



Departamento de Informática  
Universidad de Valladolid  
Campus de Segovia

---

# TEMA 9: CRITERIOS DE CALIDAD DE DISEÑO MODULAR

# CRITERIOS DE CALIDAD DE DISEÑO MODULAR

- Conceptos generales
  - Cohesión y acoplamiento
  - Balanceado y factorización

# CRITERIOS DE CALIDAD

- Los criterios son el acoplamiento y la cohesión.
- Estos criterios fueron introducidos por Yourdon et al. En los años 70.
- El acoplamiento es un medio de evaluar la relación entre los distintos módulos de un sistema.
- La cohesión mide el grado de conexión funcional entre los constituyentes o elementos de un mismo módulo.

# ACOPLAMIENTO. DEFINICIÓN

- El acoplamiento es una medida del grado de interdependencia entre los módulos de un sistema.
- Lo deseable es tener módulos con poco acoplamiento (o independiente entre sí).
  - Para ser capaces de realizar el mantenimiento de un módulo sin tener que cambiar otros módulos.
- Este criterio se puede relacionar con la complejidad de una interfaz.

# TIPOS DE ACOPLAMIENTO

- NORMAL

Menos acoplamiento

- POR DATOS

- POR ESTAMPADO

- POR CONTROL

- COMÚN

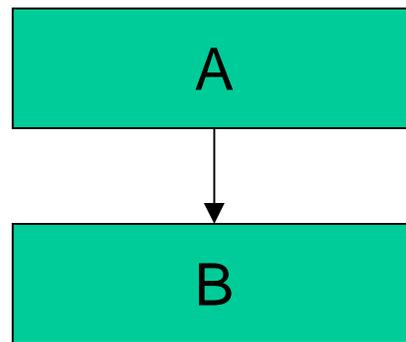
- POR CONTENIDO

Más acoplamiento



# ACOPLAMIENTO NORMAL

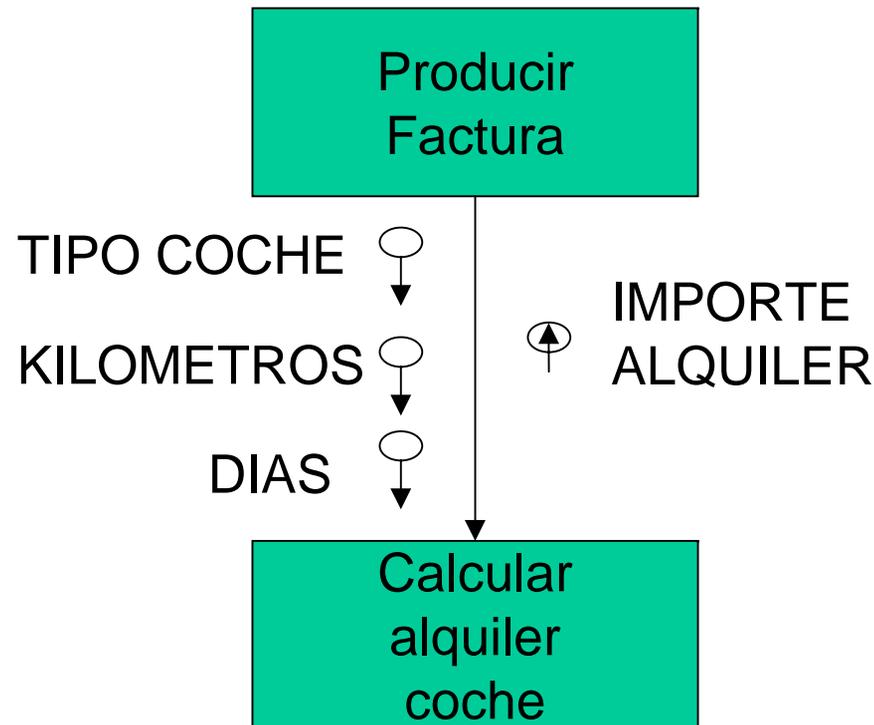
- Se dice que dos módulos “A” y “B” están acoplados normalmente si:
  - A invoca a B
  - B realiza su función y retorna el control a A
  - Toda la información que comparten la realizan a través de los parámetros presentes en la llamada.



No hay parámetros en la llamada.

