

# DISEÑO

**El** diseño de software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad

# Como Arquitecto debo responderme éstas preguntas!!!!

1. ¿Existe una arquitectura de aplicación genérica que pueda actuar como una plantilla para el sistema que se están diseñando?
2. ¿Cómo se distribuirá el sistema entre varios procesadores?
3. ¿Qué estilo o estilos arquitectónicos son apropiados para el sistema?
4. ¿Cuál será la aproximación fundamental utilizada para estructurar el sistema?

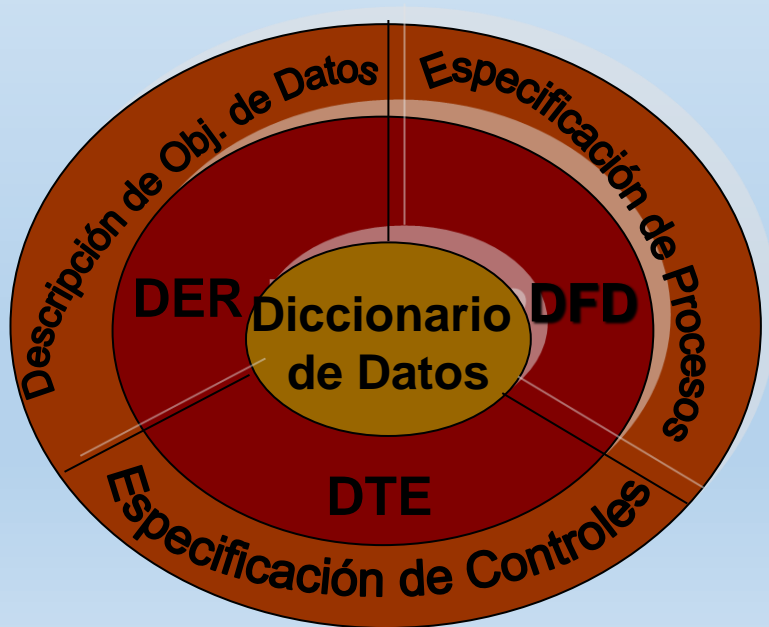
5. ¿Cómo se descompondrán en módulos las unidades estructurales del sistema?
6. ¿Qué estrategia se usará para controlar el funcionamiento de las unidades del sistema?
7. ¿Cómo se evaluará el diseño arquitectónico?
8. ¿Cómo debería documentarse la arquitectura del sistema?

# Elementos del Modelo de Análisis

El modelo de análisis debe lograr tres objetivos:

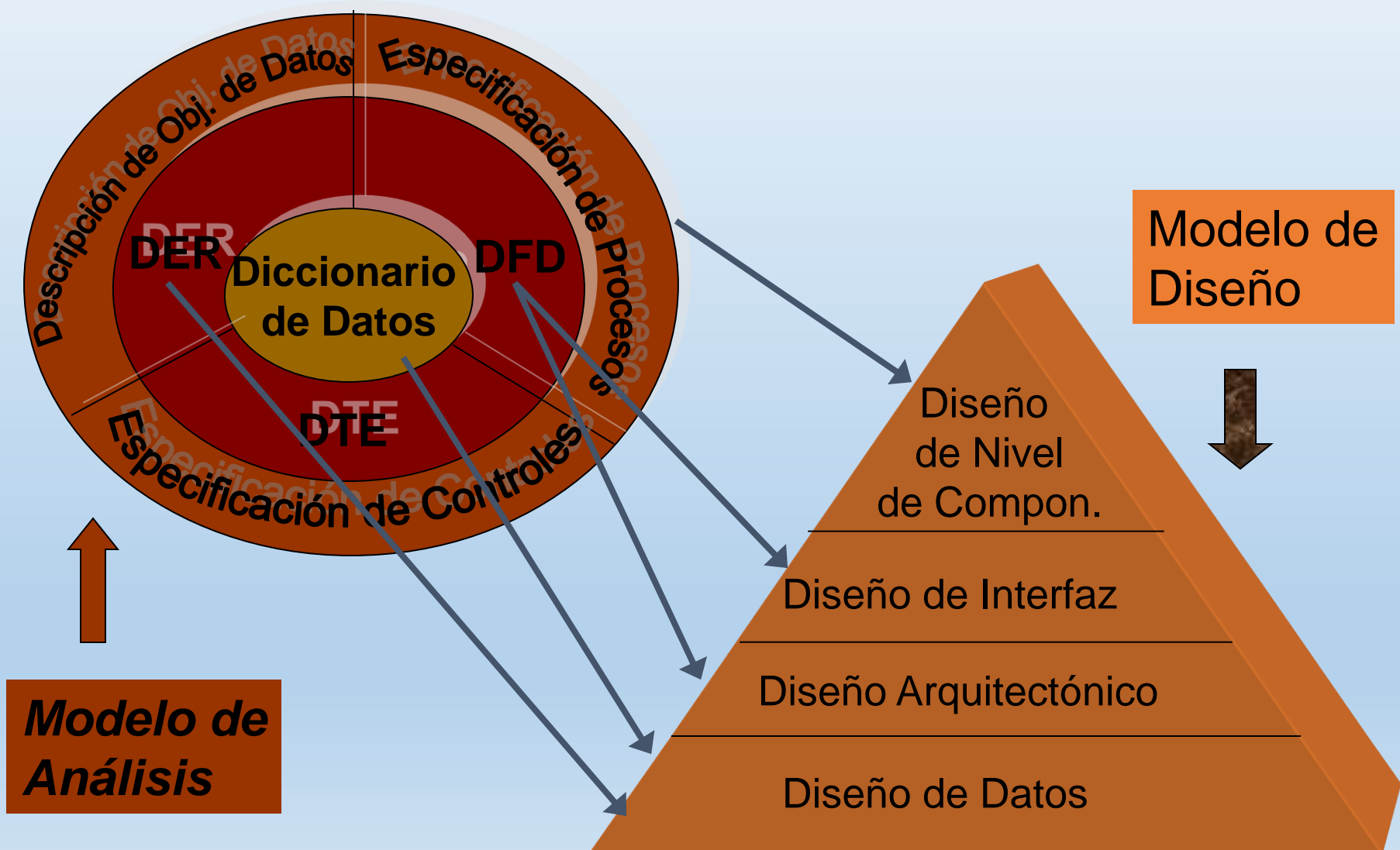
1. Describir lo que quiere el cliente
2. Establecer una base para el diseño de software
3. Definir un conjunto de requerimientos que pueda validar una vez que se construye el software.

