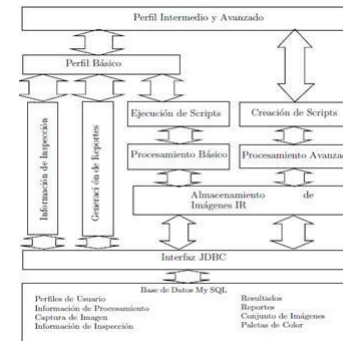


# Introducción a la Ingeniería del Diseño

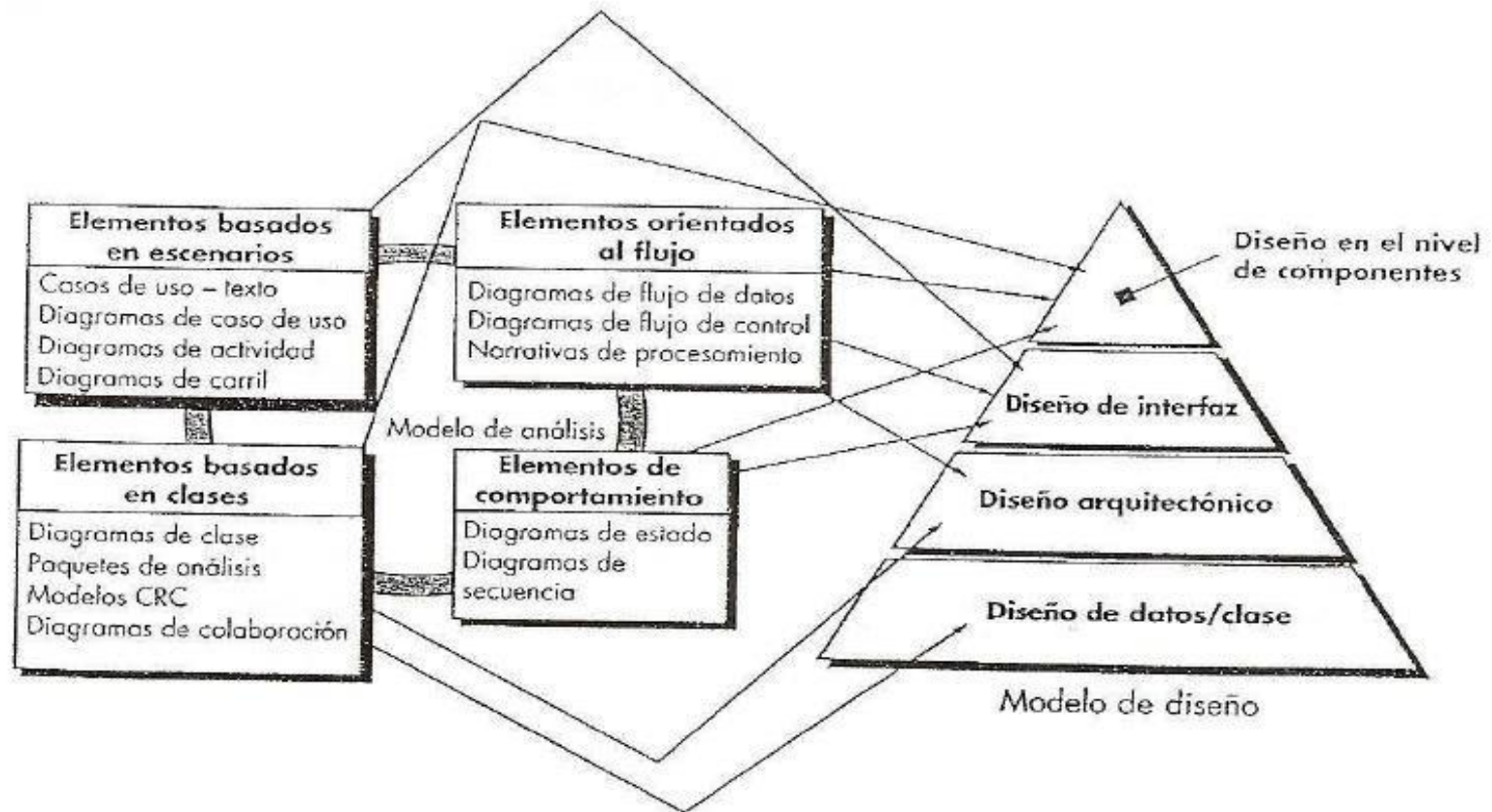
# Ingeniería del Diseño

- Es la actividad de la Ingeniería de Software donde manda la creatividad, donde los requisitos del cliente, las necesidades del negocio y las consideraciones técnicas se unen en la formulación de un producto software o sistema de calidad.
- El diseño crea una representación o modelo del software, pero a diferencia del modelo de análisis el modelo del diseño proporciona detalles acerca de la estructura de datos, las arquitecturas, las interfaces y los componentes del software que son necesarios para implementar el sistema.
- La meta de la ingeniería del diseño es producir un modelo de representación que muestre firmeza, comodidad y satisfacción de quien lo utilice.
