

Ingeniería del Software

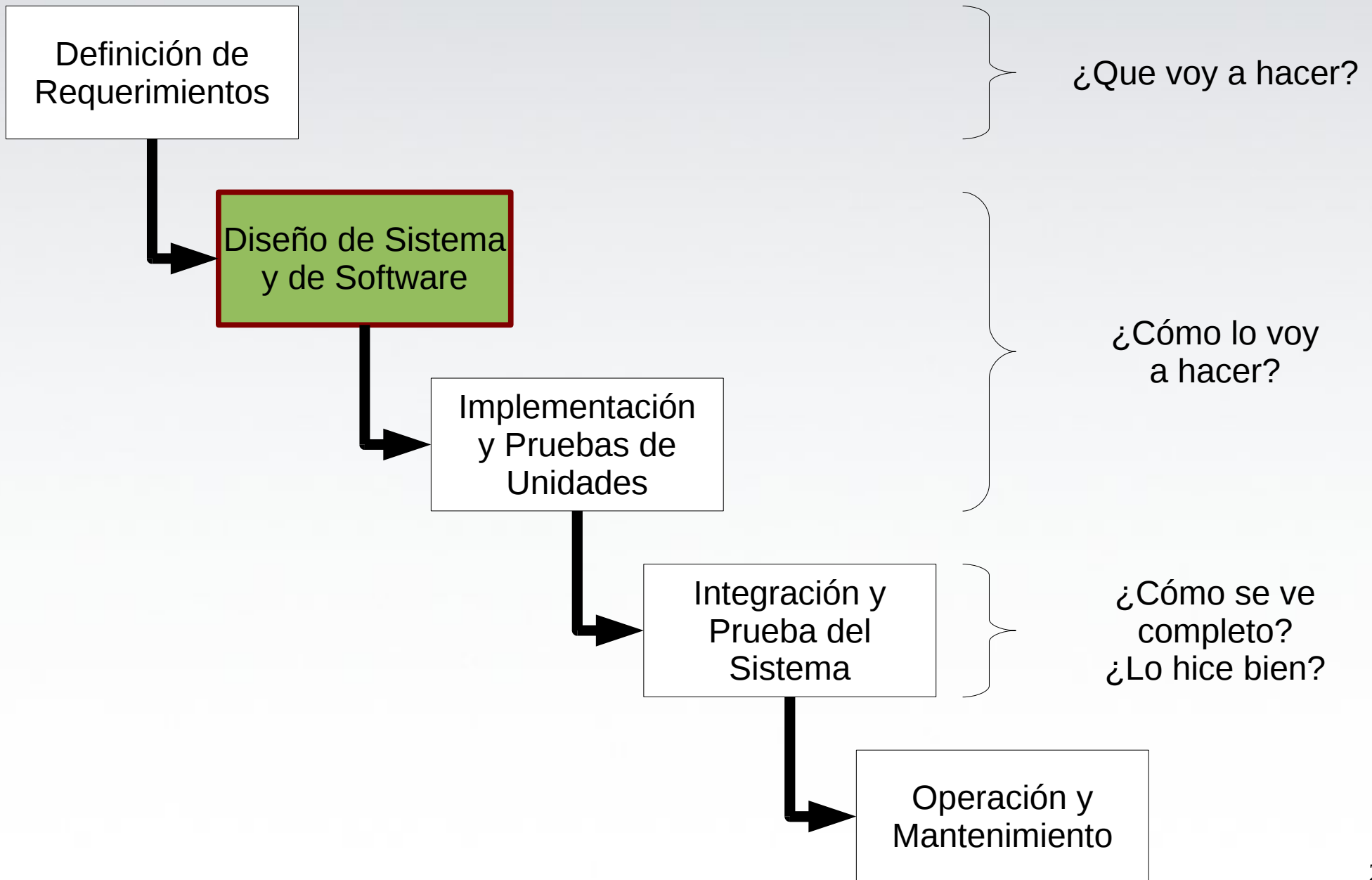
Diseño de Software

Universidad de los Andes

Demián Gutierrez

Mayo 2011

¿Qué es Diseño de Software?



