Cohesión y acoplamiento

José Galaviz

Cohesión

- Cualidad deseable. Se prefieren módulos con alta cohesión.
- Se refiere al grado en que los elementos de un módulo o clase están lógicamente vinculados. Que tan afines son.
- La cohesión se traduce en cualidades del software: robustez, confiabilidad, reusabilidad y legibilidad.

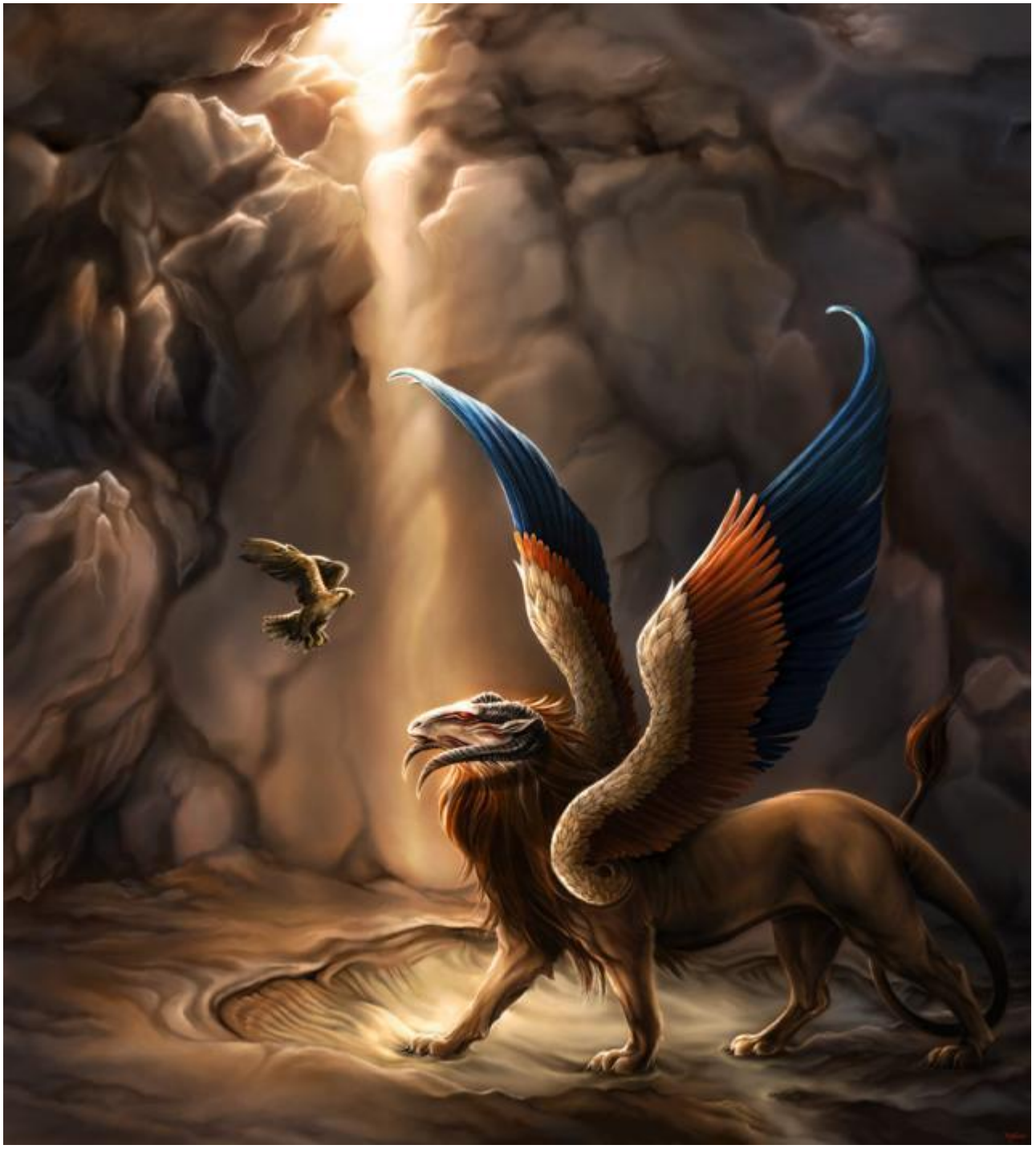
Intuitivamente

Queremos que las cosas contenidas en un módulo o en una clase tengan mucho que ver entre sí, sean parte del mismo concepto, de la misma abstracción.

