SSD

1) definición
2) procedimiento de diseño
3) análisis trasformacional
4) análisis transaccional
5) heurística de diseño
6) optimización

Método SSD

Structured System Design

Definición:

Es un método de diseño que permite obtener la estructura de un sistema software a partir de un DFD que describa su funcionamiento.

Autores: Stevens, Myers & Costantine

Método SSD

Objetivo:

- Obtener estructuras de diseño que faciliten el mantenimiento

Principios de Diseño:

- acoplamiento y cohesión
- analogía problema-solución

SSD procedimiento de diseño:

1) Revisar el modelo del sistema
2) Establecer el tipo de flujo
3) Delimitar zonas en el DFD
4) Obtener carta de estructura inicial
5) Generar la jerarquía de control mediante factorización.
6) Refinar la estructura resultante

Sistemas trasformacionales

Patrón de sistema:

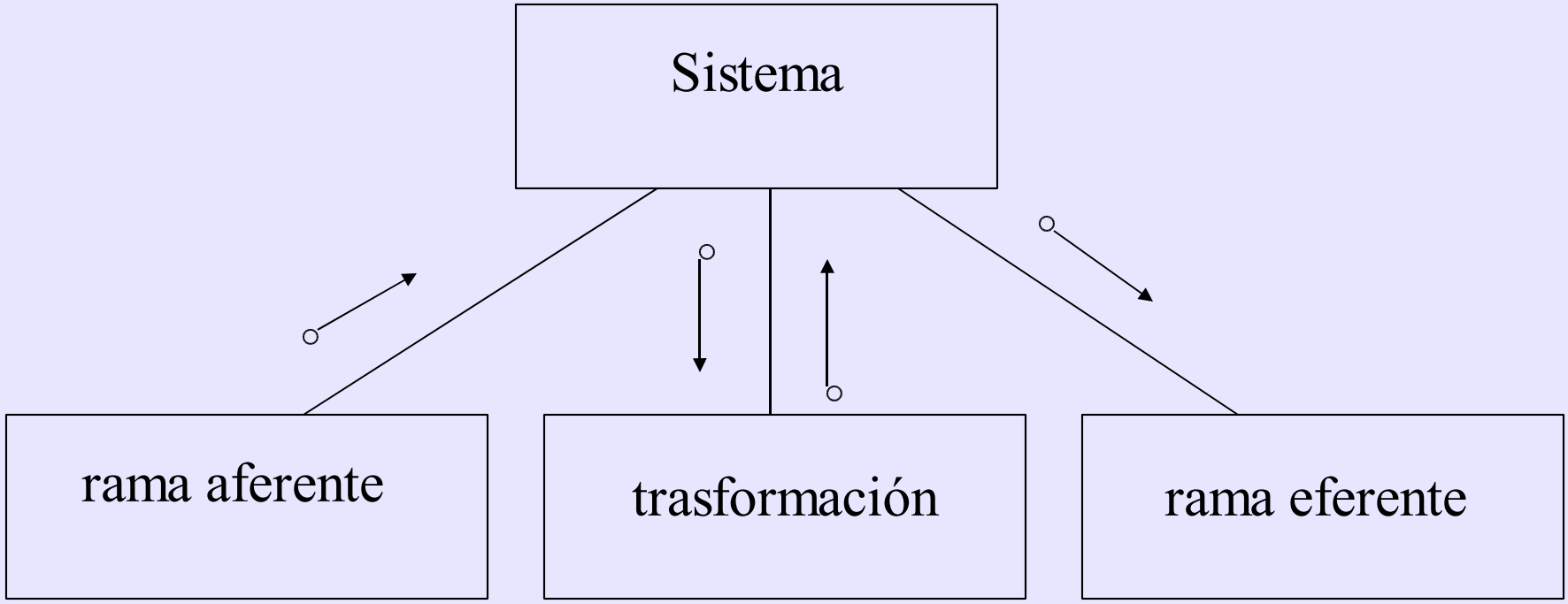
Sistemas trasformacionales

Zonas en el diagrama:

- aferente
- centro de trasformación
- eferente

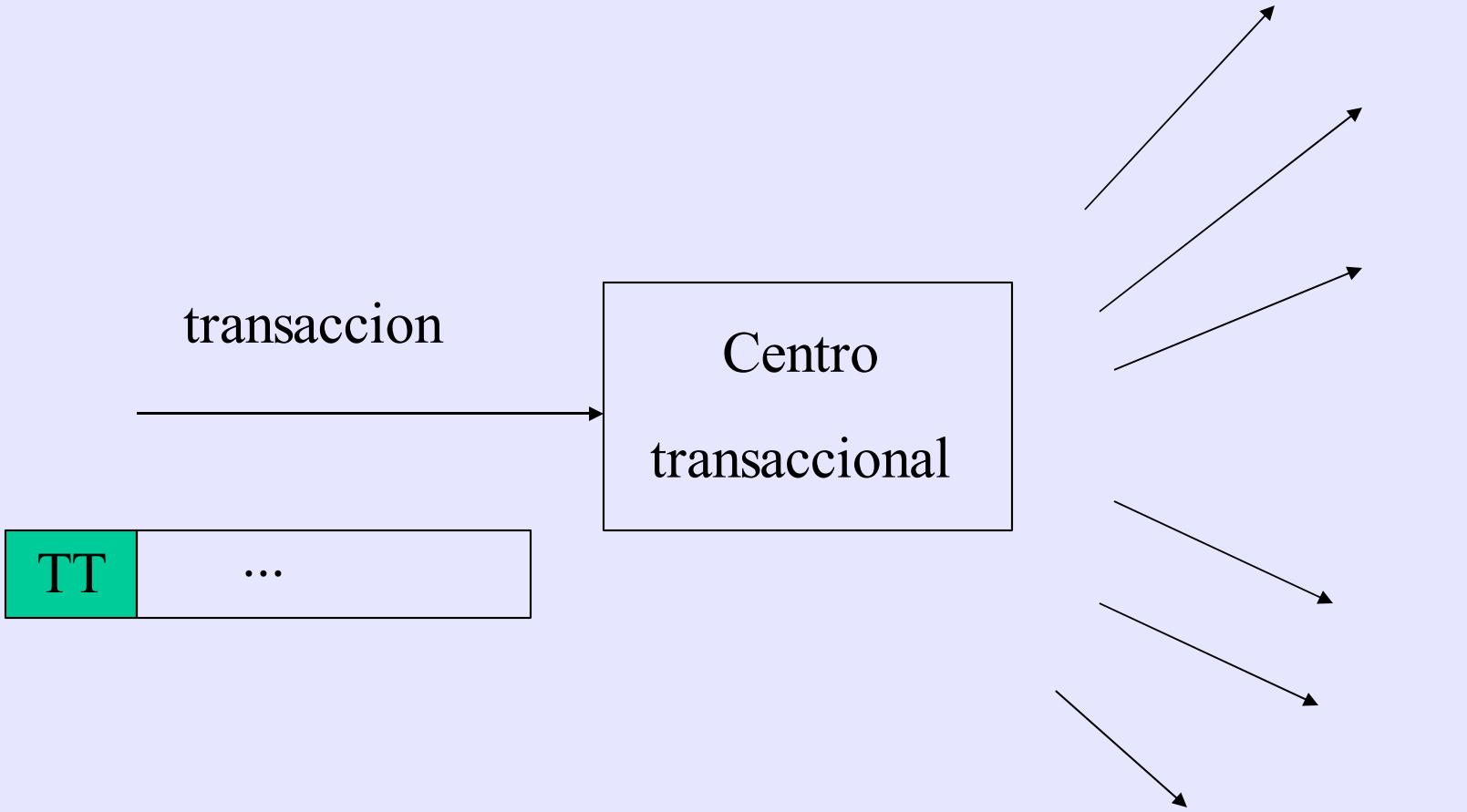
Sistemas trasformacionales

Carta de estructura inicial:



Sistemas transaccionales

Patrón de sistema:



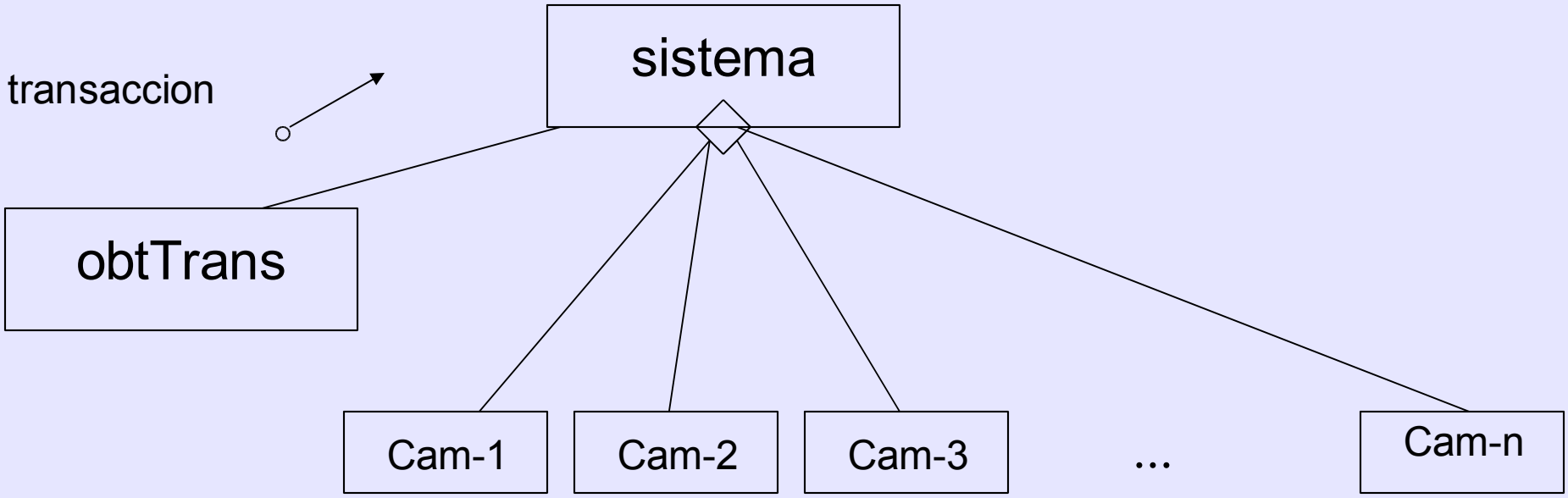
Sistemas transaccionales

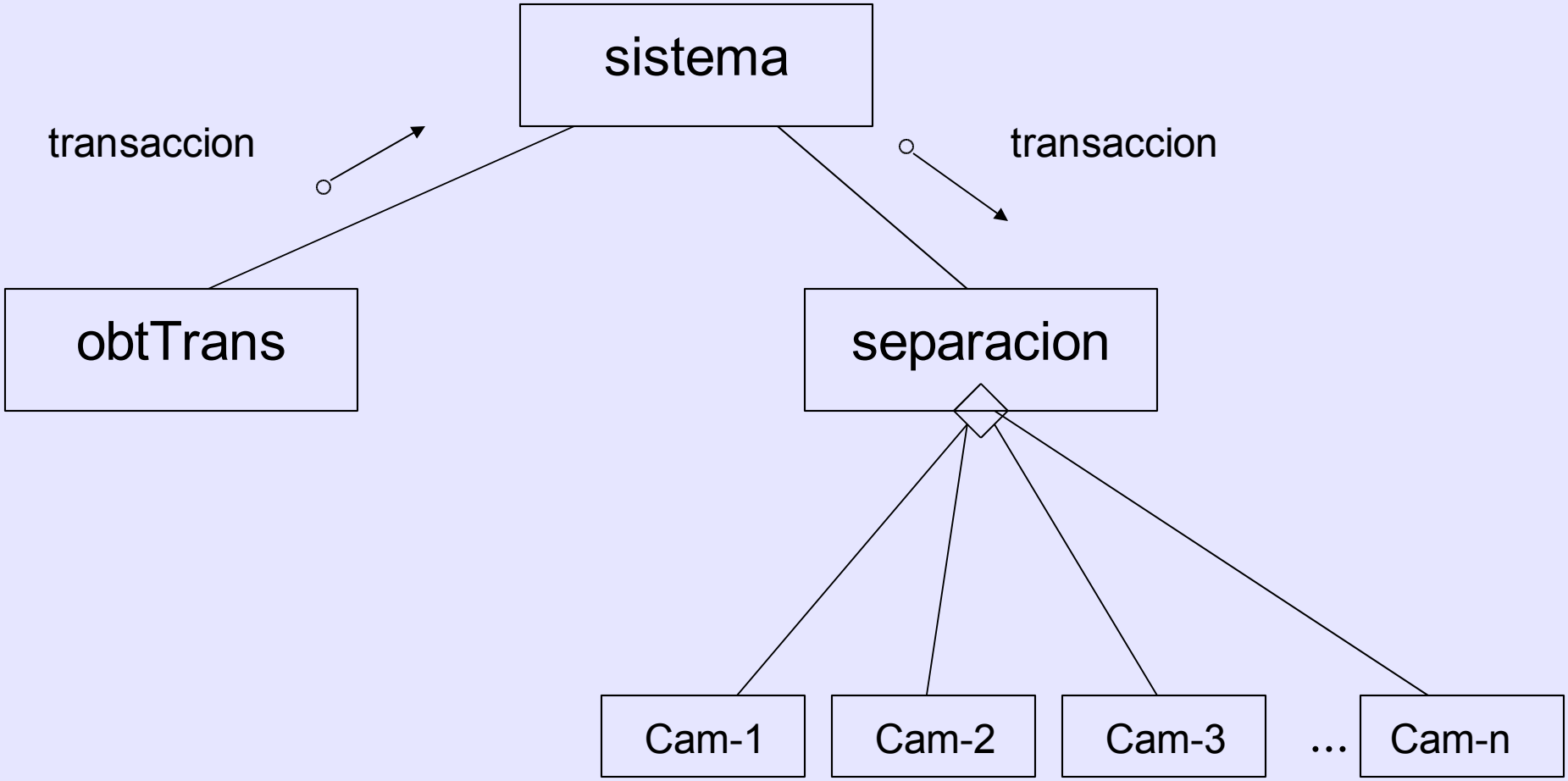
Zonas en el diagrama:

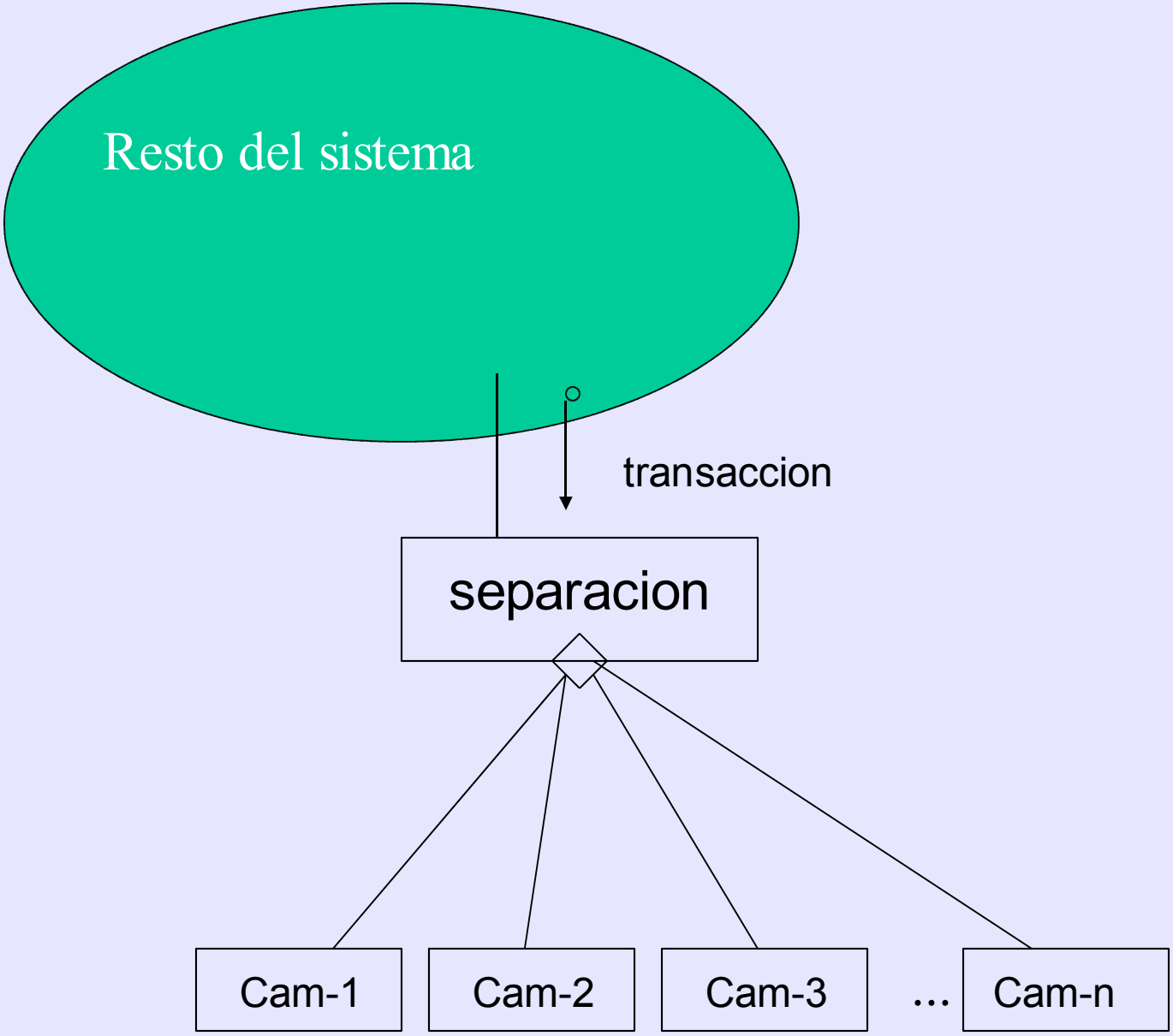
- acceso transacción
- centro transaccional
- caminos de acción

