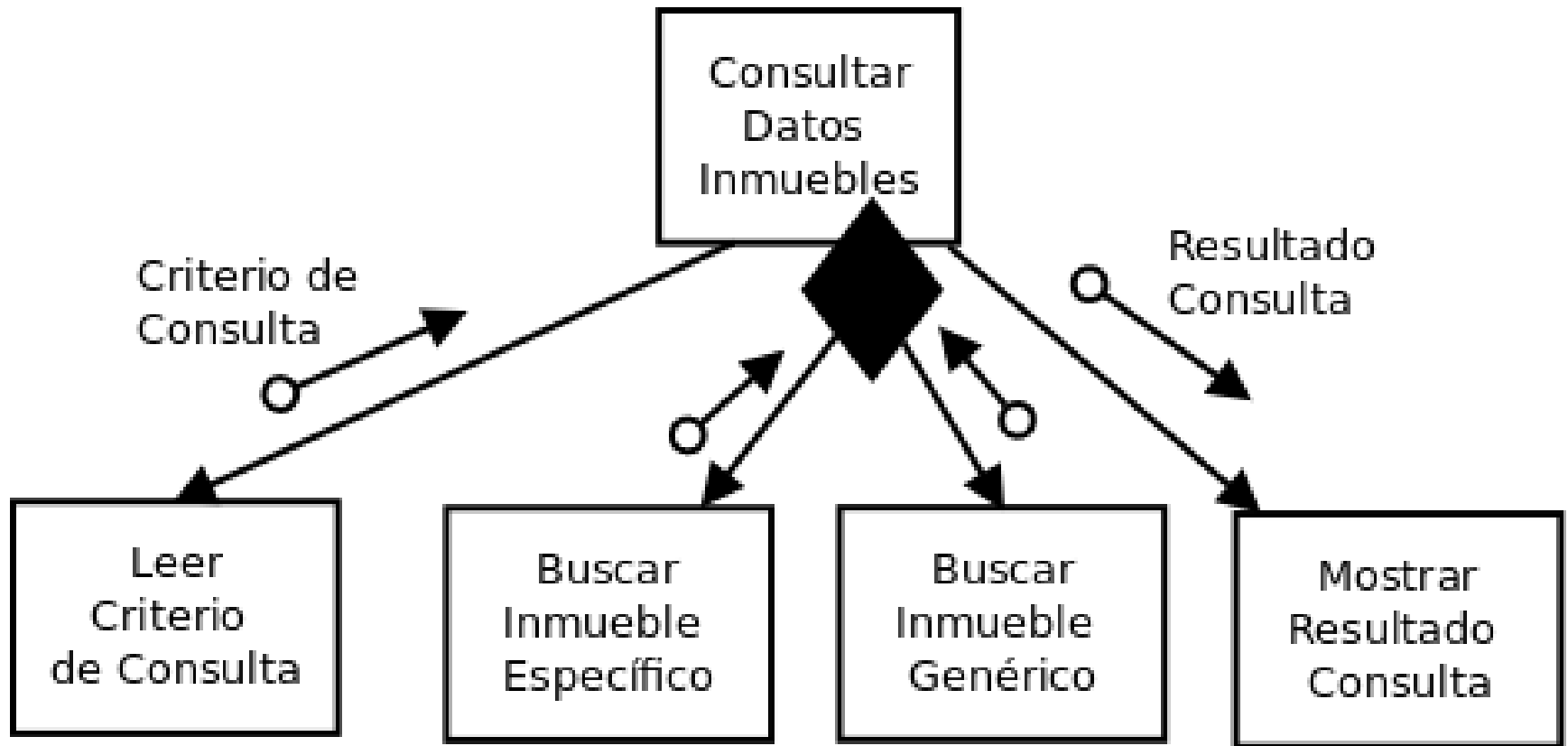
# Ejercicio

“Un cliente realiza una consulta sobre datos de inmuebles.”



# Diagramas Estructurados

---





# Diagramas Estructurados

---

- Ejemplo

## *P1 – Ingresar Datos Socio*

Descripción: transcribir los datos personales del socio desde la solicitud al sistema; eventualmente realizar alguna modificación.

