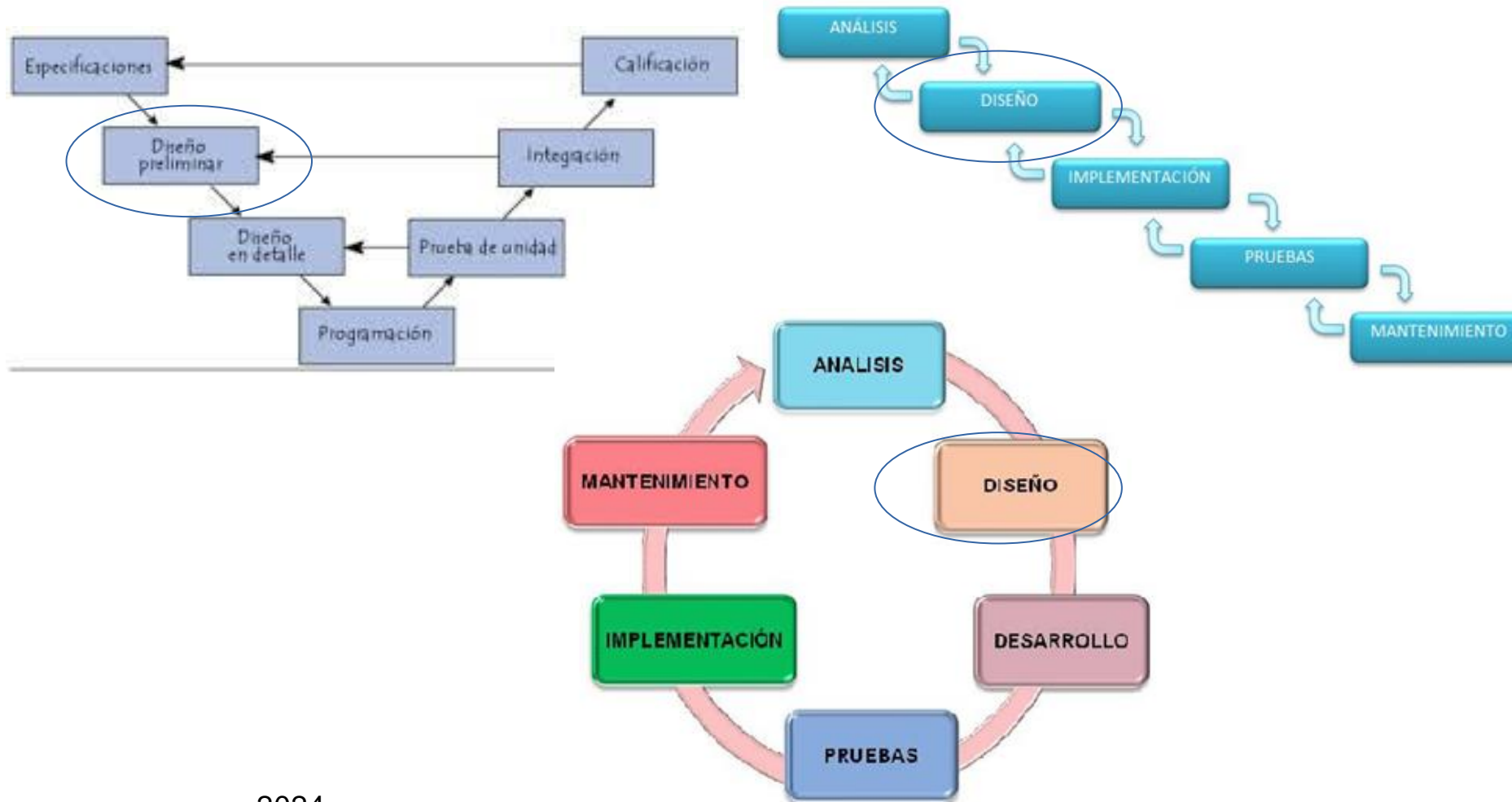




Diseño de Software - Conceptos

Ingeniería de software II - 2024

Diseño de Software



2

Diseño de Software



¿Qué es?



