



Ingeniería de software II

Diseño de Software - Conceptos



Diseño de Software

»Temas de Hoy
Tipos de diseño
Criterios técnicos
Principios
Conceptos

2



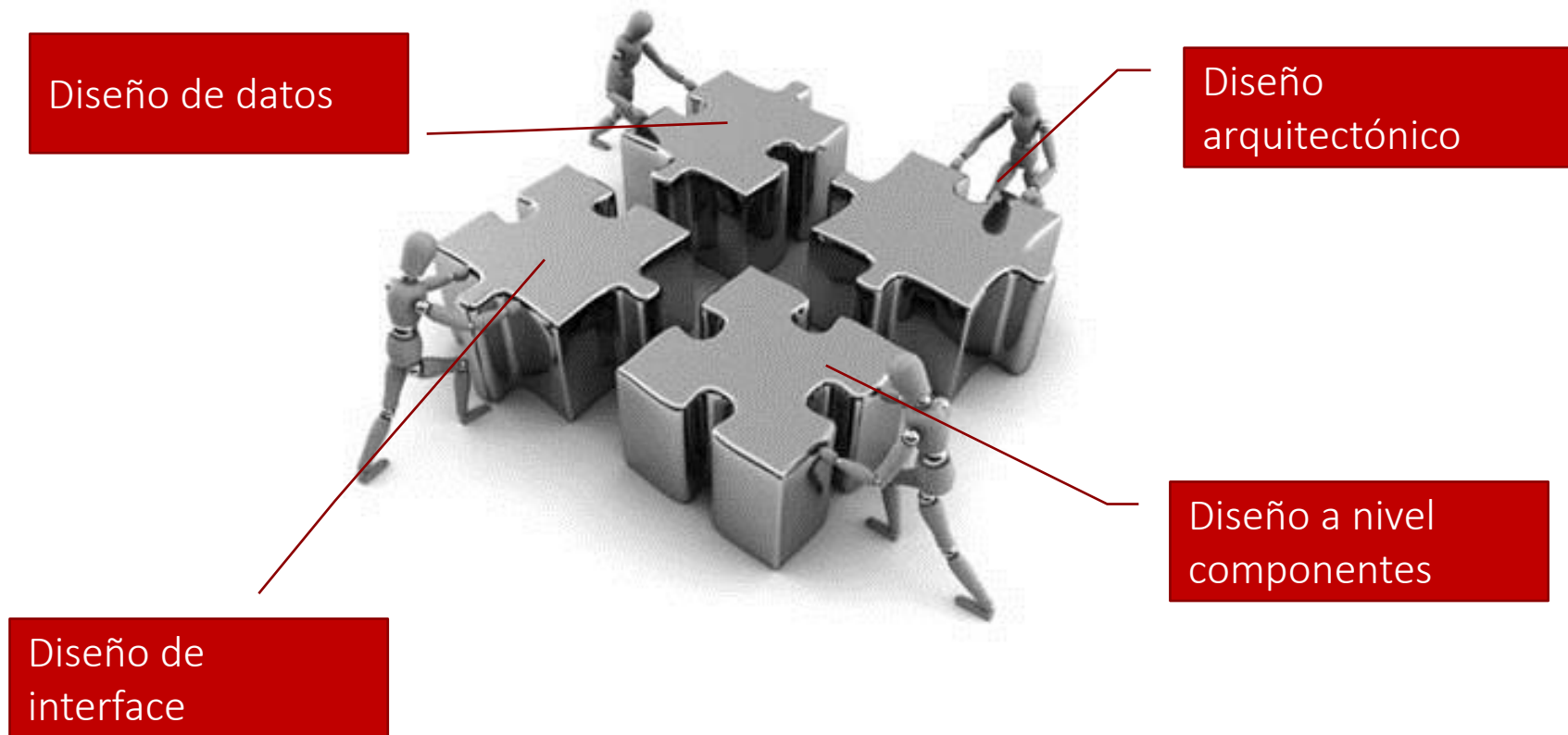
Diseño de Software

- » Es una representación significativa de ingeniería de algo que se va a construir.
- » Es el proceso creativo de transformación del problema en una solución.
- » Es el núcleo técnico de la ingeniería de software.
Una vez que se analizan y especifican los requisitos del software, el diseño es la primera actividad técnica a realizar
- » Es independiente del modelo de proceso que se utilice.
- » El diseño se centra en cuatro áreas importantes:
Datos, Arquitecturas, Interfaces y Componentes.