En contraste

La baja cohesión se evidencia en módulos o clases en las que hay un conjunto heterogéneo de objetos o funciones que hacen cosas completamente diferentes o sirven para tareas no relacionadas.

Bajo acoplamiento



Tipos de cohesión: 1-3

1. Cohesión funcional. Las partes del módulo están allí porque todas contribuyen a una tarea única y bien definida de la que el módulo es responsable.
2. Cohesión secuencial. Las funciones del módulo se encadenan: la salida de una, es la entrada de otra: función que recibe un paquete de datos y otra que extrae la dirección IP destino.
3. Cohesión comunicativa o de información. Las funciones del módulo operan sobre el mismo conjunto de datos: función que extrae la dirección IP destino del paquete y otra que verifica el CRC del paquete.

Tipos de cohesión: 4-6

4. Cohesión procedural. Las funciones del módulo forman parte de un mismo procedimiento más general: función que verifica que un archivo exista, otra que verifica los permisos y otra que lo abre.
5. Cohesión temporal. Las funciones del módulo se ejecutan en un momento particular del programa: función que despliega los resultados en la pantalla y otra que cierra el archivo.
6. Cohesión lógica. De alguna manera, obedeciendo a cierta lógica, todas las funciones del módulo están relacionadas, por ejemplo: todas las de entrada/salida.

Tipos de cohesión: 7. El peor

7. Cohesión accidental. Las partes del módulo no tienen relación clara: *utilities*.

Acoplamiento

- Característica indeseable. Se prefiere bajo acoplamiento.
- Se refiere al grado en el que un módulo depende de otro u otros módulos.
- El bajo acoplamiento se traduce en cualidades: robustez, confiabilidad, reusabilidad y legibilidad.

Intuitivamente

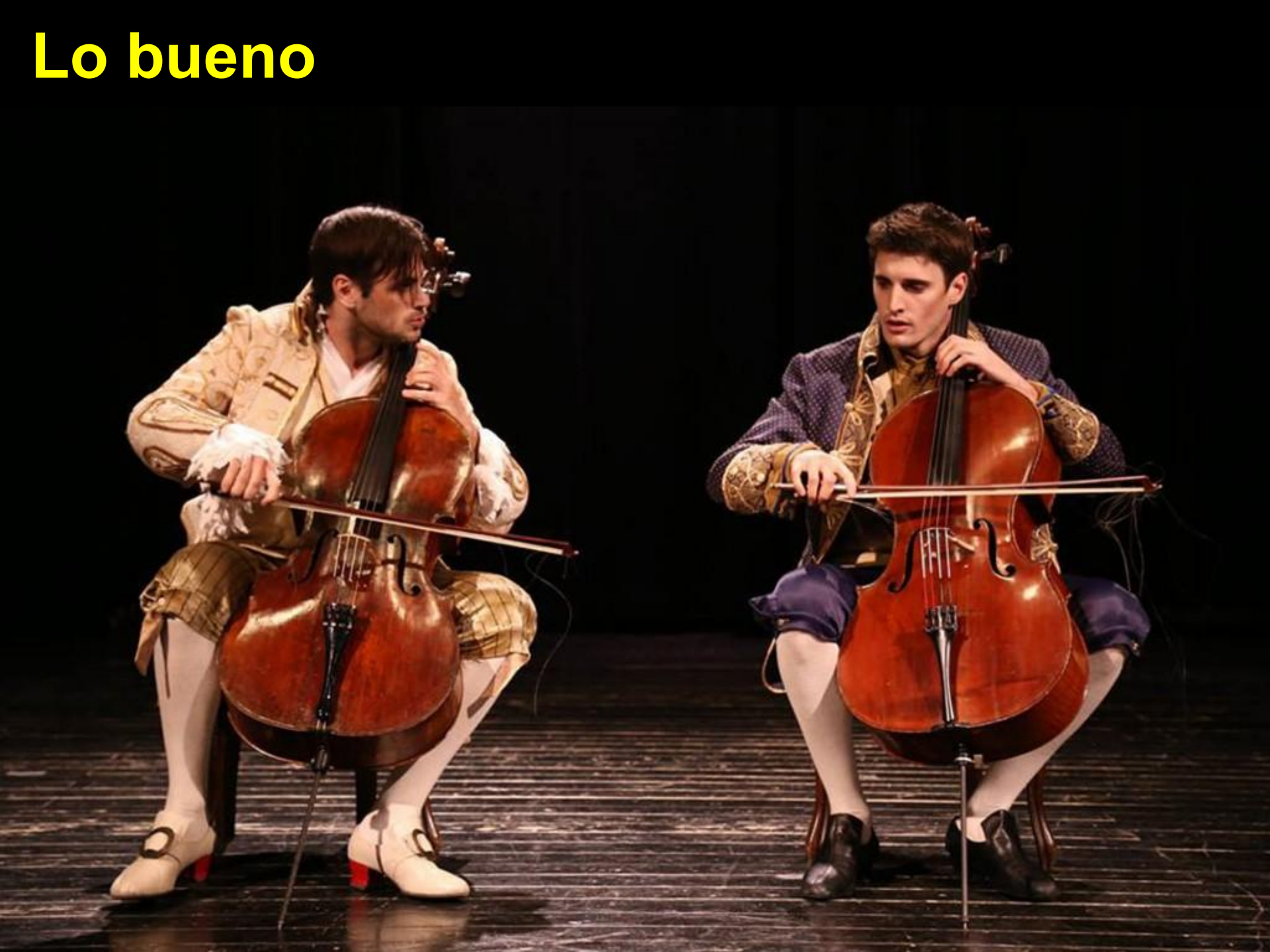
Queremos que cada módulo sea operacional *per se*, sin depender de otros. Esto no tiene sentido a ultranza, claro, pero se procura maximizar la independencia.