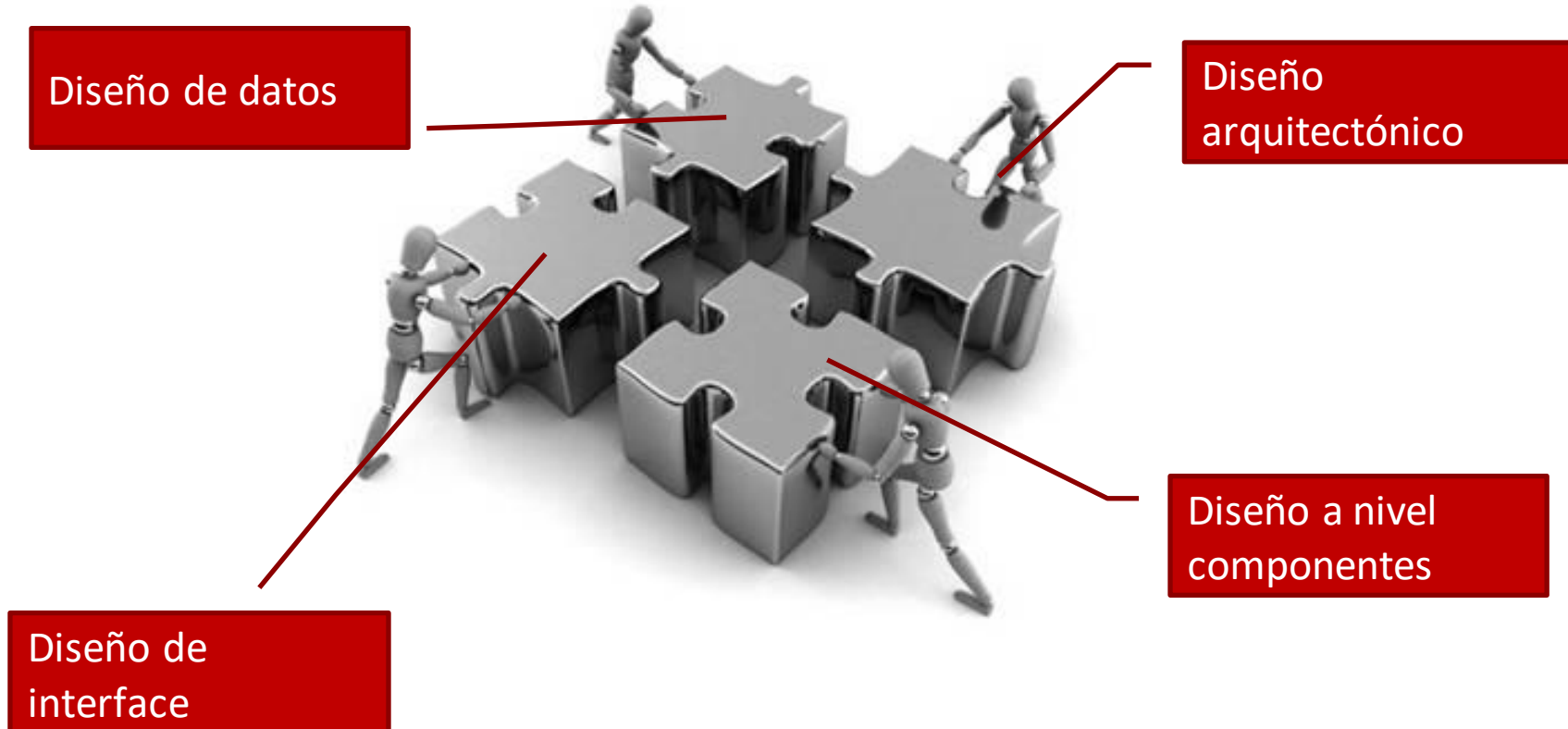
Áreas



¿Por qué es importante?

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.

5

Diseño de Software - Tipos



Diseño
arquitectónico

- Define la relación entre los elementos estructurales del software, los estilos arquitectónicos, patrones de diseño, etc.