Diseño es el **proceso creativo** de transformar un problema en una solución. La **descripción de esa solución** es, también, denominada diseño

Pfleeger, 1998

El diseño de software es el proceso de **concebir (modelar) y especificar** los detalles de como el sistema cumplirá las **especificaciones de requerimientos** establecidas en el análisis

En Ingeniería del Software, el diseño es definir como se transformará el que en el como

¿Qué es Diseño de Software?

Diseño = Proceso Creativo

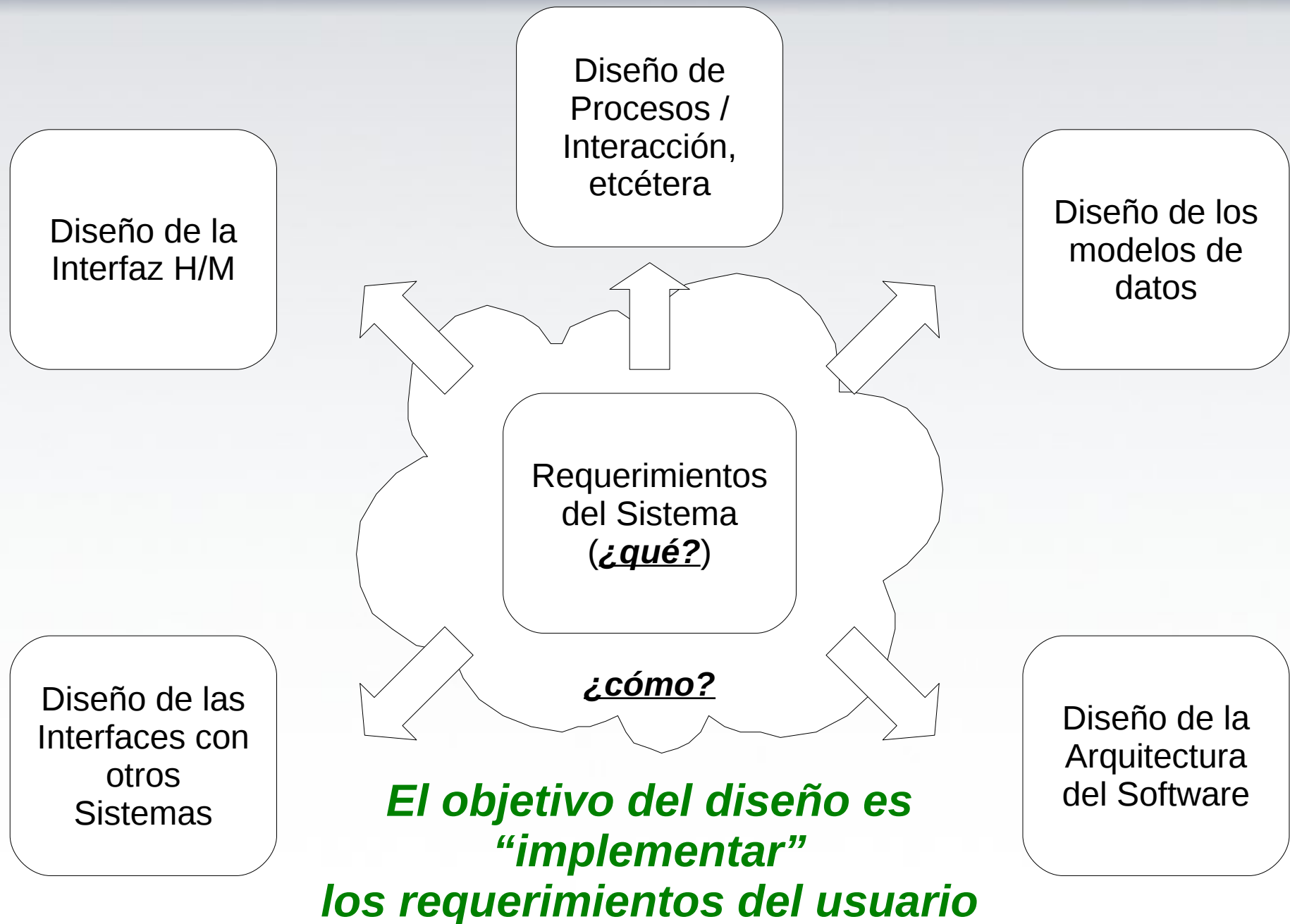
Diseño = Solución

No hay UNA única solución

No existe una solución completamente óptima
Sólo hay soluciones buenas, mediocres, malas...