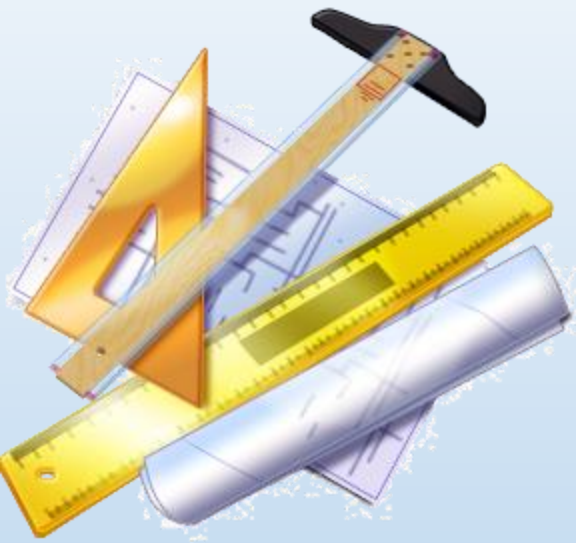
El modelo de análisis extraído de un diseño estructurado es el siguiente



c/u del modelo de análisis proporcionan la información necesaria para crear los cuatro modelos de diseño.

El flujo de información se muestra en el siguiente dibujo:





El diseño de datos transforma el modelo del dominio de información que se crea durante el análisis en las estructuras de datos que se necesitarán para implementar el software.

Los objetos de datos y las relaciones definidas en el DER (diagrama relación entidad) y el contenido de datos detallado que se representa en el diccionario de datos proporcionan la base de la actividad del diseño de datos.

El diseño arquitectónico define la relación entre los elementos estructurales principales del software, los patrones de diseño que se pueden utilizar para lograr los requisitos que se han definido para el sistema, y las restricciones que afectan a la manera en que se pueden aplicar los patrones de diseño arquitectónicos.



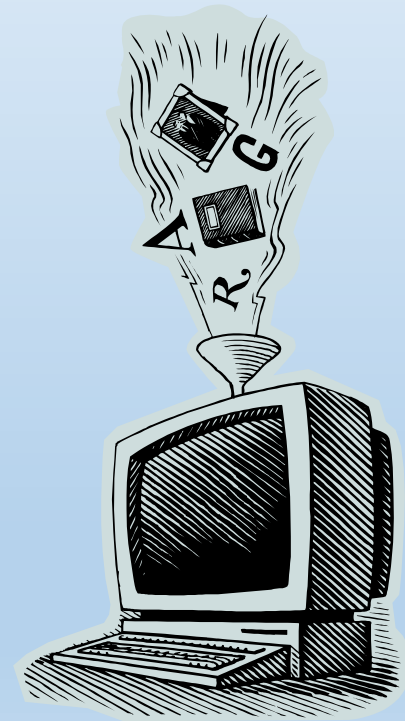


La representación del diseño arquitectónico puede derivarse de la especificación del sistema, del modelo de análisis y de la interacción del subsistema definido dentro del modelo de análisis.