# ACOPLAMIENTO NORMAL. POR DATOS

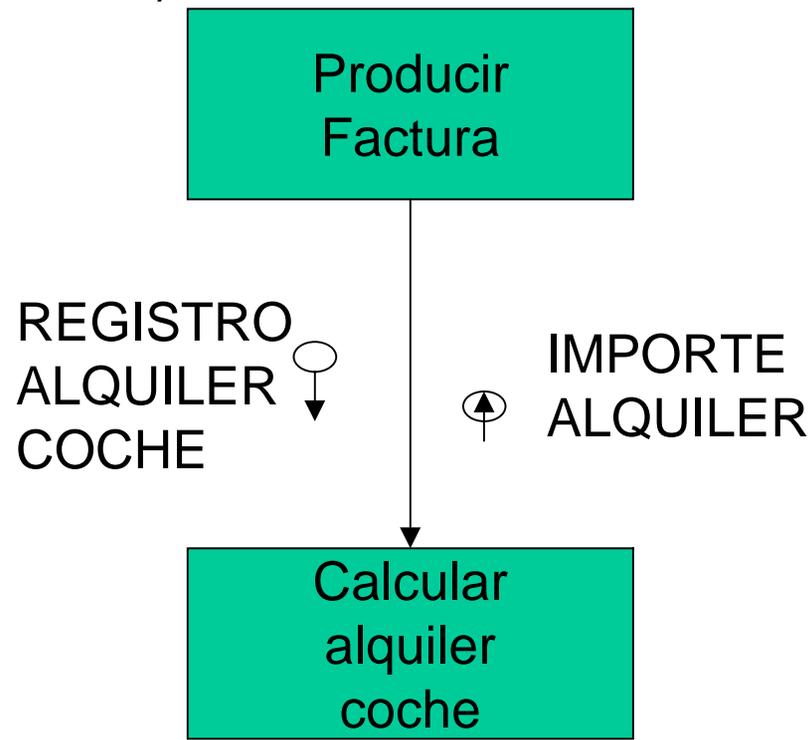
- Dos módulos están acoplados por datos si están acoplados normalmente y además todos los parámetros que se intercambian son datos elementales sin estructura interna (tipos básicos).



Los datos que se pasan en este ejemplo son básicos

# ACOPLAMIENTO NORMAL. POR ESTAMPADO

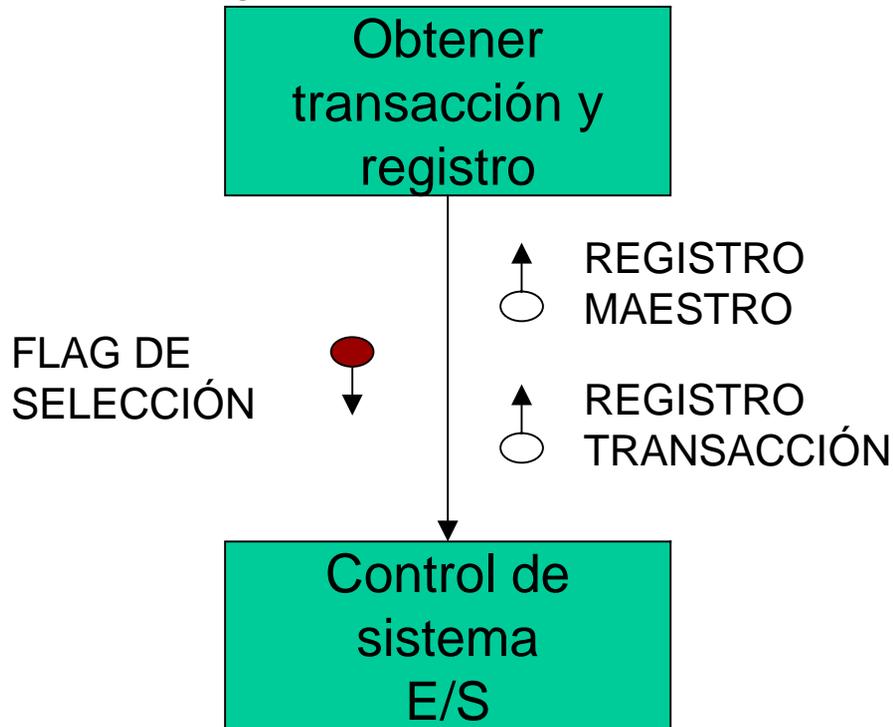
- Dos módulos acoplados normalmente lo están por estampado si uno le pasa al otro un dato compuesto (con estructura interna).



Los datos que se pasan en este ejemplo son compuestos

# ACOPLAMIENTO NORMAL. POR CONTROL

- Dos módulos acoplados normalmente lo están por control si uno le pasa al otro un dato con la intención de controlar su lógica interna.