INICIO

Tomar datos Socio

Verificar que los datos sean correctos

Si Nuevo Socio

Obtener el último Nro Socio

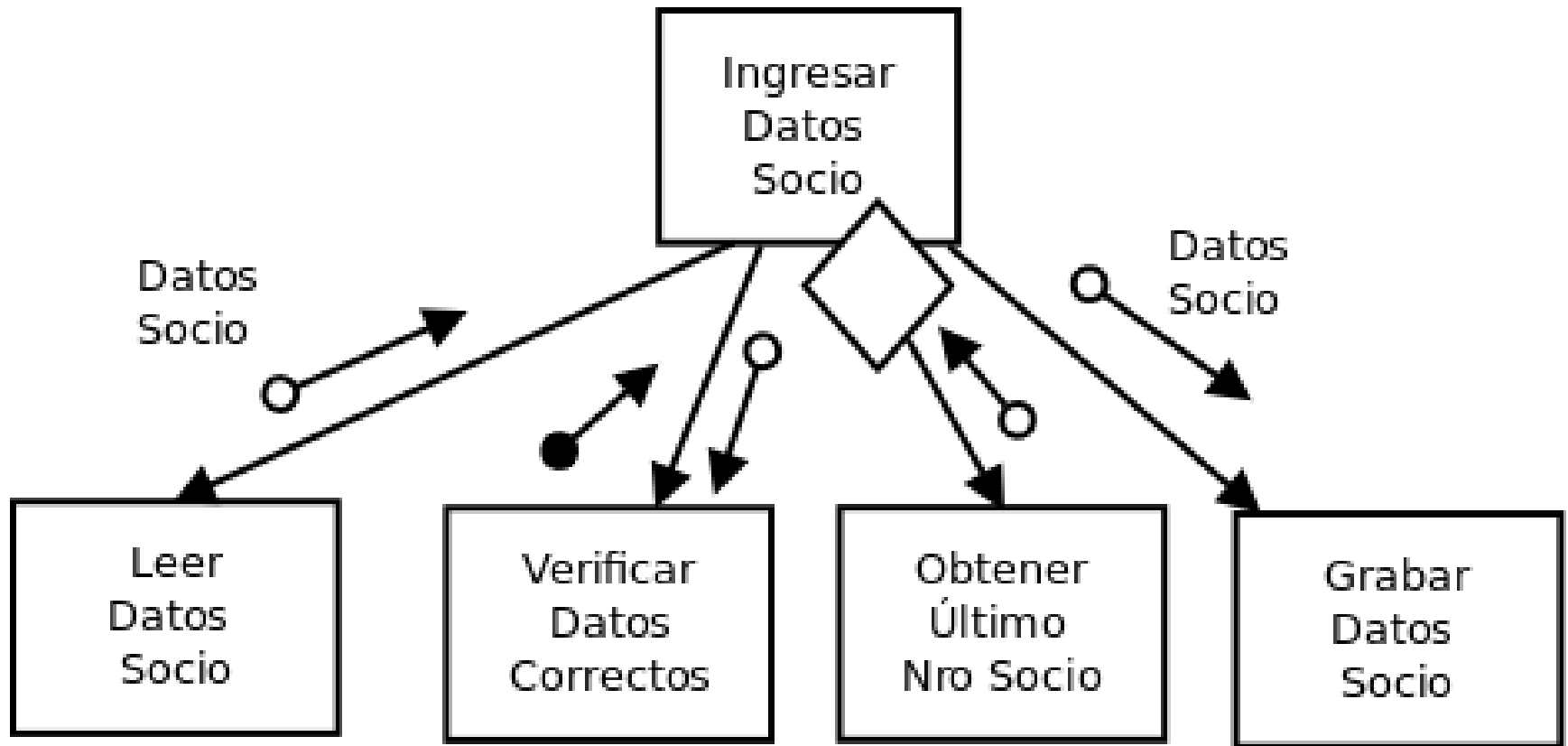
FIN SI

Grabar los datos

FIN

# Diagramas Estructurados

---



# Diagramas Estructurados

---

## *P3 – Generar Deuda*

Descripción: cada fin de mes generar la deuda del mes siguiente para todos los socios. Chequear el mes anterior y si hay deuda impaga del socio cambiar el estado del socio a moroso. Por simplicidad se toma que el importe a cobrar a los socios por los servicios que utilizan del club es la suma del importe de cada uno de los servicios que utiliza.