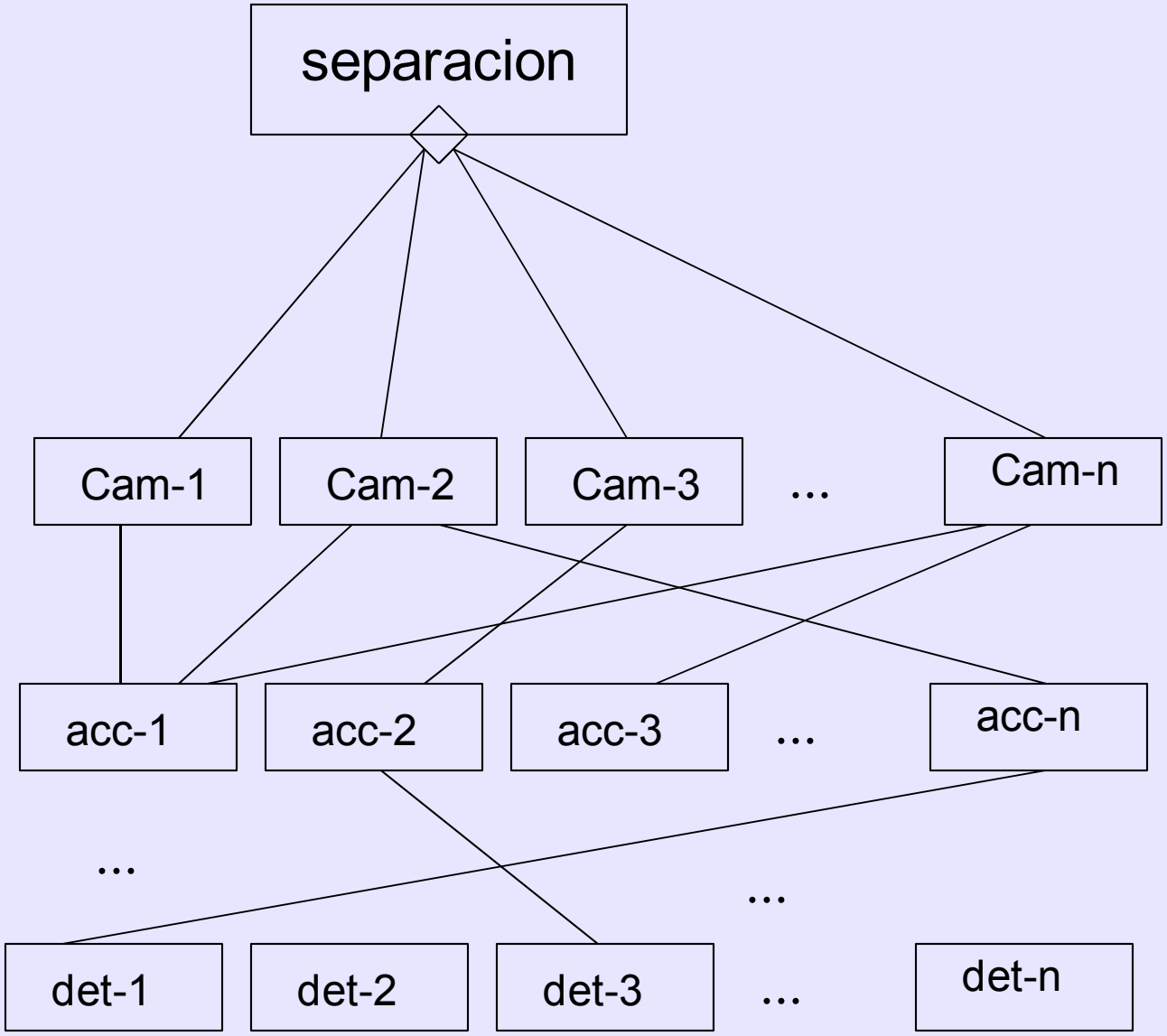
Sistemas transaccionales

Carta de estructura inicial:

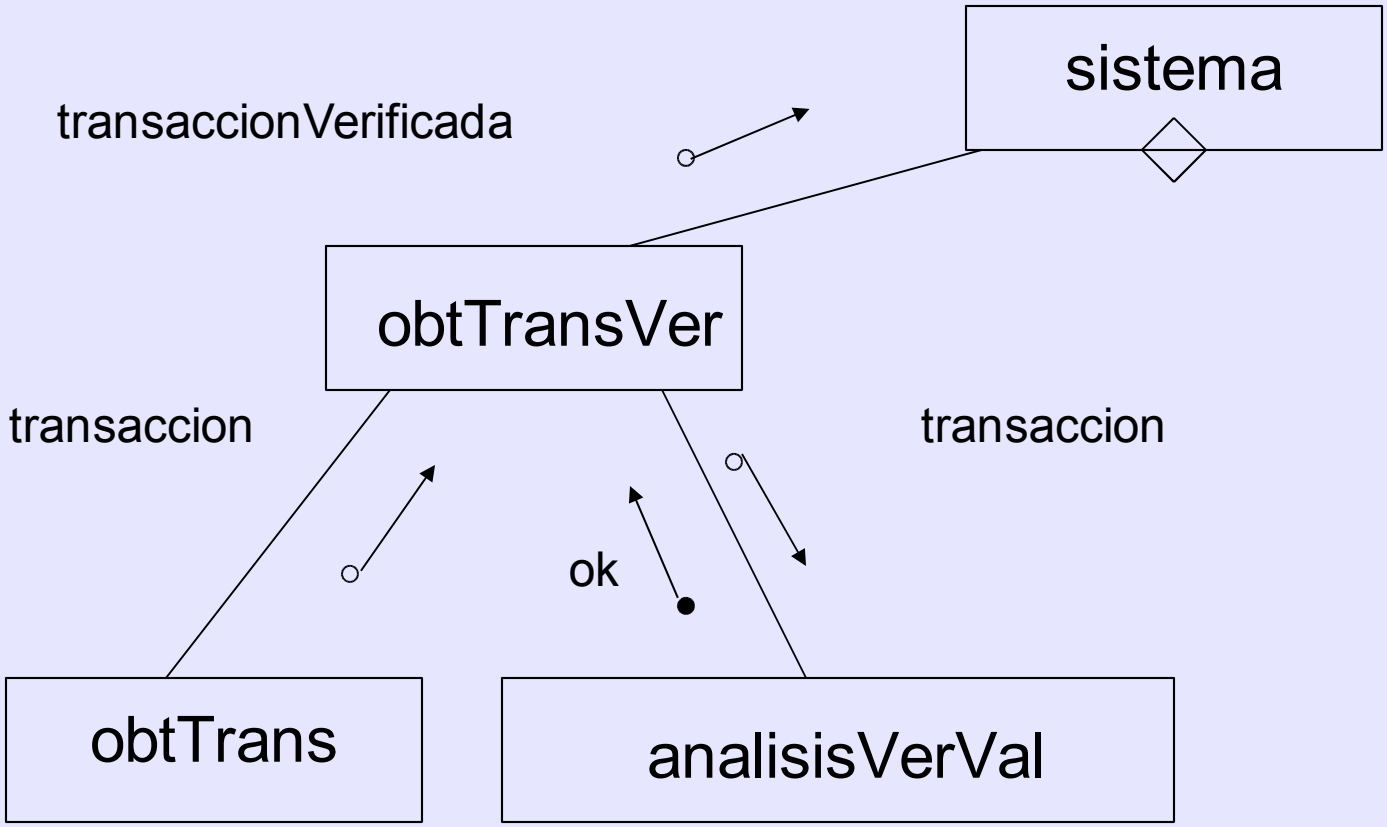



sistema
transaccion
transaccion
obtTrans
separacion
Cam-1
Cam-2
Cam-3
...
Cam-n

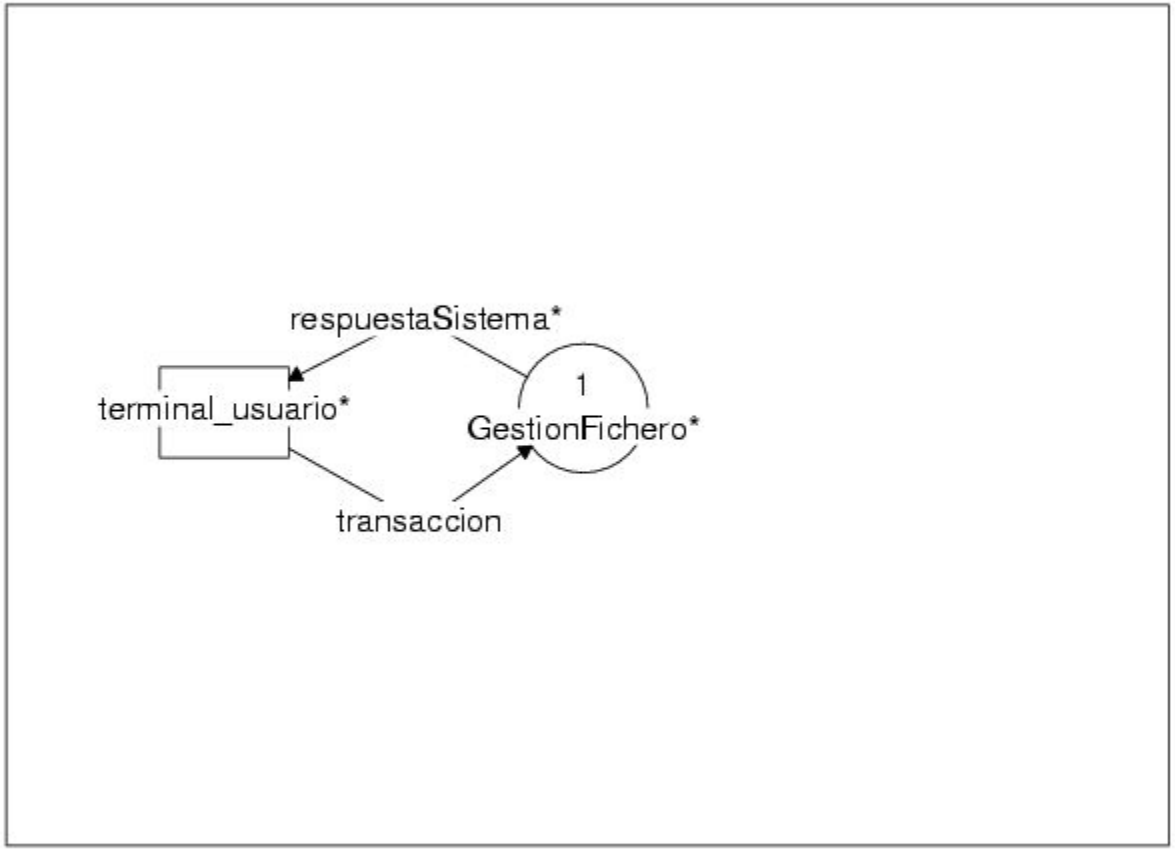

Resto del sistema
transaccion
separacion
Cam-1
Cam-2
Cam-3
...
Cam-n



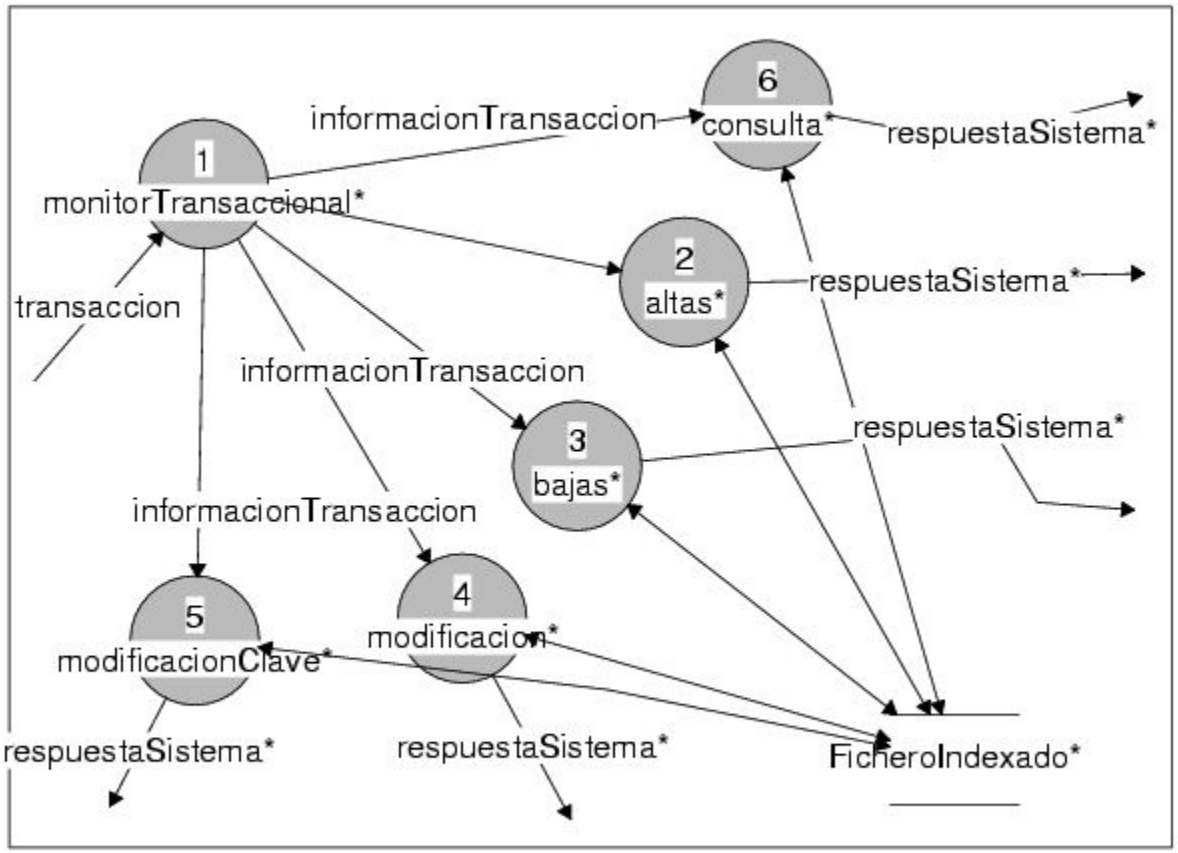
Factorización proceso transacciones



Factorización obtención de la transacción

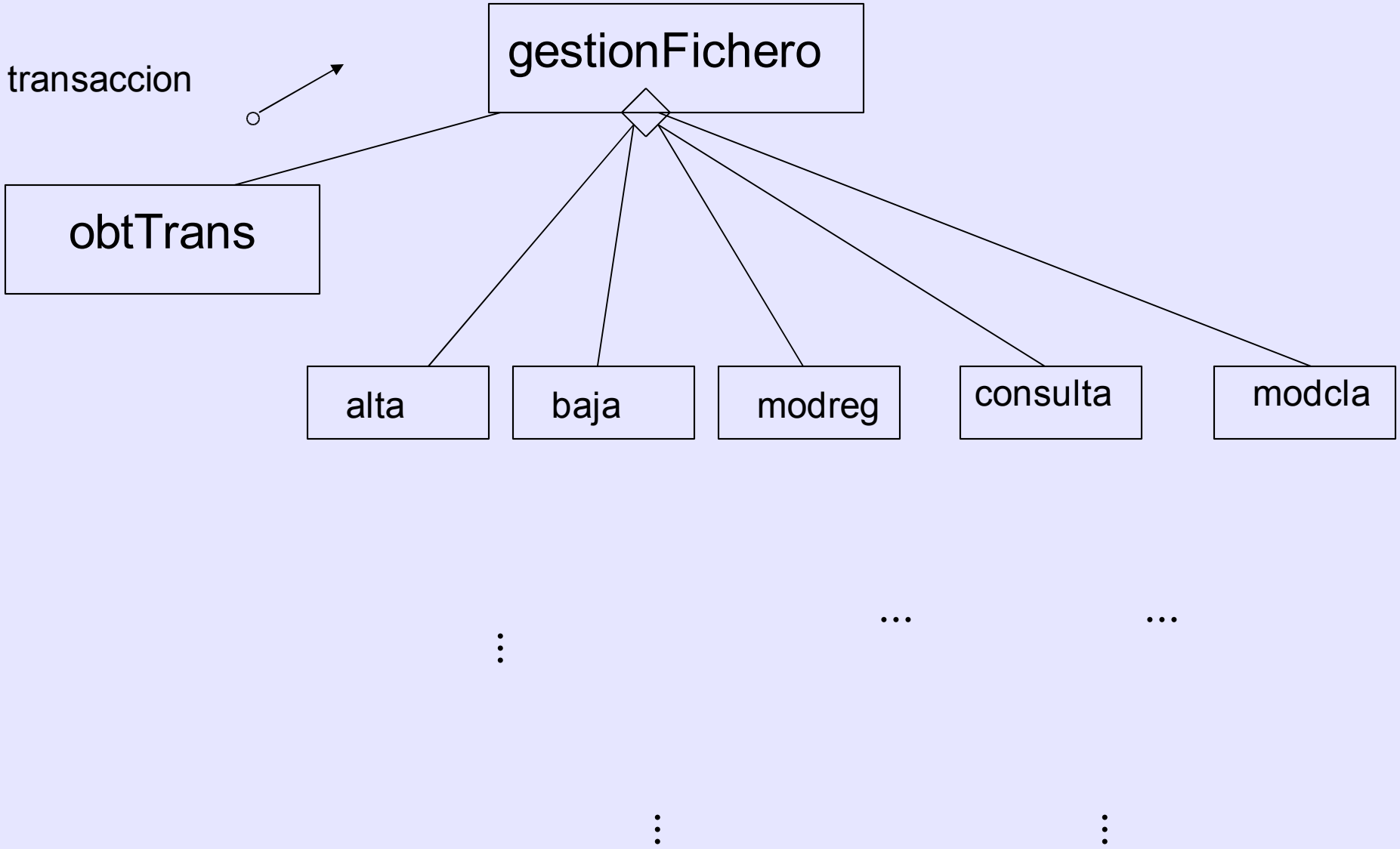

respuestaSistema*
terminal_usuario*
1
GestionFichero*
transaccion

GestionFichero 5-Dec-2002 2:40 PM



tipoTransaccion+informacionTransaccion

```
transaccion
|____tipoTransaccion
|       |____/alta
|       |____/baja
|       |____/mod
|       |____/modcla
|       \____/cons/
\____+informacionTransaccion
        |____[infAlta
        |       |____nuevaClave