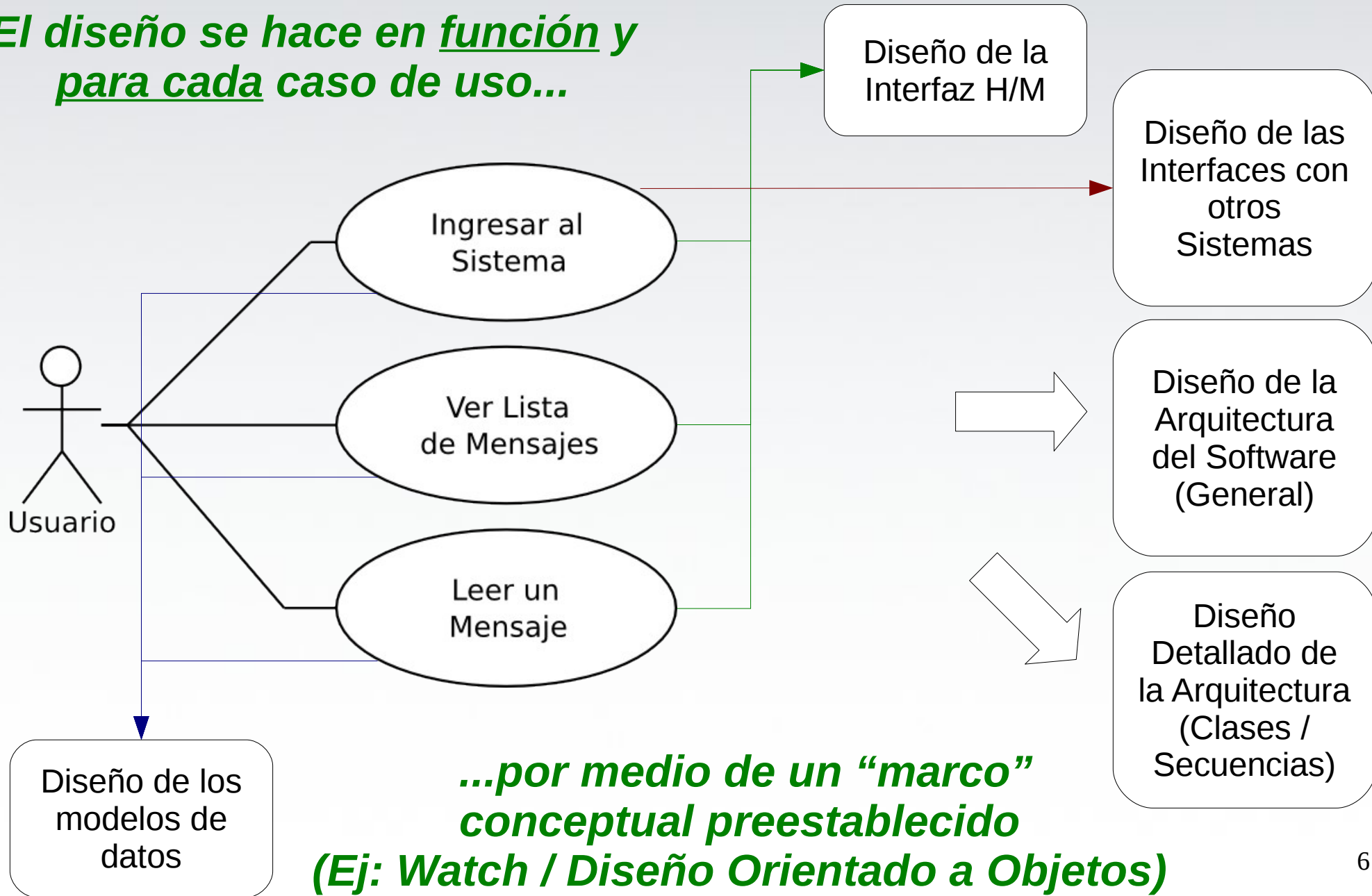
La evaluación y elección depende del **cliente**, de los **requerimientos**, del **criterio del diseñador**, del **contexto**, etcétera

¿Qué es Diseño de Software?



