Cohesión y acoplamiento

Cohesión
Medida del grado de identificación
de un módulo con una función concreta.

Cohesión aceptable (fuerte)

- ü COHESIÓN FUNCIONAL (un módulo realiza una única acción).
- ü COHESIÓN SECUENCIAL (un módulo contiene acciones que han de realizarse en un orden particular sobre unos datos concretos).
- ü COHESIÓN DE COMUNICACIÓN (un módulo contiene un conjunto de operaciones que se realizan sobre los mismos datos).
- ü COHESIÓN TEMPORAL (las operaciones se incluyen en un módulo porque han de realizarse al mismo tiempo; p.ej. inicialización).

Cohesión inaceptable (débil)

- ü COHESIÓN PROCEDURAL (un módulo contiene operaciones que se realizan en un orden concreto aunque sean independientes).
- ü COHESIÓN LÓGICA (cuando un módulo contiene operaciones cuya ejecución depende de un parámetro: el flujo de control del módulo es lo único que une a las operaciones que lo forman).
- ü COHESIÓN COINCIDENTAL (cuando las operaciones de un módulo no guardan ninguna relación observable entre ellas).

Hay que procurar evitar situaciones de
cohesión procedural, lógica o coincidental

Acoplamiento
Medida de la interacción
de los módulos que constituyen un programa.

Niveles de acoplamiento (de mejor a peor):

- ACOPLAMIENTO DE DATOS (acoplamiento normal): Todo lo que comparten dos módulos se especifica en la lista de parámetros del módulo invocado.

- ACOPLAMIENTO DE CONTROL: Cuando un módulo pasa datos que le indican a otro qué hacer (el primer módulo tiene que conocer detalles internos del segundo).

- ACOPLAMIENTO EXTERNO: Cuando dos módulos utilizan los mismos datos globales o dispositivos de E/S (p.ej. ficheros).

  Si los datos son de sólo lectura, el acoplamiento se puede considerar aceptable. No obstante, en general, este tipo de acoplamiento no es deseable porque la conexión existente entre los módulos no es visible (de forma explícita).

- ACOPLAMIENTO PATOLÓGICO: Cuando un módulo utiliza el código de otro o altera sus datos locales ("acoplamiento de contenido").

  - Los lenguajes estructurados incluyen reglas para el ámbito de las variables que impiden este tipo de acoplamiento.
  - Los lenguajes orientados a objetos incluyen modificadores de visibilidad para evitar este tipo de acoplamiento.

Objetivo final
Reducir al máximo el acoplamiento entre módulos
y aumentar la cohesión interna de los módulos.