- El diseñador debe practicar la diversificación y después la convergencia.



# Transformación del modelo de análisis al modelo del diseño

TRANSFORMACION DEL MODELO DE ANALISIS EN UN MODELO DE DISEÑO

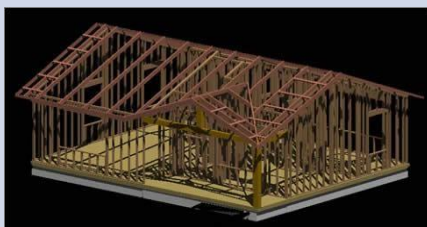


# Ingeniería del diseño

Para la construcción de una casa

En software

## Diseño base (estructural)



De madera



Metálica

## Diseño arquitectónico (estilo)



## Diseño de Interfaz

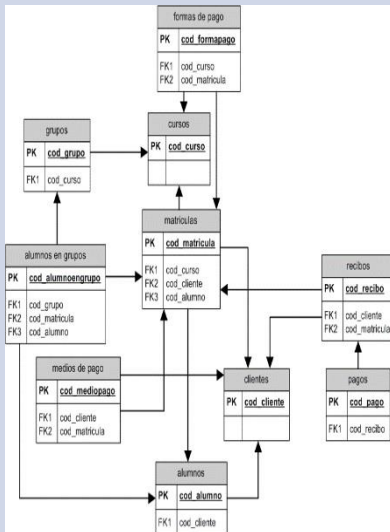


## Diseño de los componentes

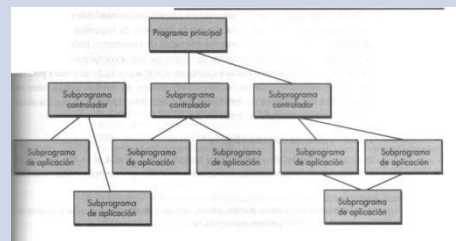
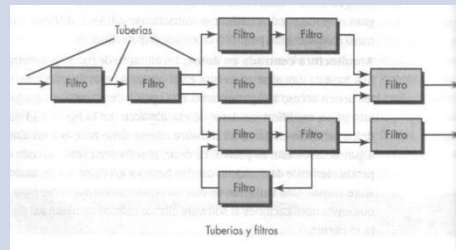
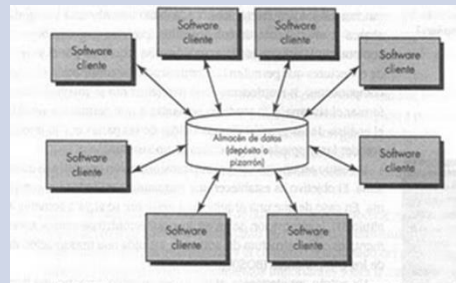


# Ingeniería del diseño

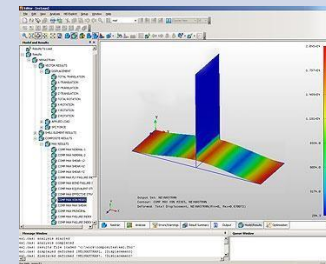
## Diseño de datos (base)



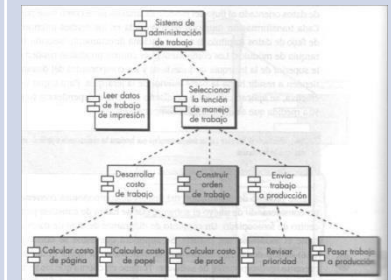
## Diseño arquitectónico (estilo, patrón)