        |       \____+cuerpoRegistro
        |____[infBaja
        |       |____claveAntigua
        |       \____+cuerpoRegistro
        |____[infMocla
        |       |____claveAntigua
        |       \____+nuevaClave
        |____[infMod
        |       |____claveAntigua
        |       \____+cuerpoRegistro
        \____[infCons]
                \____ALIAS claveAntigua
```


transaccion
gestionFichero
obtTrans
alta
baja
modreg
consulta
modcla

En los sistemas transaccionales:

- no importa que el módulo de separación tenga una cohesión baja

- la solución no es buena si el proceso de las transacciones
  - requiere información adicional
  - depende del tipo de transacción en el módulo de separación

En los sistemas transaccionales:

- la validación de la transacción no debe incluirse en la zona de separación
- validaciones:
  - material (del formato)
  - formal (del contenido)

- situar el módulo de separación muy alto puede provocar:
  - que éste tome alguna decisión importante de forma poco trasparente
  - que el sistema tenga muy poco control sobre sí

Heurística de diseño:

- reducir acoplamiento y aumentar cohesión
- fan-in, fan-out
- inclusión del alcance de efecto en el de control
- reducir la complejidad de las interfaces

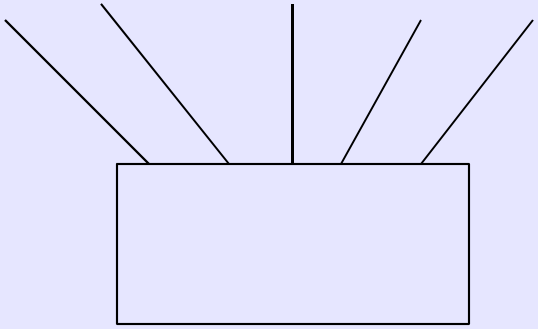
reducir acoplamiento

- mejor mantenerse lejos de la parte alta de la escala:
  - no separando lo que debe estar unido
  - retrasando tiempo de ligadura
- hacer explícitos los acoplamientos patológicos