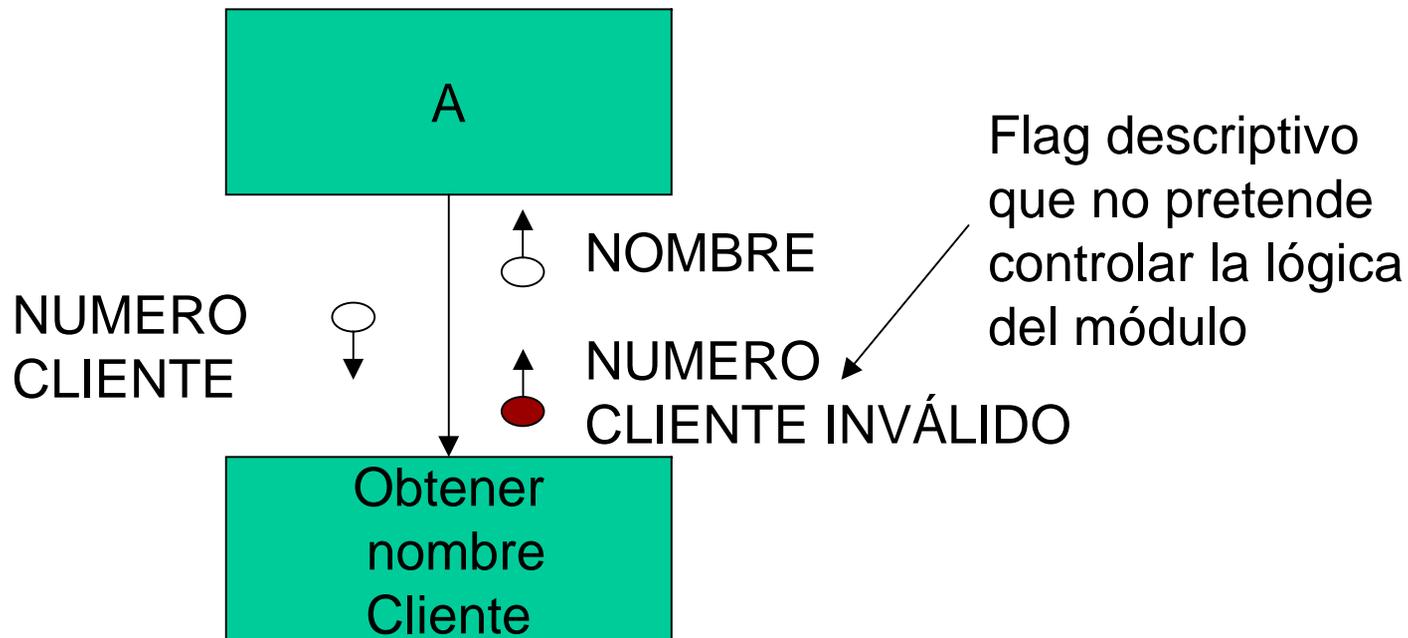
Valores del Flag de Selección:

1. Obtiene registro maestro
  2. Obtiene registro transacción
  3. Obtiene ambos
  4. Imprime cabecera.
- Etc...

El módulo jefe controla, a través del dato “Flag de selección”, la lógica del módulo subordinado. Si fuera al revés se dice que hay una **inversión de autoridad**.

# ACOPLAMIENTO NORMAL. POR CONTROL

- Con frecuencia el acoplamiento por control indica la presencia de algún error de diseño más grave (Que los módulos posean poca cohesión ).
- Dentro del acoplamiento por control hay dos tipos de flag:
  - **Flags de control:** que intentan controlar la lógica del módulo que los recibe.
  - **Flags descriptivos:** que revelan cualidades relativas al dato.



# TIPO DE INFORMACIÓN QUE INTERCAMBIAN MÓDULOS NORMALMENTE ACOPLADOS

TIPO DE INFORMACIÓN	CONTENIDO DEL NOMBRE	EJEMPLO
DATO SIMPLE O ESTRUCTURADO	SUSTANTIVO	Registro cliente Código Postal
FLAG DESCRIPTIVO	ADJETIVO	Registro es válido Código Postal numérico
FLAG DE CONTROL	VERBO	Leer siguiente registro Rechazar este cliente

# ACOPLAMIENTO COMÚN, POR VARIABLES GLOBALES

- Dos módulos A y B están acoplados globalmente si se refieren a una misma zona global de datos o variable global.
- Las áreas de datos globales no son aconsejables:
  - Un error de programación que aparece en un módulo acoplado globalmente puede aparecer en otros módulos que compartan dicha área global.
  - Estos módulos son menos reutilizables
  - Es un tipo de acoplamiento en el cuál es difícil determinar la procedencia de la información depositada en el área global.
  - Los sistemas con muchas áreas globales son muy difíciles de mantener.

# ACOPLAMIENTO POR CONTENIDO

- Se dice que dos módulos están acoplados por contenido si uno se refiere al interior del otro en alguna de las siguientes formas:
  - Modificando o leyendo sus datos internos
  - Saltando directamente al interior de su código
- En este punto se pierde totalmente la modularidad

# COMPARACIÓN DE LOS DISTINTOS TIPOS DE ACOPLAMIENTO

- Dos módulos pueden presentar más de un tipo de acoplamiento. En ese caso siempre se considerará el peor de los dos.

Tipo de acoplamiento	Modificabilidad	Comprensión	Reusabilidad
Por datos	Buena	Buena	Buena
Por estampado	Buena	Media	Media
Por control	Pobre	Pobre	Pobre
Global	Media	Mala	Pobre
Contenido	Mala	Mala	Mala

# COHESIÓN

- La cohesión es una medida de la fuerza de la relación funcional entre los elementos de un módulo.
- Se entiende por elemento de un módulo:
  - Una instrucción
  - Un grupo de instrucciones
  - Una definición de datos
  - Una llamada a otro módulo
- Lo ideal es disponer de módulos fuertemente cohesivos, cuyos elementos tengan poca relación con otros elementos de otros módulos.