6

Diseño de Software - Tipos

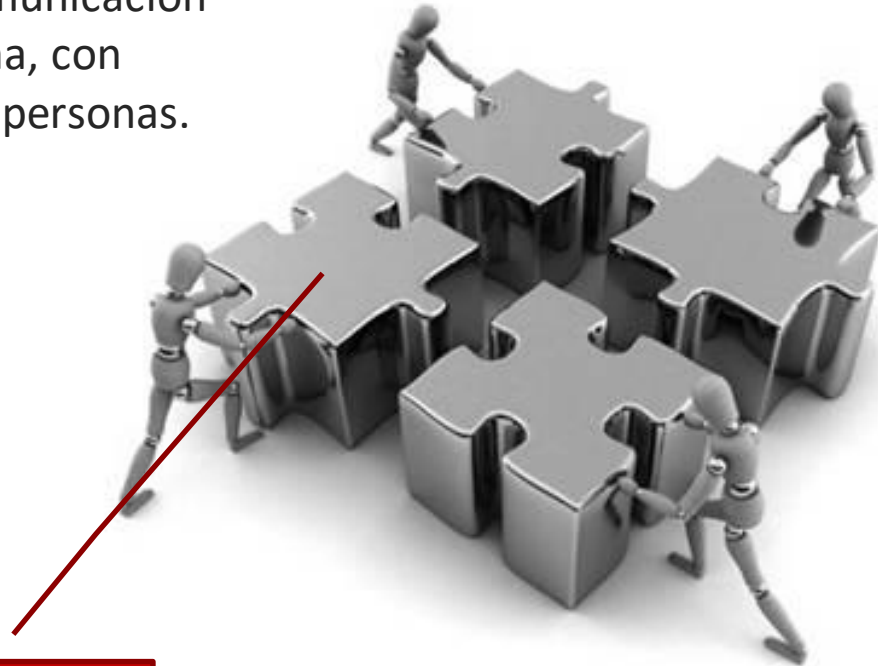
- Transforma los elementos estructurales de la arquitectura de software en una descripción procedimental de los componentes del software.



Diseño a nivel
componentes

Diseño de Software - Tipos

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



Diseño de
interface

8

Diseño de Software - Características para su evaluación

- ❖ Deberá *implementar* todos los requerimientos explícitos del modelo de requerimientos, y e *incorporar* todos los requerimientos implícitos que desea el cliente.
- ❖ Deberá ser una *guía* legible y comprensible para aquellos que generan código y para aquellos que dan soporte al software.
- ❖ Deberá proporcionar una imagen/visión completa del software. (Compleitud)

9

Criterios técnicos para un buen diseño



1. Deberá presentar una estructura arquitectónica que:
Se haya creado mediante patrones de diseño reconocibles, que esté formado por componentes con buen diseño y se implemente en forma evolutiva.
1. Deberá ser modular.
2. Deberá contener distintas representaciones.
3. Deberá conducir a estructuras de datos adecuadas y que procedan de patrones de datos reconocibles.
4. Deberá conducir a componentes que presenten características funcionales independientes.
5. Deberá conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y con el entorno externo
6. Deberá derivarse mediante un método repetitivo y controlado por la información obtenida durante el análisis de los requisitos del software.
7. Deberá representarse por medio de una notación que comunique de manera eficaz su significado.

10

Evolución del diseño de software

Es un proceso continuo que abarca mas de seis décadas
Desde Programas modulares y refinamiento de estructuras de software, a enfoques orientado a objetos, orientado a modelos o a pruebas.
Todos tienen características comunes:

1. un mecanismo para traducir el modelo de requerimientos en una representación de diseño
2. una notación para representar los componentes funcionales y sus interfaces
3. heurísticas para refinamiento
4. lineamientos para evaluación de calidad

Sin importar el tipo de diseño, es necesario aplicar un conjunto de conceptos básicos.

Conceptos de Diseño - Importancia

- ❖ ¿Qué criterios se usan para dividir el software en sus componentes individuales?
- ❖ ¿Cómo se extraen los detalles de la función o la estructura de datos de la representación conceptual del software?
- ❖ ¿Cuáles son los criterios uniformes que definen la calidad técnica de un diseño de software?

12

Conceptos de Diseño



13

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

Conceptos de Diseño

❖ Abstracción

Permite concentrarse en un problema a un nivel de *generalización* sin tener en cuenta los detalles de bajo nivel

Tipos :

Procedimental	<i>Secuencia “nombrada” de instrucciones que tienen una funcionalidad específica</i>
De datos	<i>Colección “nombrada” de datos que definen un objeto real</i>

14



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

15



Conceptos de Diseño

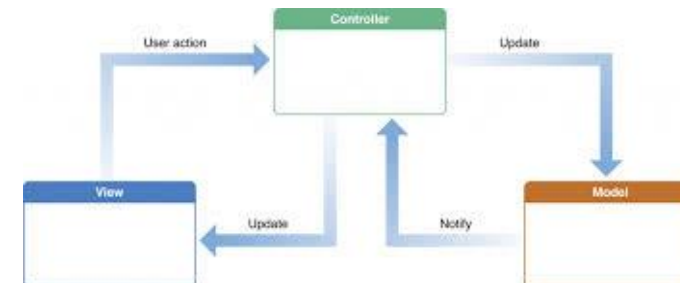
❖ Patrones

Describen una estructura de diseño que resuelve un problema particular dentro de un contexto específico.