3



Diseño de Software - Tipos



4

Diseño de Software - Tipos

» Diseño de datos

Transforma el modelo del dominio, obtenido del análisis, en estructuras de datos, objetos de datos, relaciones , etc.

No vamos a entrar en detalles de modelado de datos (IBD)

» Diseño arquitectónico

Define la relación entre los elementos estructurales más importantes del software, los estilos arquitectónicos, patrones de diseño, etc., para lograr los requisitos del sistema.

La información para realizar el diseño puede derivarse de la especificación, del modelo de análisis y de la interacción de los subsistemas definidos.

5



Diseño de Software - Tipos

» Diseño a nivel componentes

Transforma los elementos estructurales de la arquitectura de software en una descripción procedimental de los componentes del software.

La información obtenida de los modelos basados en clases, modelos de flujos, de comportamiento (DTE) sirven como base.

» Diseño de interface

Describe la forma de comunicación dentro del mismo sistema, con otros sistemas, y con las personas.

Una interface implica flujo de información (datos o control) y comportamiento.

6



Diseño de Software

- » El diseño es la etapa en la que se fomentará la calidad.
- » Proporciona las representaciones del software susceptibles de evaluar respecto de la calidad
- » Sin diseño se corre el riesgo de construir un sistema inestable, el cual fallará cuando se realicen cambios pequeños, será difícil de probar.

7



Diseño de Software

» Características para la evaluación de un diseño

El diseño deberá implementar todos los requisitos explícitos del modelo de análisis, y deberá ajustarse a todos los requisitos implícitos que desea el cliente.

El diseño deberá ser una guía legible y comprensible para aquellos que generan código y para aquellos que comprueban y consecuentemente, dan soporte al software.

El diseño deberá proporcionar una imagen completa del software, enfrentándose a los dominios de comportamiento funcionales y de datos desde una perspectiva de implementación.

8



Criterios técnicos para un buen diseño

1. Un diseño deberá presentar una estructura arquitectónica que:

Se haya creado mediante patrones de diseño reconocibles.

Que esté formado por componentes que exhiban características de buen diseño.

Se implemente en forma evolutiva.

2. Un diseño deberá ser modular, el software deberá dividirse lógicamente en elementos que realicen funciones y sub-funciones específicas.
3. Un diseño deberá contener distintas representaciones de datos, arquitectura, interfaces y componentes (módulos).

9



Criterios técnicos para un buen diseño

10

4. Un diseño deberá conducir a estructuras de datos adecuadas para los objetos que se van a implementar y que procedan de patrones de datos reconocibles.
5. Un diseño deberá conducir a componentes que presenten características funcionales independientes.
6. Un diseño deberá conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y con el entorno externo.
7. Un diseño deberá derivarse mediante un método repetitivo y controlado por la información obtenida durante el análisis de los requisitos del software.
8. Un diseño debe representarse por medio de una notación que comunique de manera eficaz su significado.



Diseño

- » El diseño es tanto un proceso como un modelo.
- » El proceso de diseño es una secuencia de pasos que hacen posible que el diseñador describa todos los aspectos del software que se va a construir.
- » El modelo de diseño es equivalente a los planos de un arquitecto para una casa. Comienza con la totalidad y refina. Proporciona diversas visiones.

11



Principios del Diseño

1. En el proceso de diseño se deben tener en cuenta enfoques alternativos.
2. El diseño deberá poderse rastrear hasta el modelo de análisis.
3. El diseño no deberá inventar nada que ya esté inventado.
4. El diseño deberá minimizar la distancia intelectual entre el software y el problema.
5. El diseño deberá presentar uniformidad e integración.
6. El diseño deberá estructurarse para admitir cambios.

12



Principios del Diseño

7. El diseño deberá estructurarse para degradarse poco a poco, incluso cuando se enfrenta con datos, sucesos o condiciones de operaciones aberrantes.
8. El diseño no es escribir código y escribir código no es diseñar.
9. El diseño deberá evaluarse en función de la calidad mientras se va creando, no después de terminado.
10. El diseño deberá revisarse para minimizar los errores conceptuales.

13



Conceptos de Diseño

“El comienzo de la sabiduría para un ingeniero de software es reconocer la diferencia entre hacer que un programa funcione y conseguir que lo haga correctamente”.

M. A. Jackson

14

Los conceptos de diseño de software fundamentales proporcionan el marco de trabajo necesario para conseguir que lo haga correctamente.