En contraste

El alto acoplamiento se percibe en módulos en los que las tareas a realizar dependen de la ejecución de otras en otros módulos o de cómo se llevan a cabo esas tareas.

Lo malo

Lo bueno



Tipos de acoplamiento: 1-3

1. Sin acoplamiento: dos módulos no se comunican entre sí.
2. Acoplamiento por mensajes. Un módulo le envía mensajes a otro.
3. Acoplamiento de datos. Un módulo llama a ejecutar procedimientos del otro y le pasa parámetros.

Tipos de acoplamiento: 4-5

4. Acoplamiento por estructura compartida (*stamp coupling*). Los módulos comparten una estructura de datos, si uno cambia la estructura el otro debe cambiar su operación, aún cuando no requiera la parte cambiada.
5. Acoplamiento de control. Un módulo controla el flujo de otro diciéndole que hacer a través de variables que codifican la acción (what-to-do flag).

Tipos de acoplamiento: 6-8, lo peor

6. Acoplamiento externo. Un módulo utiliza a otro que le impone un cierto formato de datos, un protocolo de comunicación o una interfaz de dispositivo.
7. Acoplamiento global. Un módulo (o varios) comparten datos globales (o comunes a ellos).
8. Acoplamiento de contenido. Un módulo depende, para su operación, del acceso a datos contenidos en otro o de la implementación de otro.

Cuantificación de acoplamiento

Variable	Significado
d_i	Datos de entrada
d_o	Datos de salida
c_i	Parámetros de entrada usados para control
c_o	Valores de salida usados para control
g_d	Variables globales usadas como dato
g_c	Variables globales usadas para control
f_o	Fan-out, módulos usados
f_i	Fan-in, módulos que lo usan

Cuantificación

$$C = 1 - \frac{1}{d_i + d_o + g_d + f_o + f_i + 2(c_i + c_o + g_c)}$$

Por ejemplo si un módulo es usado por otro (f_i = 1) recibe un parámetro de entrada (d_i = 1) y entrega un resultado (d_o = 1) de salida:

$$C = 1 - \frac{1}{3} = 0.67$$

Cuanto mayor el valor, peor el acoplamiento