Deben proporcionar una descripción que permita determinar si

- ✓ es aplicable al trabajo
- ✓ se puede reutilizar
- ✓ puede servir como guía para desarrollar un patrón similar pero diferente en cuanto a la funcionalidad o estructura.

16

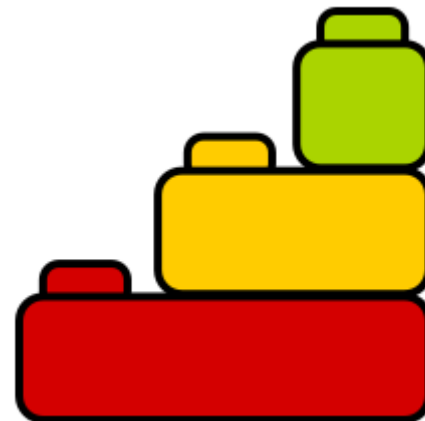


Conceptos de Diseño

❖ Modularidad

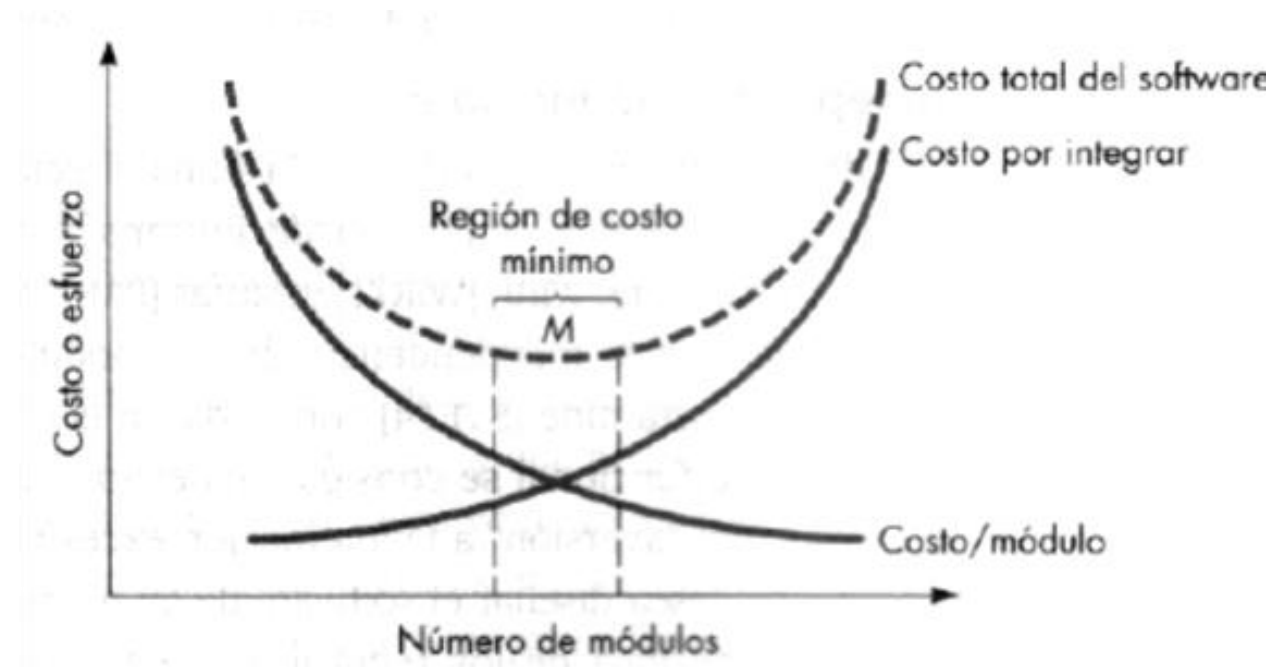
El software se divide en componentes nombrados y abordados por separado, llamados frecuentemente módulos, que se integran para satisfacer los requisitos del problema.