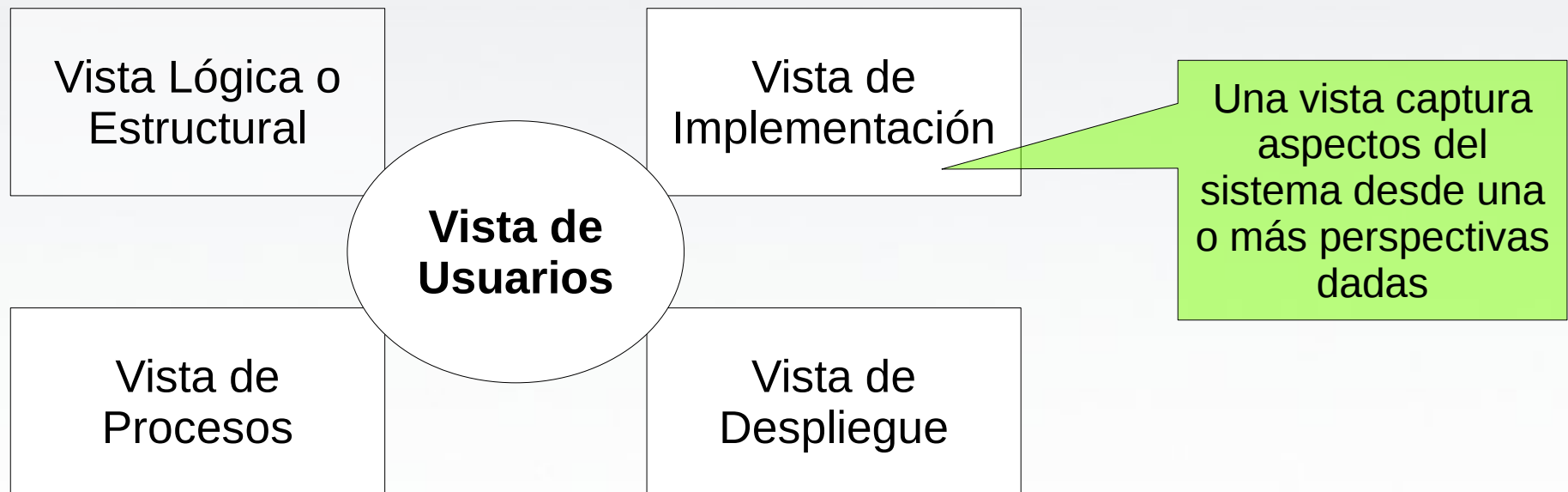
¿Qué es Diseño de Software?

El diseño se hace en función y para cada caso de uso...