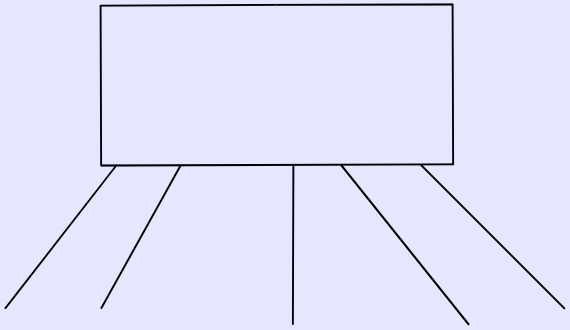
aumentar cohesión

- buscar la cohesión fucional
- puede requerir una granularidad más fina de la descomposición modular

fan-in: *grado de dependencia*



fan-out: *grado de responsabilidad de coordinación*



fan-in, fan-out:

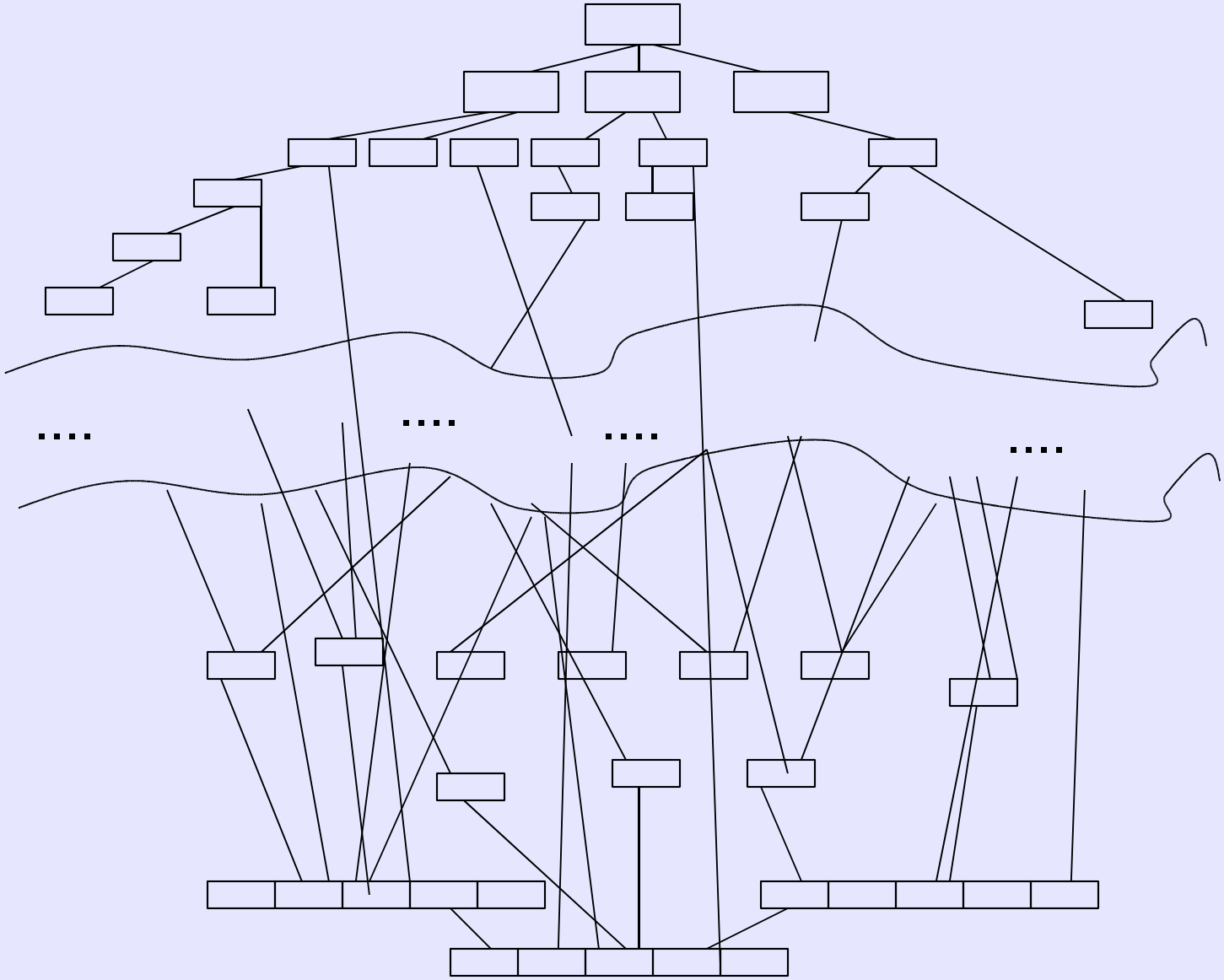
- evitar los fan-out excesivamente altos

  dan lugar a módulos demasiado complejos

- hacer que el fan-in aumente gradualmente al profundizar en la estructura

  estamos reutilizando código

- la estructura terminará teniendo una aspecto ovalado



alcance de efecto de un módulo *m*:

conjunto de módulos *en los que influye m*

alcance de control:

conjunto de módulos que dependen directa ó indirectamente de *m*

consecuencias de la *no* inclusión del alcance de efecto en el de control:

- toma de decisiones replicada
- aumenta el acoplamiento

reducir la complejidad de las interfaces

- usar nombres significativos para módulo y parámetros

  el sentido de las llamadas debe ser claro en su contexto

- evitar un número excesivo de argumentos

- documentar bien las interfaces

Optimización del diseño

- Un mal diseño --un diseño incorrecto-- nunca es óptimo

- A veces los problemas de eficiencia no está donde se supone

Guía de optimización:

- no preocuparse demasiado de los tiempos de ejecución durante el diseño arquitectónico
- refinar durante el diseño detallado los algoritmos de los módulos sospechosos
- codificación de alto nivel, siguiendo un buen estilo
- probar el sistema para identificar los problemas reales de eficiencia
- volver a codificar, o incluso a diseñar, las zonas problemáticas