17



Conceptos de Diseño

❖ ¿Cuántos módulos tiene que tener un programa?



18

Conceptos de Diseño

❖ Ocultamiento de información

La información que está dentro un módulo es inaccesible a otros que no la necesiten.



19

Conceptos de Diseño

Independencia Funcional = *Modularidad + Abstracción + Ocultamiento de Información.*

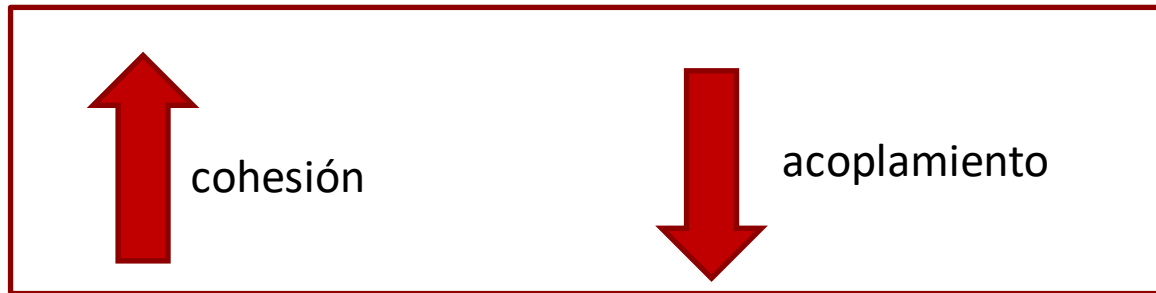


20



la cohesión y el acoplamiento entre los módulos

se busca..



Independencia Funcional

Cohesión (Coherente)

Medida de fuerza o relación funcional existente entre las sentencias o grupos de sentencias de un mismo módulo.

21



Alta cohesión



Baja cohesión

Independencia Funcional - Tipos de Cohesión

22

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

Las bajo

Cuando las sentencias tiene que ejecutarse en un orden específico

Cuando las sentencias de un módulo están relacionadas en el desarrollo de una única función

Coincidental

Lógica

Temporal

Procedimental

Comunicacional

Funcional

Cuando las sentencias llevan a cabo un conjunto de tareas que no están relacionadas o tienen

Cuando las sentencias se relacionan lógicamente

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

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.