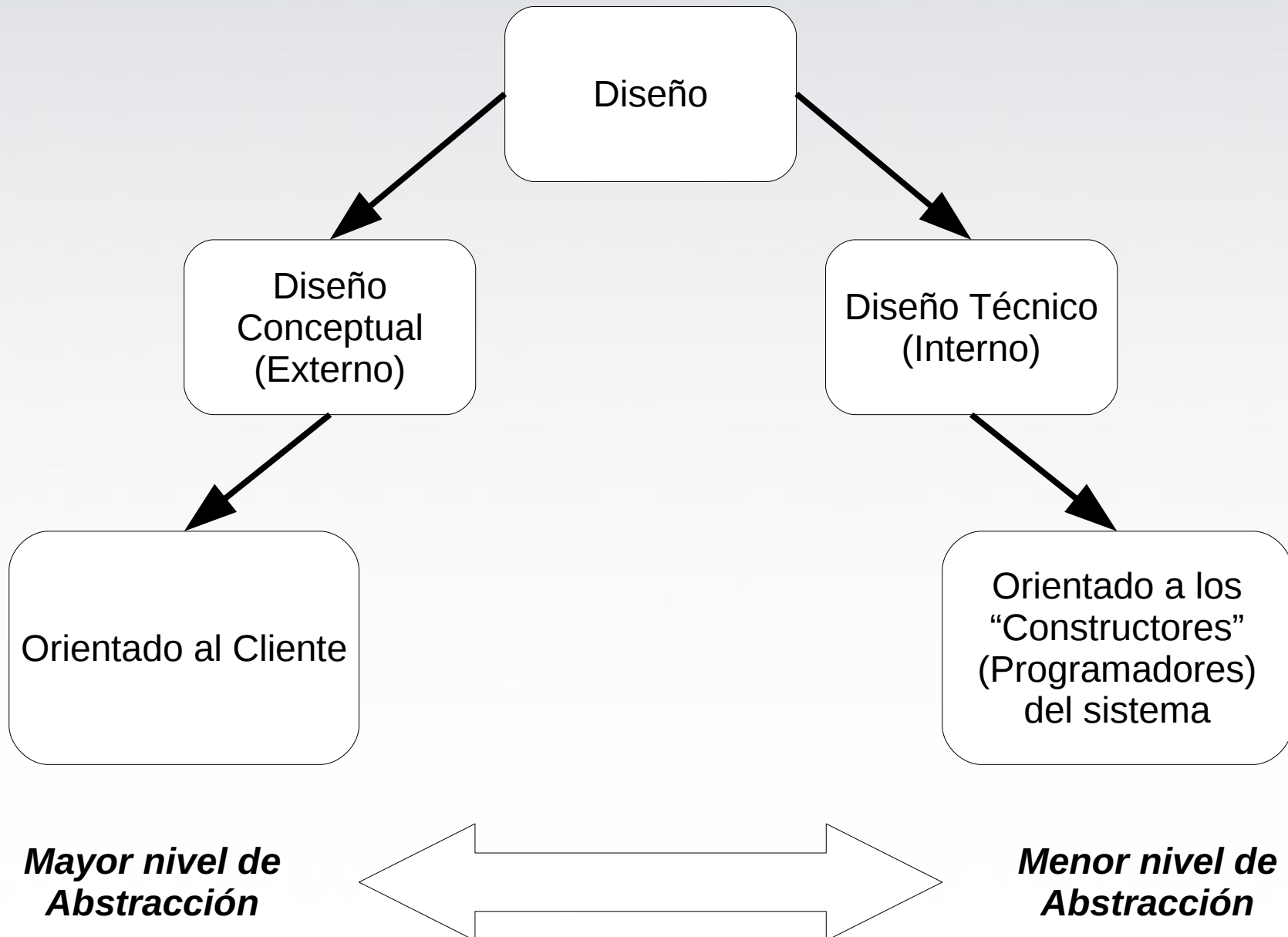
Usando UML, el sistema también puede ser **diseñado** en términos de **vistas**

Introducido por Philippe Kruchten en 1995



Es un enfoque que permite ver distintas partes (facetas) de la **arquitectura** de un sistema por separado

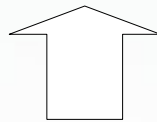
¿Diseño de Software Conceptual / Técnico?