## Diseño de Interfaz (IU, internas, externas)



## Diseño de los componentes



# Características de un buen diseño

- ☐ El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis, y debe ajustarse a todos los requisitos implícitos que desea el cliente.
- ☐ El diseño debe ser una guía legible y comprensible para quienes generan código y quienes realizan pruebas.
- ☐ El diseño debe proporcionar una imagen completa del software dando dirección a los dominios de datos, funcionales y de comportamiento desde una perspectiva de implementación.



# Modelo del Diseño

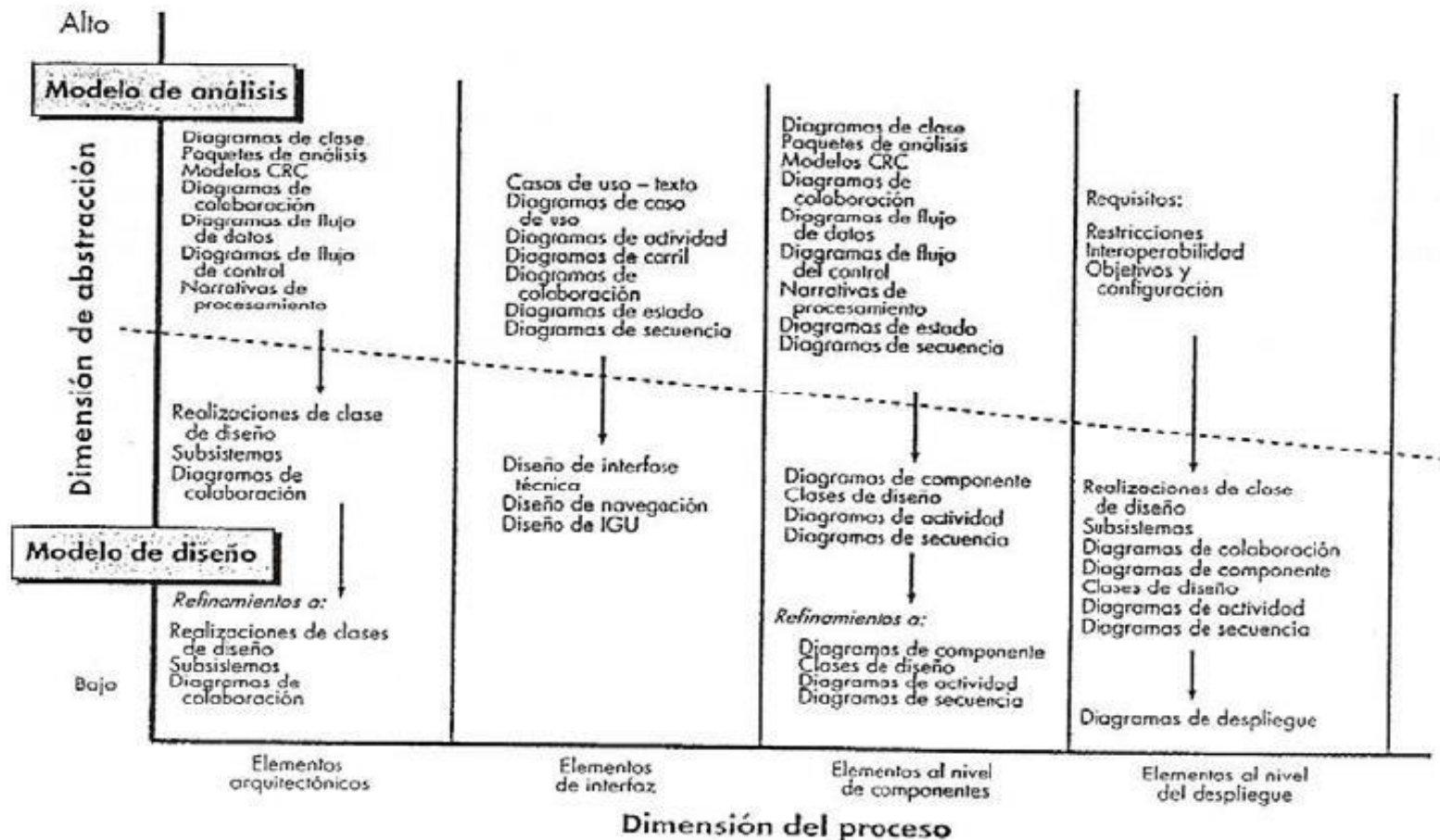
**Puede verse de 2 dimensiones diferentes:**