# ESCALAS DE COHESIÓN

Cohesión más fuerte (Mejor Mantenimiento)

FUNCIONAL

SECUENCIAL

COMUNICACIONAL

PROCEDURAL

TEMPORAL

LÓGICA

CASUAL



Cohesión más débil (Peor Mantenimiento)

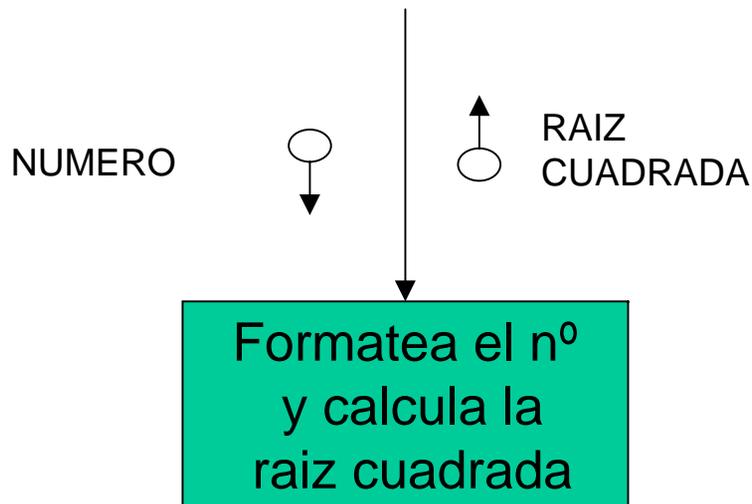
- En los tres primeros niveles los módulos se comportan como cajas negras y por tanto representan los niveles más deseables de cohesión dentro de un sistema.

# COHESIÓN FUNCIONAL

- Un módulo presenta cohesión funcional si contiene elementos que contribuyen a la realización de una sola función.
- Son los módulos más reutilizables ya que solo es necesario conocer la función que cumplen.
- El nombre de este tipo de módulos suele indicar claramente la función que realizan:
  - Calcular suma
  - Validar campo numérico
  - Calcular seno del ángulo

# COHESIÓN SECUENCIAL

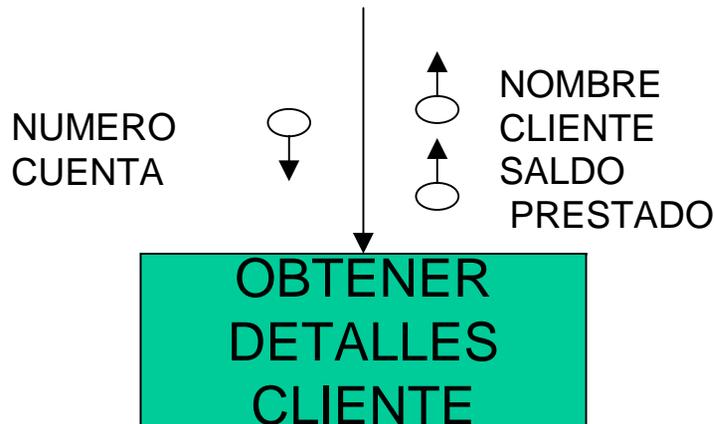
- Un módulo presenta cohesión secuencial si sus elementos están envueltos en tareas, tal que la salida de una tarea sirve de entrada para la siguiente.
- Como se puede ver el módulo implementa un conjunto de funciones relacionadas entre sí.



**Módulo** Formatea n<sup>o</sup> y calcula raiz cuadrada  
**usa** Número  
**retorna** Raiz cuadrada  
**principio**  
Formatea el número  
Calcula raiz cuadrada  
**Fin.**

# COHESIÓN COMUNICACIONAL

- Un módulo presenta cohesión comunicacional si contiene actividades que comparten los mismo datos ya sean de entrada o de salida.