23



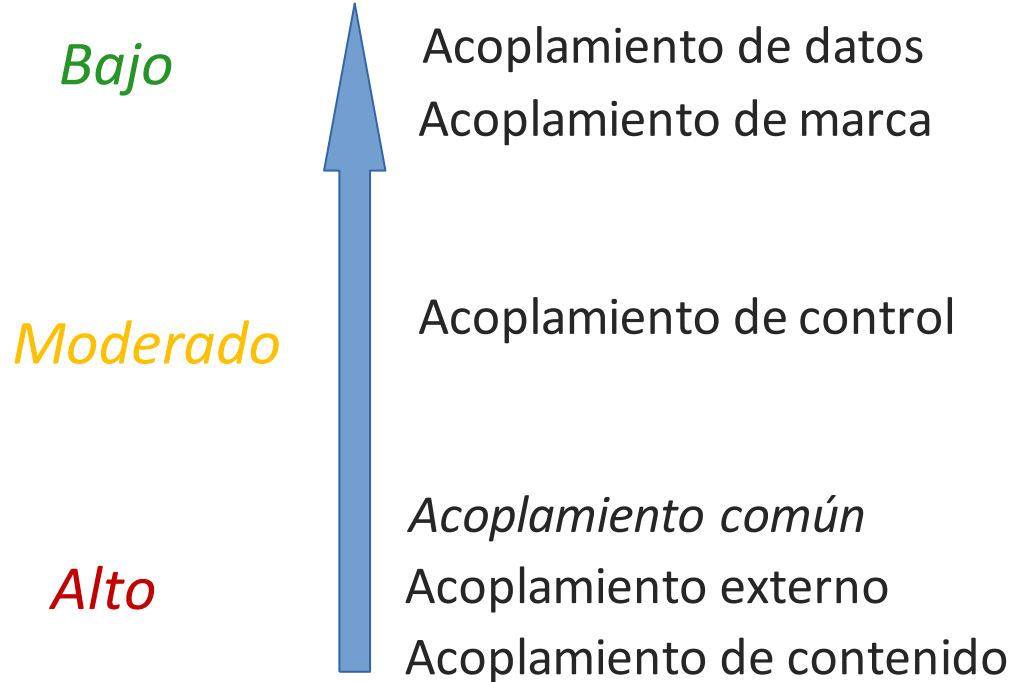
Bajo Acoplamiento



Alto Acoplamiento

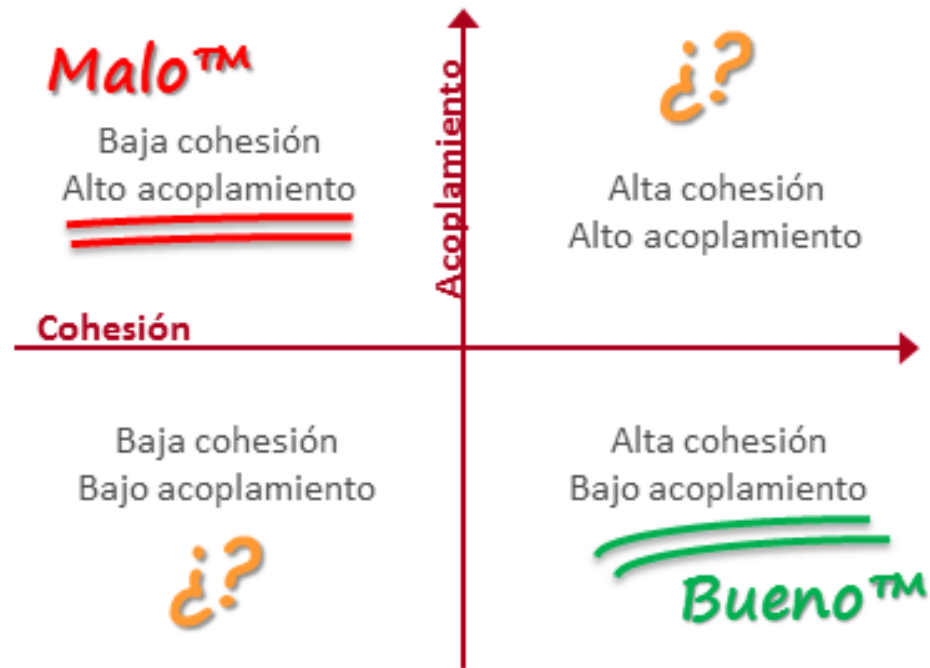
Independencia Funcional

Niveles de Acoplamiento



24

Independencia Funcional



25

Conceptos de Diseño

❖ Refinamiento

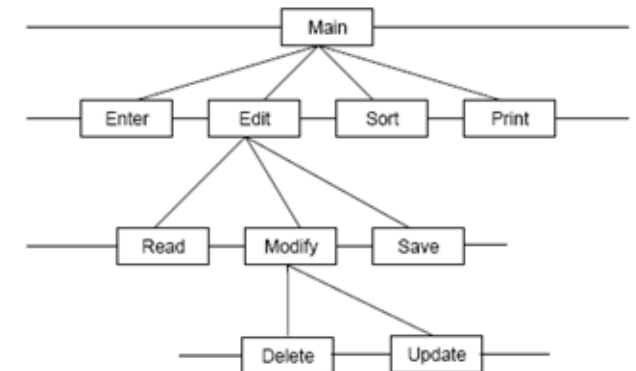
Se refina de manera sucesiva.

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.

26



Conceptos de Diseño

❖ Refabricación o rediseño (Refactoring)

Técnica de reorganización que simplifica el diseño de un componente sin cambiar su función o comportamiento.

27



Principios del modelado de diseño

1. El diseño debe poder rastrearse hasta el modelo de requerimientos
2. Considerar siempre la arquitectura del sistema que se va a crear
3. El diseño de los datos es tan importante como el diseño de funciones
4. Las interfaces deben diseñarse con cuidado
5. El diseño de interfaz debe ajustarse a las necesidades del usuario, haciendo énfasis en la facilidad de uso
6. El diseño a nivel de componentes debe ser funcionalmente independiente
7. Los componentes deben tener un bajo acoplamiento
8. Las representaciones de diseño debe entenderse fácilmente
9. El diseño debe desarrollarse de manera iterativa
10. La creación de un modelo de diseño no impide metodología ágil.

28