El diseño de la interfaz describe la manera de comunicarse el software dentro de sí mismo, con sistemas que interoperan dentro de él y con las personas que lo utilizan.



Una interfaz implica un flujo de información (por ejemplo, datos y/o control) y un tipo específico de comportamiento. Por tanto, los diagramas de flujo de control y de datos proporcionan gran parte de la información que se requiere para el diseño de la interfaz.



El diseño a nivel de componentes transforma los elementos estructurales de la arquitectura del software en una descripción procedimental de los componentes del software.

La información que se obtiene de Especificación de Procesos, Especificación de Controles y de DTE sirve como base para el diseño de los componentes.

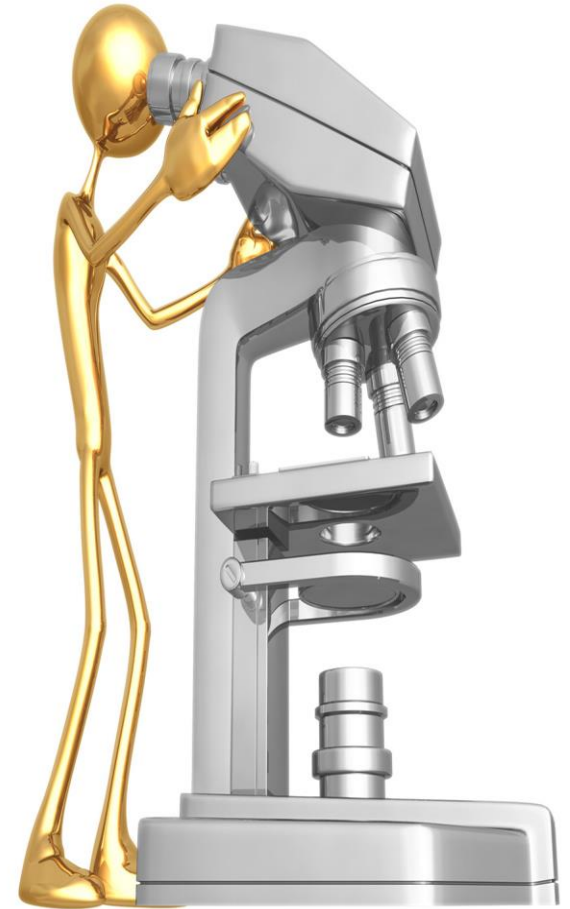


La importancia del diseño del software se puede describir con una sola palabra “**CALIDAD**”.

El diseño es el lugar en donde se fomentará la calidad en la ingeniería del software.

El diseño proporciona las representaciones del software que se pueden evaluar en cuanto a calidad.

Sin un diseño, corremos el riesgo de construir un sistema inestable.



# Veán el siguiente caso:



**La escena:** El cubículo de Jamie, cuando el equipo se prepara para traducir a diseño los requerimientos.

**Participantes:** Jamie, Vinod y Ed, miembros del equipo de ingeniería de software para CasaSegura.

**La conversación:**

**Jamie:** Ustedes saben, Doug [el gerente del equipo] está obsesionado con el diseño. Tengo que ser honesto, lo que realmente amo es codificar. Denme C++ o Java y soy feliz.

**Ed:** No... te gusta diseñar.

**Jamie:** No me estás escuchando; codificar es lo mío.

**Vinod:** Creo que Ed quiere decir que en realidad no es codificar lo que te gusta; te gusta diseñar y expresarlo en código. El código es el lenguaje que usas para representar el diseño.

**Jamie:** ¿Y qué tiene de malo?

**Vinod:** El nivel de abstracción.

**Jamie:** ¿Qué?

**Ed:** Un lenguaje de programación es bueno para representar detalles tales como estructuras de datos y algoritmos, pero no es tan bueno para representar la arquitectura o la colaboración entre componentes... algo así.

**Vinod:** Y una arquitectura complicada arruina al mejor código.

**Jamie (piensa unos momentos):** Entonces, dicen que no puede representarse la arquitectura con código... eso no es cierto.

**Vinod:** Claro que es posible implicar la arquitectura con el código, pero en la mayor parte de lenguajes de programación, es muy difícil lograr un panorama rápido y amplio de la arquitectura con el análisis del código.