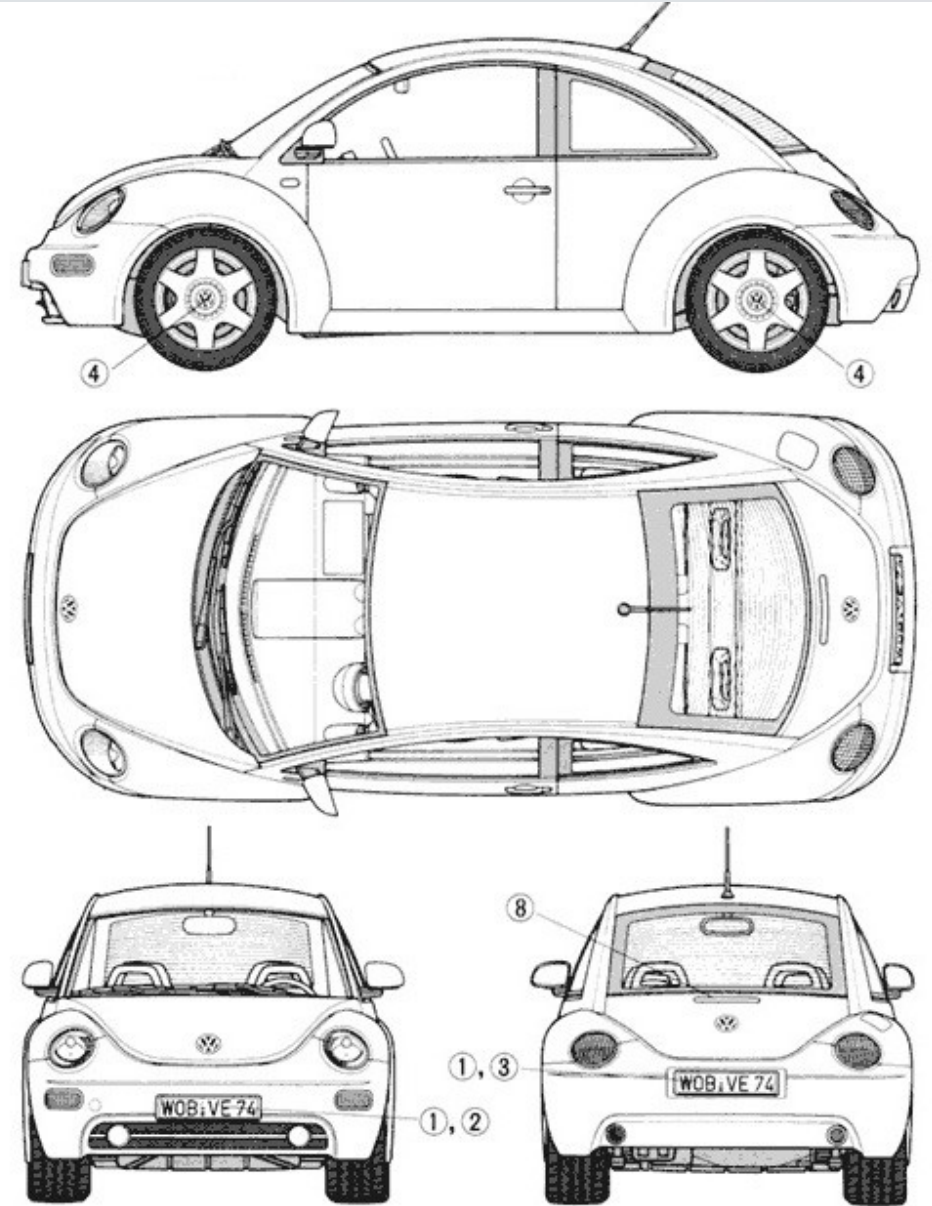
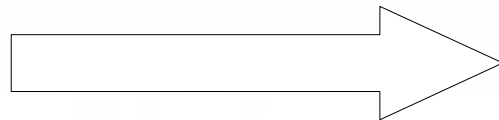
¿Diseño de Software Conceptual / Técnico?