La dimensión del proceso que indica la evolución del modelo de diseño conforme se ejecutan las tareas de diseño como una parte del proceso del software. A través del proceso del diseño, la calidad en evolución de éste se evalúa con una serie de revisiones técnicas formales con revisiones de diseño.

La dimensión de abstracción que representa el grado de detalle a medida que cada elemento del modelo de análisis se transforma en un equivalente del diseño y después se refina de manera iterativa

# Dimensiones del modelo del diseño

Dimensiones del modelo de diseño.

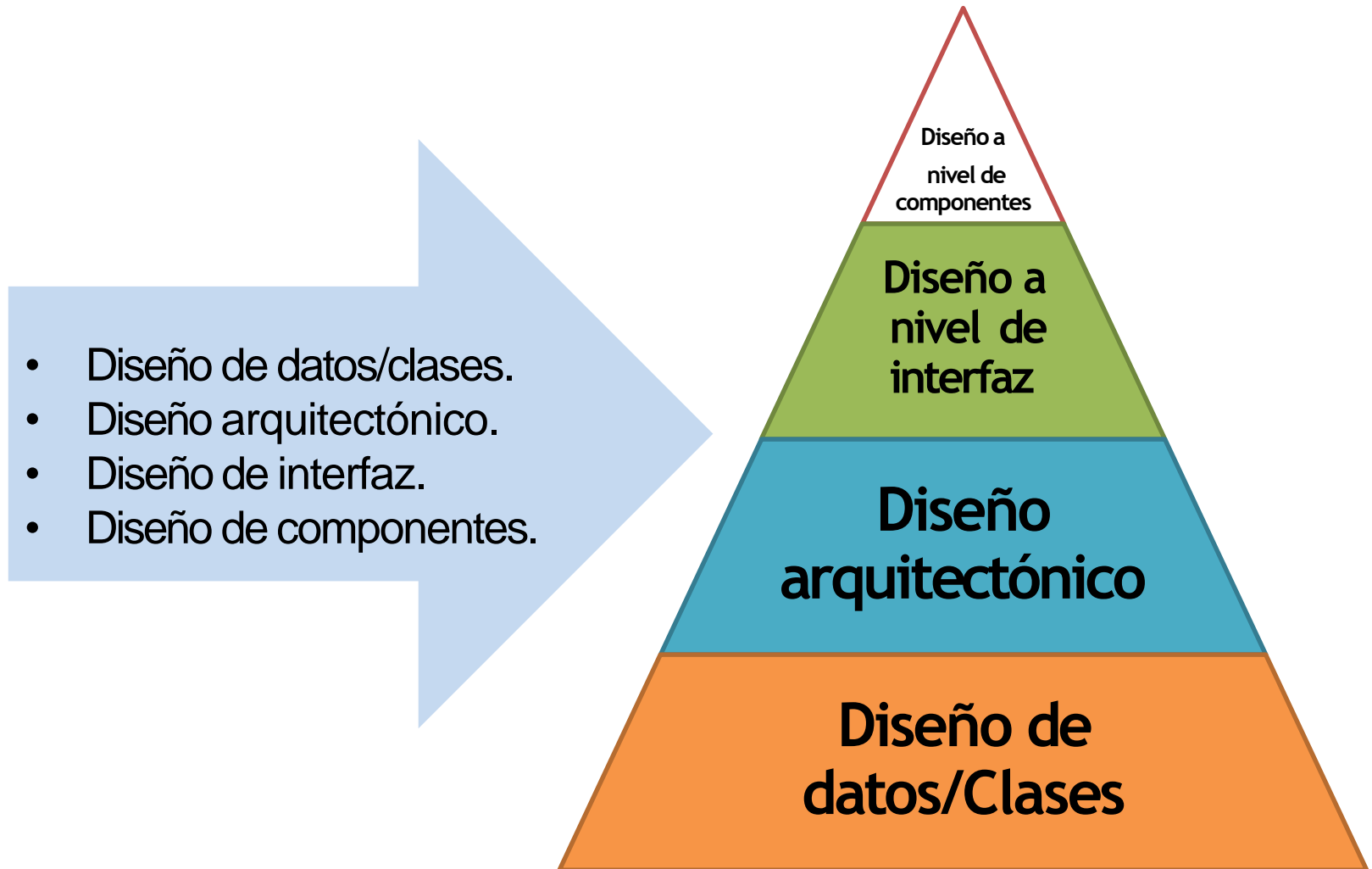




# Modelo del diseño

- ✓ Describe el **cómo** debe funcionar el sistema
- ✓ Describe la **estructura del sistema**
  - ✓ Qué **componentes** deben existir en el sistema?
  - ✓ Qué papel juega cada uno de los componentes
  - ✓ Cómo se **relacionan los componentes**
- ✓ Justifica las **decisiones** de diseño
- ✓ Emplea diagramas y notaciones formales
- ✓ Debe acomodarse a los **cambios** que se producirán
- ✓ El diseño **debe ser independientemente** del lenguaje de programación que se utilizará e independientemente del sistema operativo y la máquina a utilizar
- ✓ Debe ser una **guía para la implementación**