**Módulo** obtener detalles de Cliente  
**usa** Número cuenta  
**retorna** Nombre cliente  
Saldo préstamo

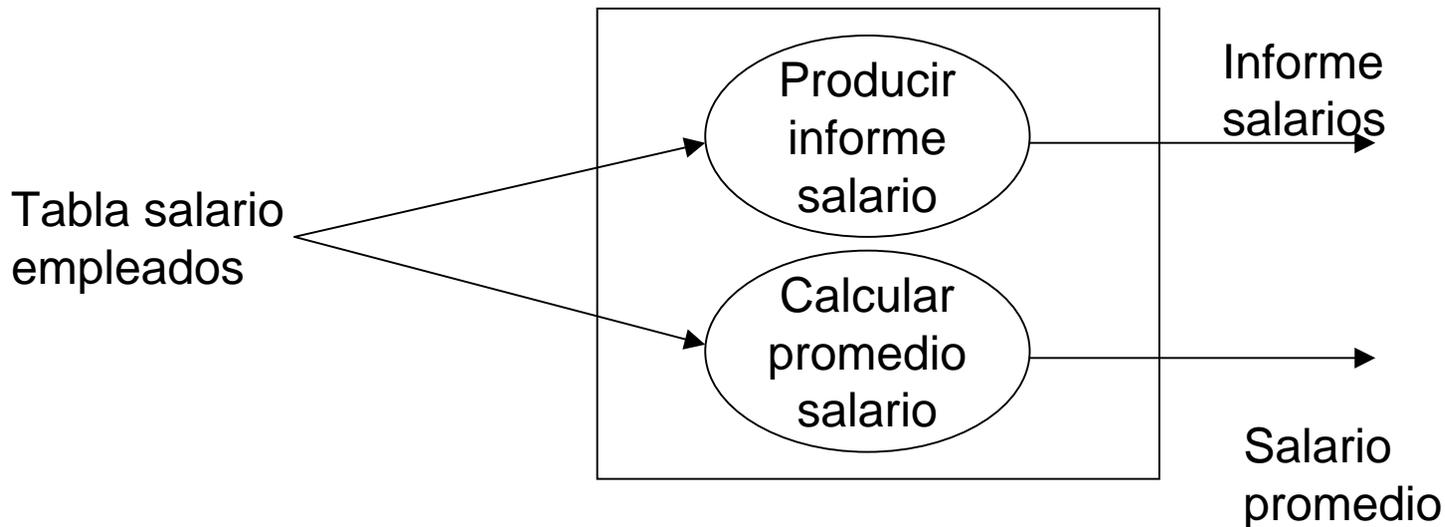
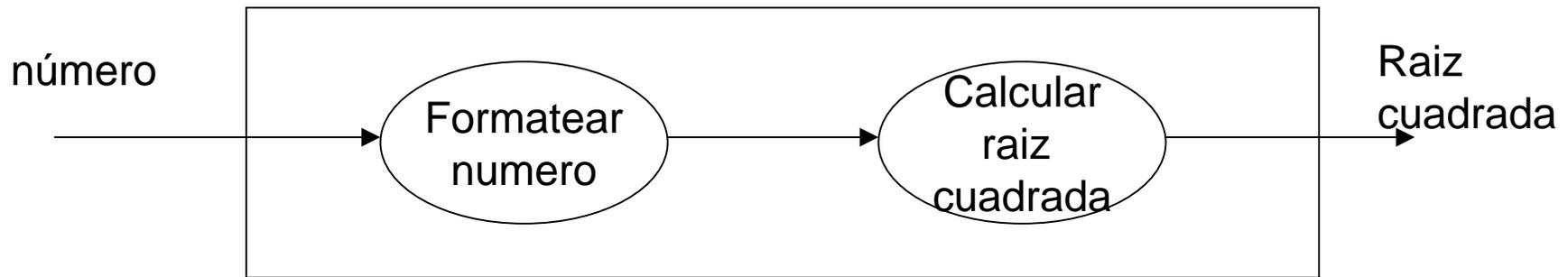
**principio**  
Encontrar nombre  
Encontrar saldo préstamo

**Fin.**

En algunos casos un módulo con cohesión comunicacional puede ser dividido en dos o más módulos con cohesión funcional.

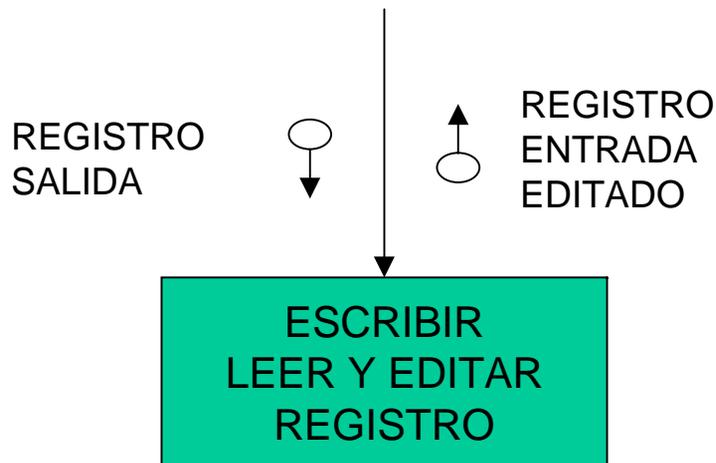
# DIFERENCIAS ENTRE COHESIÓN SECUENCIAL Y COMUNICACIONAL

- En el módulo con cohesión secuencial es relevante el orden mientras que en el comunicacional no.



# COHESIÓN PROCEDURAL

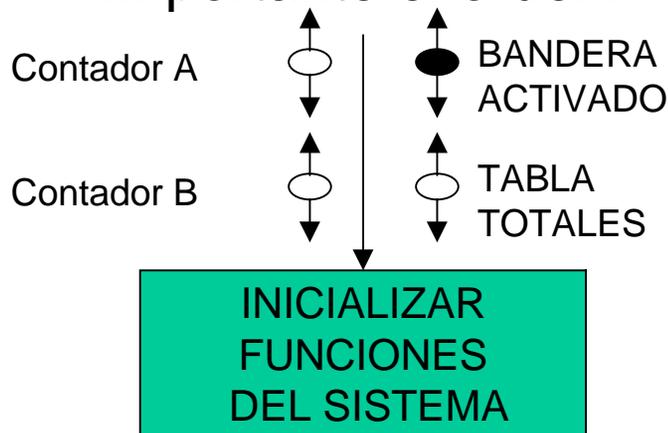
- En un módulo con cohesión procedural los elementos desarrollan actividades diferentes, posiblemente sin relación alguna, tal que el flujo de control fluye de una actividad a la siguiente.
- En estos módulos las diferentes actividades no comparten datos, lo único que las relaciona es el flujo de control.