Conceptos de Diseño

- » Cada concepto proporciona al diseñador una base para aplicar los métodos de diseño
- » Los conceptos van a ayudar al diseñador a responder esas preguntas

15



Conceptos de Diseño

- » Abstracción
- » Arquitectura
- » Patrones
- » Modularidad
- » Ocultamiento de información
- » Independencia funcional
- » Refinamiento
- » Refabricación



16

Conceptos de Diseño

» Abstracción

La noción de abstracción permite concentrarse en un problema a un nivel de generalización sin tener en cuenta los detalles irrelevantes de bajo nivel

17

Nivel de abstracción

A medida que profundizamos en la solución del problema se reduce el nivel de abstracción.

Desde los requerimientos (abstractos) hasta llegar al código fuente.

Conceptos de Diseño

»Tipos de abstracción

Procedimental

Secuencia “nombrada” de instrucciones que tienen una funcionalidad específica.

Ej.: Módulos (procedimientos, funciones, unidades, etc.)

De datos

Colección “nombrada” de datos que definen un objeto real

Ej.: un registro que representa una persona con sus datos, el objeto persona en POO

18



Conceptos de Diseño

» Arquitectura del software

Es la estructura general del software y las formas en que la estructura proporciona una integridad conceptual para un sistema.

Más adelante se estudiarán las arquitecturas con más detalle

19

Conceptos de Diseño

»Patrones

“Un patrón es una semilla de conocimiento, la cual tiene un nombre y transporta la esencia de una solución probada a un problema recurrente dentro de cierto contexto”.

Dicho de otro modo, describe una estructura de diseño que resuelve un problema de diseño particular dentro de un contexto específico.

20



Conceptos de Diseño

»Patrones

La finalidad de cada patrón de diseño es proporcionar una descripción que le permita al diseñador determinar

Si es aplicable al trabajo actual

Si se puede reutilizar

Si puede servir como guía para desarrollar un patrón similar pero diferente en cuanto a la funcionalidad o estructura.

21



Conceptos de Diseño

» Modularidad

El software se divide en componentes nombrados y abordados por separado, llamados frecuentemente módulos, que se integran para satisfacer los requisitos del problema.

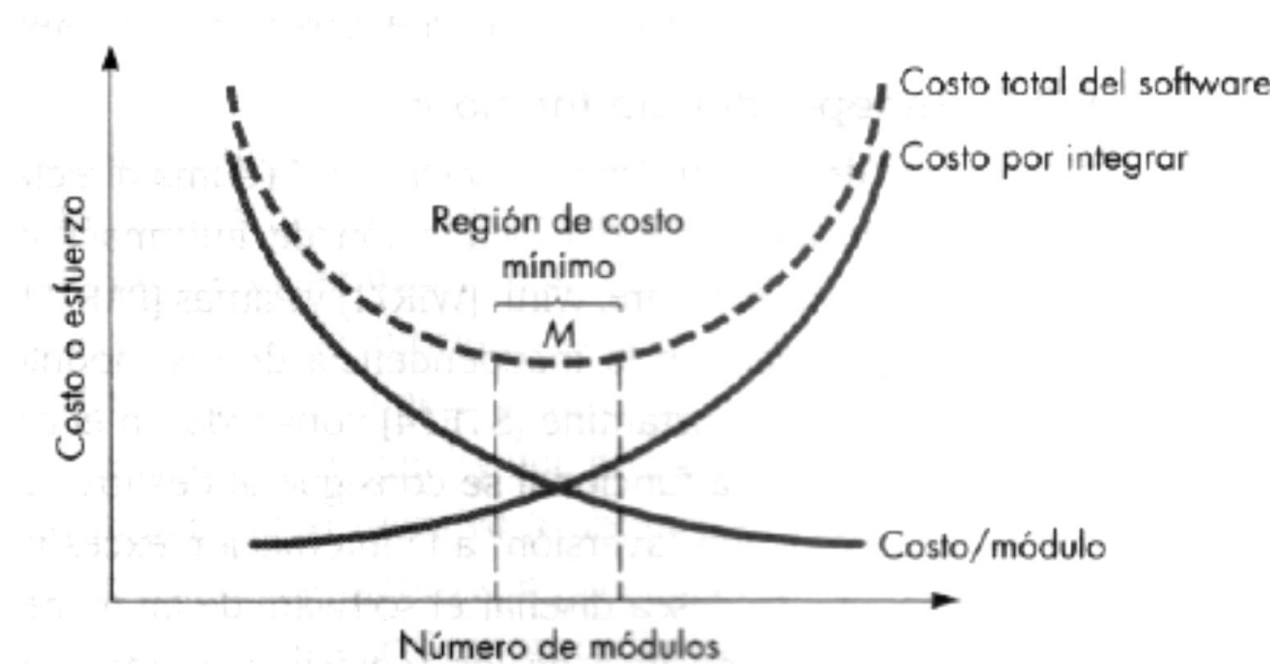
22

Códigos Monolíticos (un único módulo) y
Modularización excesiva (a nivel de instrucciones)



Conceptos de Diseño

» ¿Cuántos módulos tiene que tener un programa?