**Ed:** Y eso es lo que queremos hacer antes de empezar a codificar.

**Jamie:** Está bien, tal vez diseñar y codificar sean cosas distintas, pero aún así me gusta más codificar.

# EL PROCESO DE DISEÑO

- El diseño del software es un proceso iterativo mediante el cual los requisitos se traducen en un «plano» .
- Se representa desde un nivel mas global (abstracto) y va hacia lo más particular (comportamiento, funcionales y de datos) a medida que ocurren iteraciones el nivel de abstracción va disminuyendo.



# DISEÑO Y CALIDAD

La calidad de la evolución del diseño se evalúa con una serie de revisiones técnicas formales:

- El diseño deberá implementar todos los requerimientos explícitos del modelo de análisis y deberán ajustarse a todos los requerimientos implícitos que desea el cliente.
- El diseño proporcionará una imagen completa del software (teniendo en cuenta funciones, comportamiento, y datos) desde una perspectiva de implementación.
- El diseño es una guía para los que generan código, testing y datos desde una perspectiva de implementación.



# CRITERIOS DE CALIDAD DEL DISEÑO

Con el fin de evaluar la calidad de una representación de diseño, deberán establecerse los criterios técnicos para un buen diseño.

Un diseño deberá presentar una estructura arquitectónica que:

1. se haya creado mediante Patrones de diseño reconocibles,
2. que esté formada por componentes que exhiban características de buen diseño
3. Un diseño deberá contener distintas representaciones de datos, arquitectura, interfaces y componentes (módulos).
4. Deberá conducir interfaces fácil de conectarse entre módulos y el entorno.
5. Un diseño deberá conducir a estructuras de datos adecuadas para los objetos que se van a implementar.
6. Deberá derivarse mediante un método repetitivo y controlado por la información obtenida en el análisis.



# PRINCIPIOS DEL DISEÑO

El diseño de un software es tanto un:

**PROCESO**

Como un

**MODELO**



Secuencia de pasos  
que hacen posible que  
el diseñador describa  
todas los aspectos del  
software que se va  
construir.



Planos de la casa

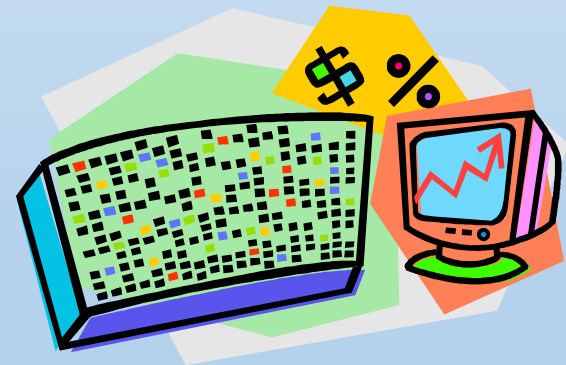


# Los principios básicos de diseño hacen posible que el ingeniero del software navegue por el proceso de diseño.

- En el proceso de diseño no deberá utilizarse «orejeras». (enfoques alternativos)
- El diseño deberá poderse rastrear hasta el modelo de análisis.
- El diseño no deberá inventar nada que ya esté inventado.
  - El diseño deberá «minimizar la distancia intelectual entre el software y el problema como si de la misma vida real se tratara.
  - El diseño deberá presentar uniformidad e integración.
- El diseño deberá estructurarse para admitir cambios.
- El diseño deberá estructurarse para degradarse poco a poco, no deberá nunca explotar como una bomba.
- El diseño no es escribir código y escribir código no es diseñar.
- El diseño deberá evaluarse en función de la calidad mientras se va creando, no después de terminarlo.
- El diseño deberá revisarse para minimizar los errores conceptuales (semánticos).

**Si aplicamos todo lo descripto anteriormente es fácil ver la calidad del software, que muestra los factores de calidad tanto internos como externos.**

Los FACTORES DE CALIDAD EXTERNOS son esas propiedades del software que pueden ser observadas fácilmente por los usuarios (por ejemplo, velocidad, fiabilidad, grado de corrección, usabilidad).



Los **FACTORES DE CALIDAD INTERNOS** tienen importancia para los ingenieros del software. Desde una perspectiva técnica conducen a un diseño de calidad alta. Para lograr los factores de calidad internos, el diseñador deberá comprender los conceptos de diseño básicos.

# Abstracción **CONCEPTOS DEL DISEÑO**

Refinamiento

Modularidad

Ocultamiento de la informacion

Independencia funcional

Aspectos

Rediseno

Arquitectura del software

Jerarquía de control

Patrones

Division de Problemas