**Módulo** Escribir, leer y editar registro  
**usa** Registro salida  
**retorna** Registro entrada editado  
**principio**  
Escribir registro salida  
leer registro de entrada  
llenar campos numéricos del registro de entrada con ceros  
**Fin.**

# COHESIÓN TEMPORAL

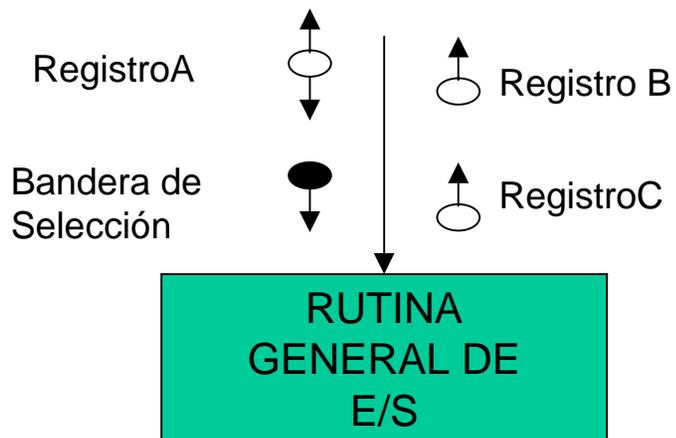
- En un módulo con cohesión temporal los elementos están envueltos en actividades que están relacionadas por el tiempo (se deben desarrollar al mismo tiempo). Normalmente cada una de estas actividades responde a una tarea diferente.
- La cohesión procedural y la temporal son muy parecidas, salvo por el hecho de que en la procedural suele ser más importante el orden.



**Módulo** Inicializar funciones del sistema  
**usa** ContadorA, ContadorB  
Bandera activado y Tabla totales  
**retorna** lo mismo inicializado  
**principio**  
pone los contadores a cero  
limpia tabla de totales  
pone bandera activado en falso  
**Fin.**

# COHESIÓN LÓGICA

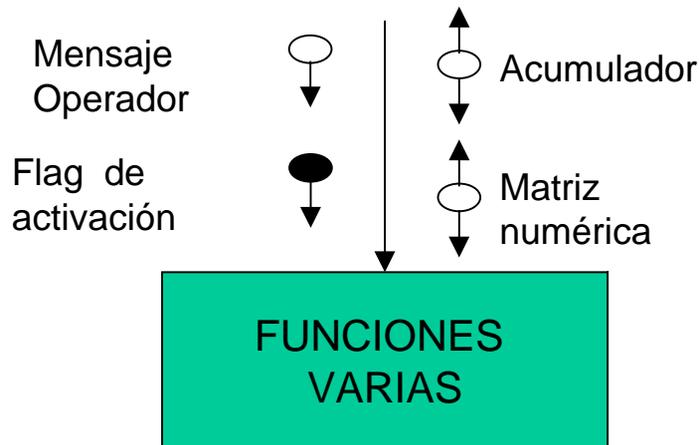
- En un módulo con cohesión lógica los elementos contribuyen en tareas de la misma categoría general y las actividades a desarrollar se seleccionan fuera del módulo.
- Estos módulos se caracterizan por:
  - Necesitan una interfaz ancha que permita seleccionar la actividad a ejecutar.
  - Son módulos difíciles de entender y mantener



**Módulo** Rutina general de E/S  
**usa** RegistroA, Bandera de selección  
**retorna** RegistroA, RegistroB, RegistroC  
**principio**  
si bandera de selección=1  
entonces escribir registro A en Fmaes.  
Leer registroB de Fichero1  
si\_no si bandera de selección=2  
Leer registroB de Fichero1  
si\_no  
Leer registroA de Fichero2  
Leer registroC de Fichero3  
**Fin.**

# COHESIÓN CASUAL

- Un módulo con cohesión casual es aquel cuyos elementos realizan tareas diferentes sin relación significativa entre ellas.
- Estos módulos aparecen debido a:
  - Intentos por ahorrar tiempo o memoria, situando dentro de un módulo trozos de código que se repite a lo largo del sistema.
  - Cambios de mantenimiento mal hechos, en módulos con poca cohesión.



**Módulo** Funciones varias

**usa** Mensaje operador, flag activación, Acumulador, Matriz numérica.

**retorna** Acumulador, Matriz numérica.

**principio**

si flag-activación=1

entonces inicializar Matriz numérica a 0.

Inicializar acumulador a 0.