23

Conceptos de Diseño

» Ocultamiento de información

La información que está dentro un módulo es inaccesible a otros que no la necesiten. Se consigue una modularidad efectiva definiendo un conjunto de módulos independientes que se comunican entre sí intercambiando sólo la información necesaria para su funcionalidad.

24



Conceptos de Diseño

» Independencia Funcional

Modularidad + Abstracción + Ocultamiento de Información.

Es deseable que cada módulo trate una subfunción de requisitos y tenga una interfaz sencilla para que sea más fácil de desarrollar, mantener, probar y reusar

Se mide mediante la cohesión y el acoplamiento entre los módulos

Se busca una alta cohesión y bajo acoplamiento

25



Conceptos de Diseño

» Independencia Funcional

Cohesión (Coherente)

Se define como la medida de fuerza o relación funcional existente entre las sentencias o grupos de sentencias de un mismo módulo.

Un módulo es altamente cohesivo cuando lleva a cabo solo una tarea dentro del procedimiento y requiere poca interacción con el resto de los procedimientos

Un módulo es poco cohesivo cuando realiza tareas muy diferentes o sin relación entre ellas

26



Conceptos de Diseño

» Independencia Funcional

Tipos de Cohesión

Funcional → Cuando las sentencias o grupos de sentencias de un mismo módulo están relacionadas en el desarrollo de una única función (la cohesión más alta).

Coincidental (Casual) → Cuando las sentencias llevan a cabo un conjunto de tareas que no están relacionadas o tienen poca relación (la cohesión más baja)

Lógica → Cuando las sentencias se relacionan lógicamente

27



Conceptos de Diseño

» Independencia Funcional

Tipos de Cohesión

Temporal → Cuando las sentencias se deben ejecutar en el mismo intervalo de tiempo

Procedimental → Cuando las sentencias tiene que ejecutarse en un orden específico

Comunicacional → Cuando los elementos de procesamiento se centran en los datos de entrada y salida

28



Conceptos de Diseño

» Independencia Funcional

Acoplamiento

Es la medida de interconexión entre los módulos

Punto donde se realiza la entrada o referencia y los datos que pasan a través de la interfaz.

Una conectividad sencilla entre módulos da como resultado una conectividad mas fácil.

29

Conceptos de Diseño

» Independencia Funcional Niveles de Acoplamiento

Bajo:

Acoplamiento de datos

Acoplamiento de marca

Moderado:

Acoplamiento de control

Alto:

Acoplamiento común

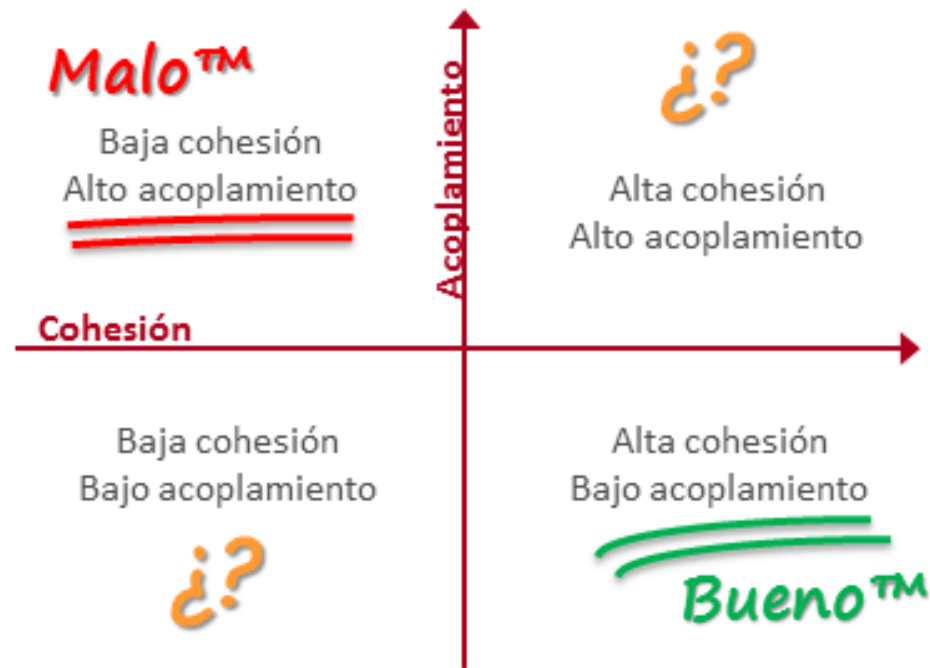
Acoplamiento externo

Acoplamiento de contenido

30



Conceptos de Diseño



Conceptos de Diseño

Independencia Funcional - acoplamiento

a y d no hay acoplamiento

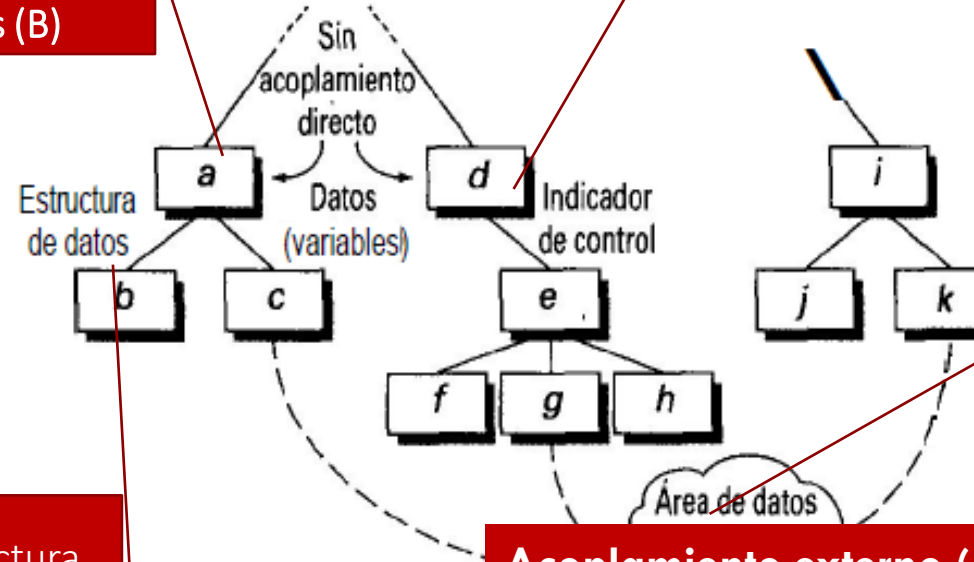
a y c lista de argumentos por la que pasan datos
Acoplamiento de datos (B)

d y e se pasa un indicador de control
Acoplamiento de control (M)

c, g y k utilizan variables globales o compartidas (por ejemplo archivos)
Acoplamiento común (A)

a y b pasa parte de la estructura de datos (no un argumento simple)
Acoplamiento de marca (B)

Acoplamiento externo (A):
E/S protocolos de comunicación
Acoplamiento de contenido (A):
Un módulo hace uso de datos de control mantenidos dentro de los límites de otro módulo



32

Conceptos de Diseño

»Refinamiento

Se refina de manera sucesiva los niveles de detalle procedimentales.

El refinamiento es un proceso de elaboración.

Se comienza con una descripción de información de alto nivel de abstracción, sobre una funcionalidad puntual, sin conocer las características del funcionamiento, se va trabajando sobre la funcionalidad original proporcionando en cada iteración un mayor nivel de detalle hasta obtener todos los detalles necesarios para conocer su funcionamiento.

La abstracción y el refinamiento son conceptos complementarios. La abstracción permite especificar procedimientos y datos sin considerar detalles de grado menor. El refinamiento ayuda a revelar los detalles de grado menor mientras se realiza el diseño.

33



Conceptos de Diseño

» Refabricación o rediseño (Refactoring)

Técnica de reorganización (sugerida por las metodologías ágiles) que simplifica el diseño de un componente sin cambiar su función o comportamiento.

Cuando se refabrica el diseño existente, se examina en busca de redundancias, elementos inútiles, algoritmos innecesarios, estructuras de datos inapropiadas, etc.

Ej : Una primera iteración del diseño podría producir un componente con poca cohesión. El diseñador puede decidir que el componente debe refabricarse en componentes distintos para elevar la cohesión.

34