# Elementos (tareas) del diseño de software



# Diseño de clases/datos

- Transforma los modelos de análisis y clases en las clases de diseño y las estructuras de datos que se requieren para implementar el software.



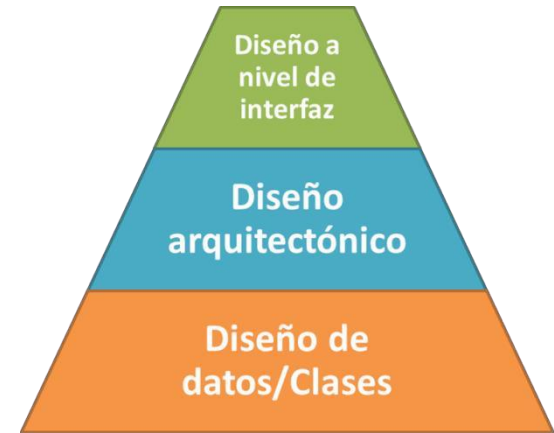
## Diseño arquitectónico.

- Diseño arquitectónico, define la relación entre los elementos estructurales más importantes del software, los estilos y patrones de diseño que pueden usarse para satisfacer los requisitos definidos por el sistema y las restricciones que lo afectan.



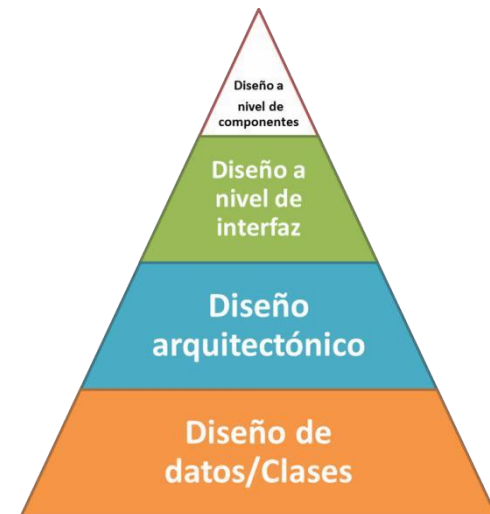
## ■ El diseño de la interfaz

Describe la forma en que el software se comunica con los sistemas que interactúan con él y con las personas que los utilizan.



## ■ El diseño al nivel de componentes

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



# Proceso y calidad del diseño

- El diseño del software es un proceso iterativo mediante el cual los requisitos se traducen en un “plano” para construir el software. El diseño se inicia con un grado alto de abstracción.
- A medida en que ocurren las iteraciones del diseño, un refinamiento subsiguiente conduce a representaciones del diseño a grados mucho más bajos de abstracción.
- A través del proceso del diseño, la calidad en evolución de éste se evalúa con una serie de revisiones técnicas formales o con revisiones de diseño.