si\_no si flag-activación=2

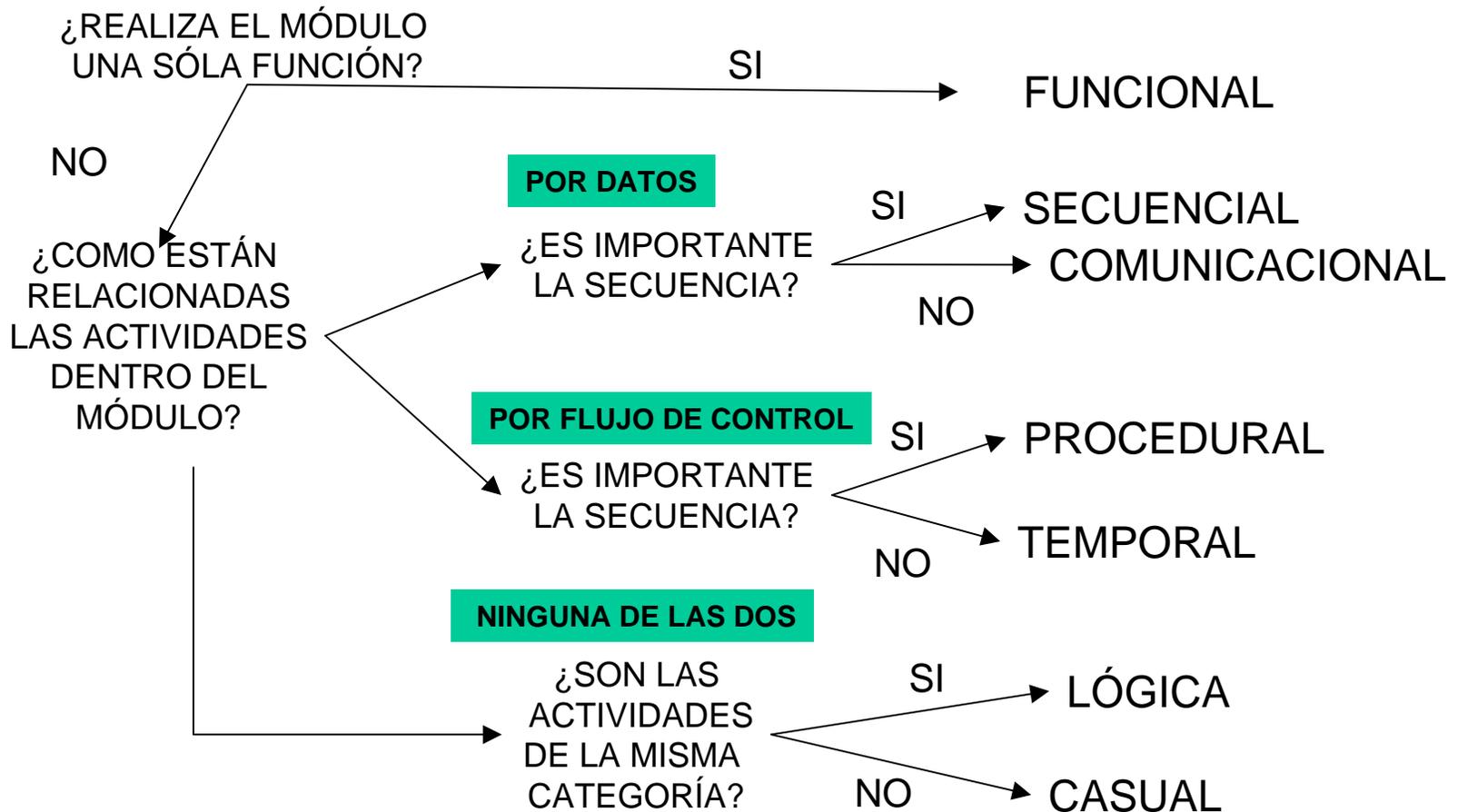
entonces imprimir cabecera

si\_no Escribir en pantalla Mensaje Operador

**Fin.**

# DETERMINACIÓN DE LA COHESIÓN DE UN MÓDULO. ÁRBOL DE COHESIÓN

- A partir de una serie de preguntas situadas en los nodos se determina la cohesión dominante del módulo.