INICIO

Leer importe de los servicios

REPETIR

Leer servicios del Socio

Leer deudas del mes anterior

SI deuda impaga

Cambiar estado del socio a 'Moroso'

FIN SI

Leer Ultimo Nro Recibo

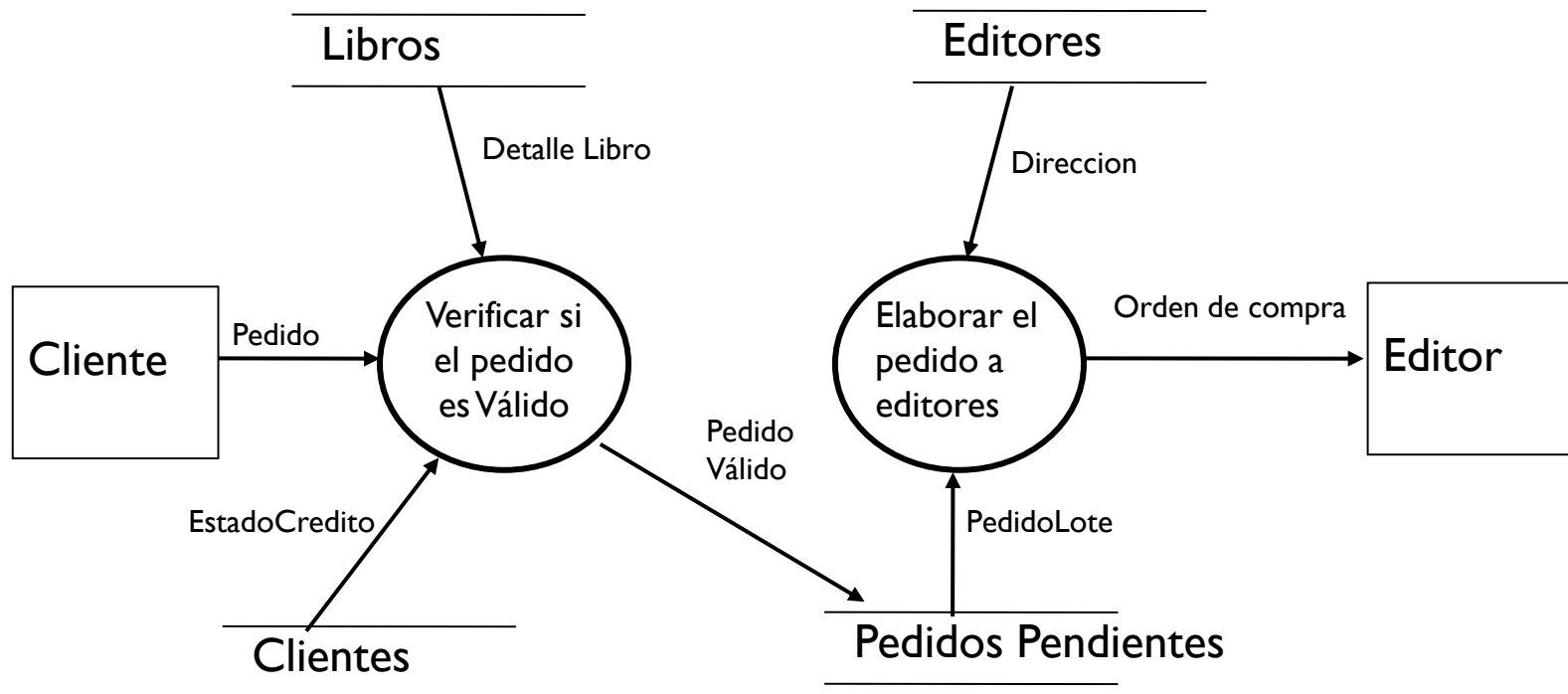
Guardar la nueva deuda

FIN REPETIR

FIN

# Diagramas Estructurados

- Un cliente pide un nuevo libro
  - Si el libro no está disponible se pide al editor



---

## ► Referencias

- “Análisis Estructurado Moderno”, Edward Yourdon, Prentice Hall, 1989
- “Ingeniería de Software, Un enfoque Práctico”, 7ma Ed., Roger Pressman, 2010
  - Cap. 9.6 MAPEO DE LA ARQUITECTURA CON EL USO DEL FLUJO DE DATOS
- “Análisis y Diseño de sistemas”, 9na Ed, K. Kendall y J. Kendall, Pearson, 2011