# Atributos de calidad

- ❑ La funcionalidad
- ❑ La facilidad de uso (usabilidad)
- ❑ La confiabilidad
- ❑ El desempeño (rendimiento)
- ❑ La soportabilidad, mantenibilidad

Es importante puntualizar que estos atributos de calidad deben considerarse al comienzo del diseño.



# Algunos conceptos que se utilizan en la fase del diseño

## Modularidad

- El software se divide en componentes con nombres independientes y que es posible abordar en forma individual. Estos componentes llamados módulos se integran para satisfacer los requisitos del problema.

## Ocultación de información

- Los módulos deben especificarse y diseñarse de manera que la información (procedimiento y datos) que está dentro del modulo sea inaccesible para otros módulos que no necesiten esa información.

## Independencia funcional

- Se desea diseñar el software de tal manera que cada modulo aborde una subfunción específica de los requisitos y tenga una sola interfaz cuando se observe desde otras partes de la estructura del programa.

# Algunos conceptos que se utilizan en la fase del diseño

## Refinamiento

- Es una estrategia de diseño descendente, el desarrollo de un programa se realiza al refinar de manera sucesiva los niveles de detalle procedimentales.
- El refinamiento hace que el diseñador trabaje sobre el enunciado original y que proporcione más y más detalles conforme se realiza cada refinamiento sucesivo (elaboración).
- La abstracción y el refinamiento son conceptos complementarios. La abstracción le permite a un diseñador especificar procedimientos y datos sin considerar detalles de grado menor. El refinamiento ayuda al diseñador a revelar los detalles de grado menor mientras se realiza el diseño.

# Algunos conceptos que se utilizan en la fase del diseño

## Re fabricación

- Técnica de reorganización que simplifica el diseño (o código) de un componente sin cambiar su función o comportamiento.
- La refabricación es el proceso de cambiar un sistema de software de tal forma que no se altere el comportamiento externo de su código (diseño) y aún así se mejore su estructura interna.
- Cuando un software se refabrica, el diseño existente se examina en busca de redundancias, elementos de diseño inútiles, algoritmos innecesarios o insuficientes, estructuras de datos inapropiadas o construidas de manera incorrecta, o cualquier otra falla de diseño que se pueda corregir para lograr un mejor diseño.

# Algunos conceptos que se utilizan en la fase del diseño

## Clases de diseño

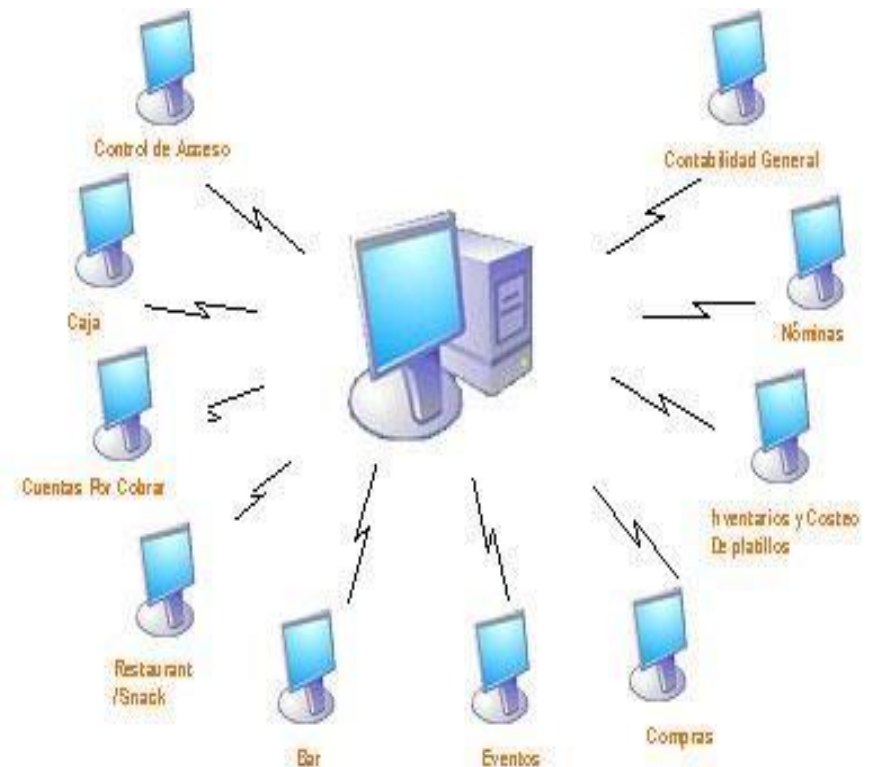
Conforme evoluciona el modelo de diseño, el equipo de software debe definir un conjunto de clases de diseño que 1) refine las clases de análisis al proporcionar detalles del diseño que permitirán la implementación de las clases, y 2) produzca un conjunto nuevo de clases de diseño que implementen una infraestructura de software para soportar la solución del negocio. Se sugieren cinco diferentes tipos de clases de diseño, cada uno representa una capa distinta de la arquitectura de diseño:

- Las clases de interfaz con el usuario
- Las clases del dominio de negocios
- Las clases de proceso
- Las clases persistentes
- Las clases de sistema

# Principios del Diseño

- El diseño debe ser modular

Debemos descomponer el sistema en módulos interconectados



# Cohesión

- ❑ Propiedad intra-modular
- ❑ Mide la *relación entre los contenidos de un módulo*
- ❑ Diversos grados: de fuerte a débil
- ❑ Objetivo en el diseño : *Maximizar* la cohesión

- **Función única de un componente**

- **La cohesión es la medida cualitativa de cuan estrechamente relacionados están los elementos internos de un módulo.**

- **Qué tan bien están funcionando los elementos internos de un módulo: función intra-modular**



## ....Coherencia

- Es la fuerza de las dependencias dentro de un módulo o subsistema.
- Si un subsistema o módulo contiene muchos objetos que están relacionados entre sí y realizan tareas similares, su coherencia es alta.
- Si un subsistema contiene varios objetos que no están relacionados su coherencia es baja.
- Lo deseable es una coherencia alta

# Acoplamiento

- Propiedad inter-modular
- Mide la *relación entre los módulos de un sistema*
- Diversos grados, de fuerte a débil
- Objetivo: *Minimizar* el acoplamiento

## ...Acoplamiento

- Es la fuerza de dependencias entre dos subsistemas o módulos .
- Si dos subsistemas o módulos están poco acoplados son relativamente independientes y, por lo tanto, las modificaciones a uno de los subsistemas o módulos tendrá poco impacto en el otro.
- Lo deseable es que los subsistemas o módulos deben estar lo menos acoplados posibles, es decir que sean lo mas independiente posible.
- Mas acoplamiento mas dependencia.

## Entonces...

- Hay un compromiso entre la coherencia y el acoplamiento porque:
  - Para aumentar la **coherencia** dentro de un subsistema o módulo podemos dividir un subsistema en otros más pequeños pero el **acoplamiento** aumenta.

# Ocultamiento de Información

Mostrar al usuario sólo lo que corresponde...

Interfaz : <i>Visible , Público</i>	Implementación: <i>Oculto, Privado</i>
<ul style="list-style-type: none"><li>❑ Parte <i>pública</i> y visible del módulo</li><li>❑ Determina qué servicios se ofrecen al usuario</li><li>❑ Indica el modo de uso (“instrucciones”)</li><li>❑ Orientado al usuario</li></ul>	<ul style="list-style-type: none"><li>❑ Parte <i>privada</i>, oculta del módulo</li><li>❑ Determina cómo funcionan los servicios ofrecidos</li><li>❑ Oculta detalles no relevantes para el usuario</li><li>❑ Sólo el ingeniero de software o implementador puede acceder a la implementación</li><li>❑ El usuario se mantiene aislado</li></ul>

# Ocultamiento de Información



**Visible > Interfaz**



**Implementación >  
Oculto, Privado**



## Pasos para el diseño...

- Examinar el modelo de datos y diseñar las estructuras de datos apropiadas para los objetos de datos y sus atributos.
- Por medio del modelo de análisis, seleccionar un estilo arquitectónico que sea apropiado para el software.
- Dividir el modelo de análisis en subsistemas o módulos de diseño y ubicar estos dentro de la arquitectura. Diseñar las interfaces del subsistema.
- Crear un conjunto de clases de diseño y componentes.