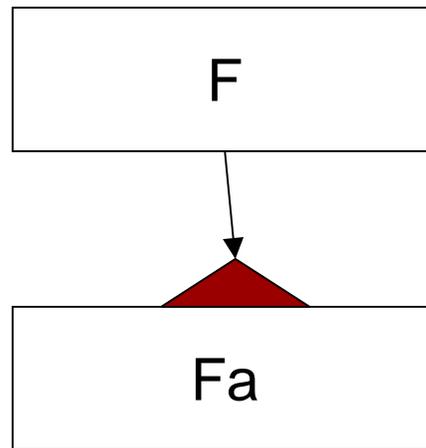


# OTROS CRITERIOS DE DISEÑO

- FACTORIZACIÓN
- POR LA FORMA DEL SISTEMA
  - SISTEMAS BALANCEADOS

# FACTORIZACIÓN

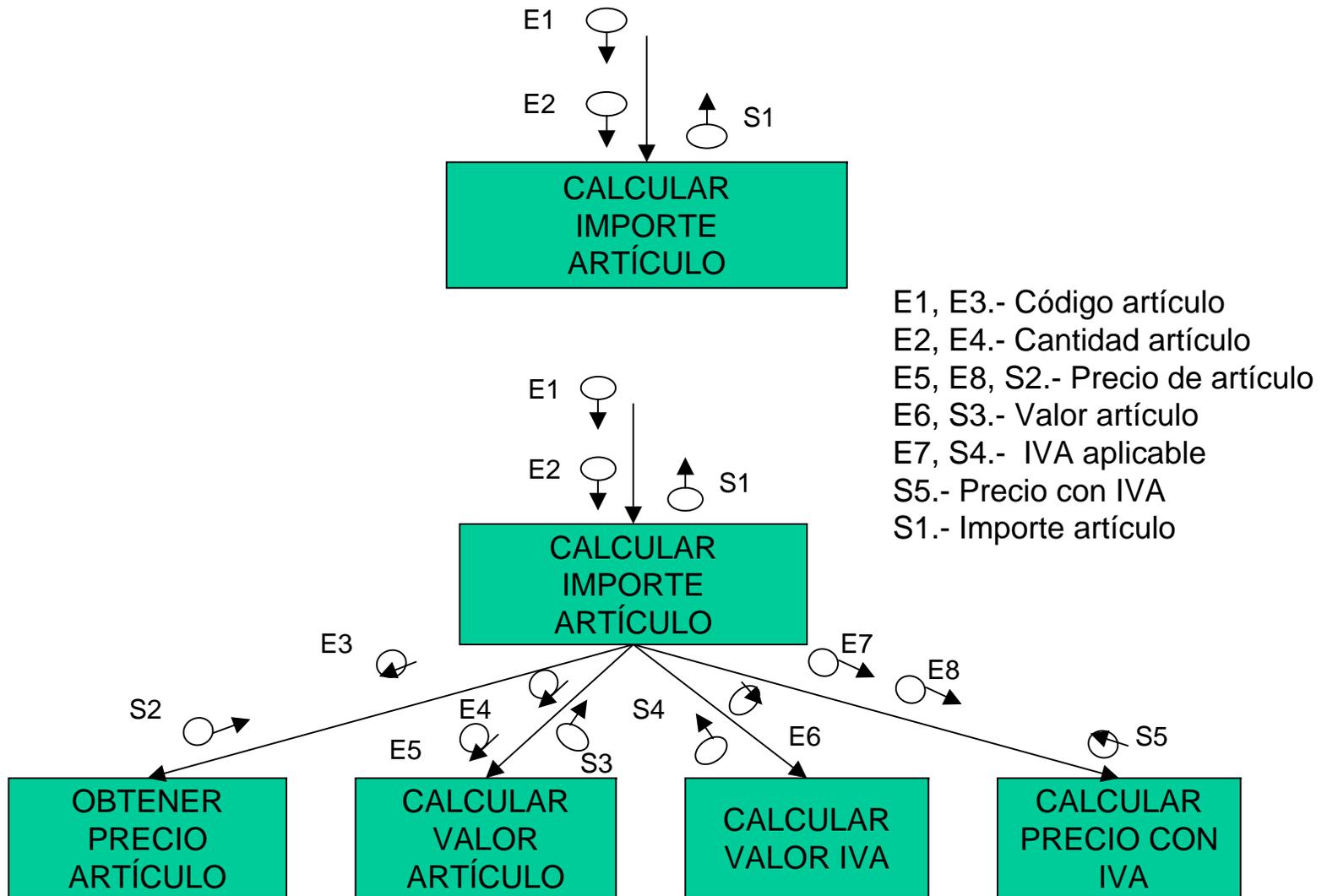
- La factorización consiste en la separación de la función contenida como código dentro de un módulo, para formar un nuevo módulo.
- El proceso inverso se denomina desfactorización y también puede representarse en el D.E.



# RAZONES DE USO DE LA FACTORIZACIÓN

- Reducir el tamaño de los módulos
- Clarificación del sistema obteniendo las ventajas del diseño descendente
- Evitar que una misma función este codificada en varios módulos
- Separar el cálculo y el procesamiento de las decisiones sobre los módulos.
- Crear módulos reusables.
- Simplificar la implementación.

# EJEMPLO DE FACTORIZACIÓN



# CRITERIOS PARA DETENER LA FACTORIZACIÓN

- Cuando no se encuentre una función bien definida susceptible de ser factorizada.
- Cuando las interfaces comiencen a ser tan complejas como los propios módulos de manera que no se mejore la comprensión del módulo.

# CLASIFICACIÓN DE LOS MÓDULOS EN FUNCIÓN DEL FLUJO DE LOS DATOS

- **Aferente:** el módulo envía información de abajo hacia arriba, es decir recoge información de algún módulo subordinado.
- **Eferente:** el módulo envía información de arriba hacia abajo.
- **Transformación:** Toma información del módulo jefe y la transforma en datos que le devuelve.
- **Coordinación:** coordina la información entre los módulos subordinados.
- Los módulos pueden presentar una combinación de estos tipos.

# FORMA DEL SISTEMA

- A partir de la forma del diagrama de estructura se puede evaluar la calidad del diseño.
- En un diagrama de estructura se distinguen tres áreas:
  - aferente: que trata con la información de entrada.
  - Transformación: que procesa los datos de entrada para producir la salida.
  - Eferente: que trata con la información de salida
- A partir del desarrollo de cada una de estas áreas podremos evaluar el diseño.

# SISTEMAS BALANCEADOS

- Un sistema balanceado es aquel en el que los módulos superiores manejan información de naturaleza lógica más que física.
- Son los módulos de niveles inferiores de las ramas aferente y eferente los que tratan con los datos físicos de entrada y salida.
- La ventaja de los sistemas balanceados es que se mejora el acoplamiento entre los módulos y a su mantenibilidad.

# EJEMPLO DE SISTEMA BALANCEADO