Conceptual



Técnico



***Definir la estructura
general del sistema
programado***

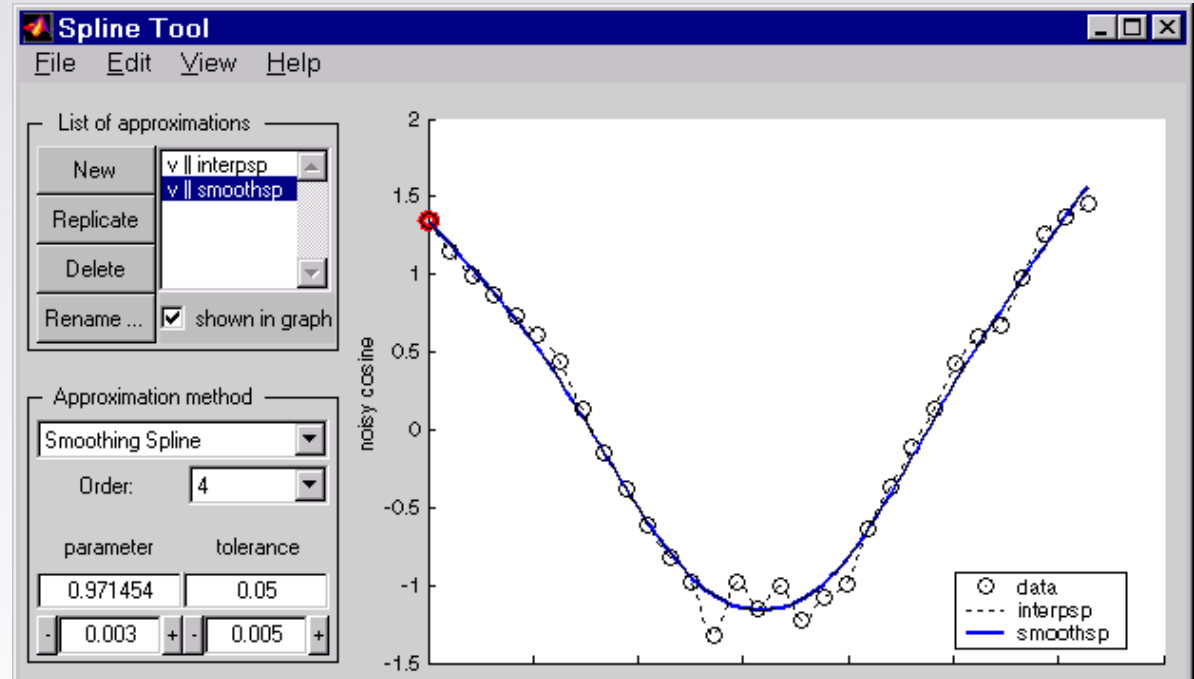


Describir las funciones que deberá ejecutar el sistema bajo el ambiente operativo establecido en los requerimientos

¿El “**qué**”? ¿Casos de Uso?

Resulta que desde cierto punto de vista, los casos de uso también forman parte el diseño

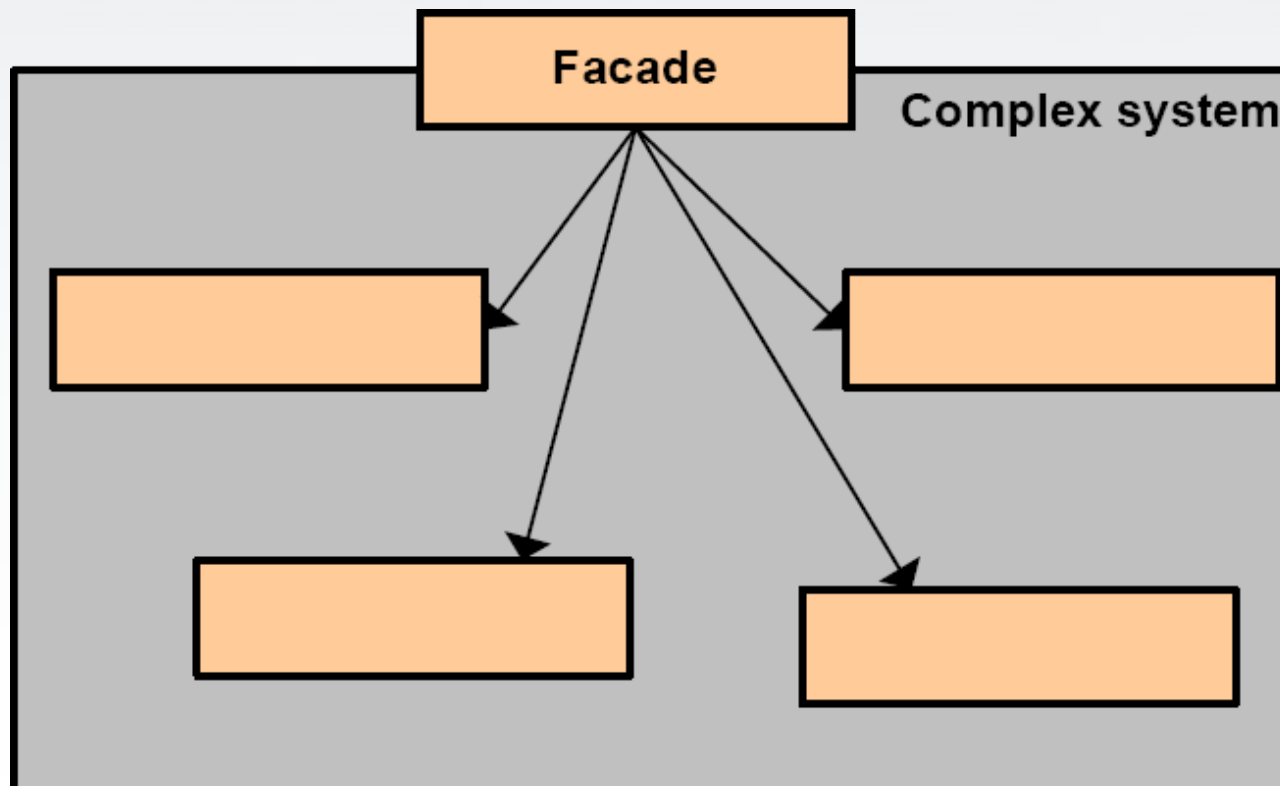
Diseñar la Interfaz
Usuario / Sistema,
incluyendo la entrada
de datos y salida de
información



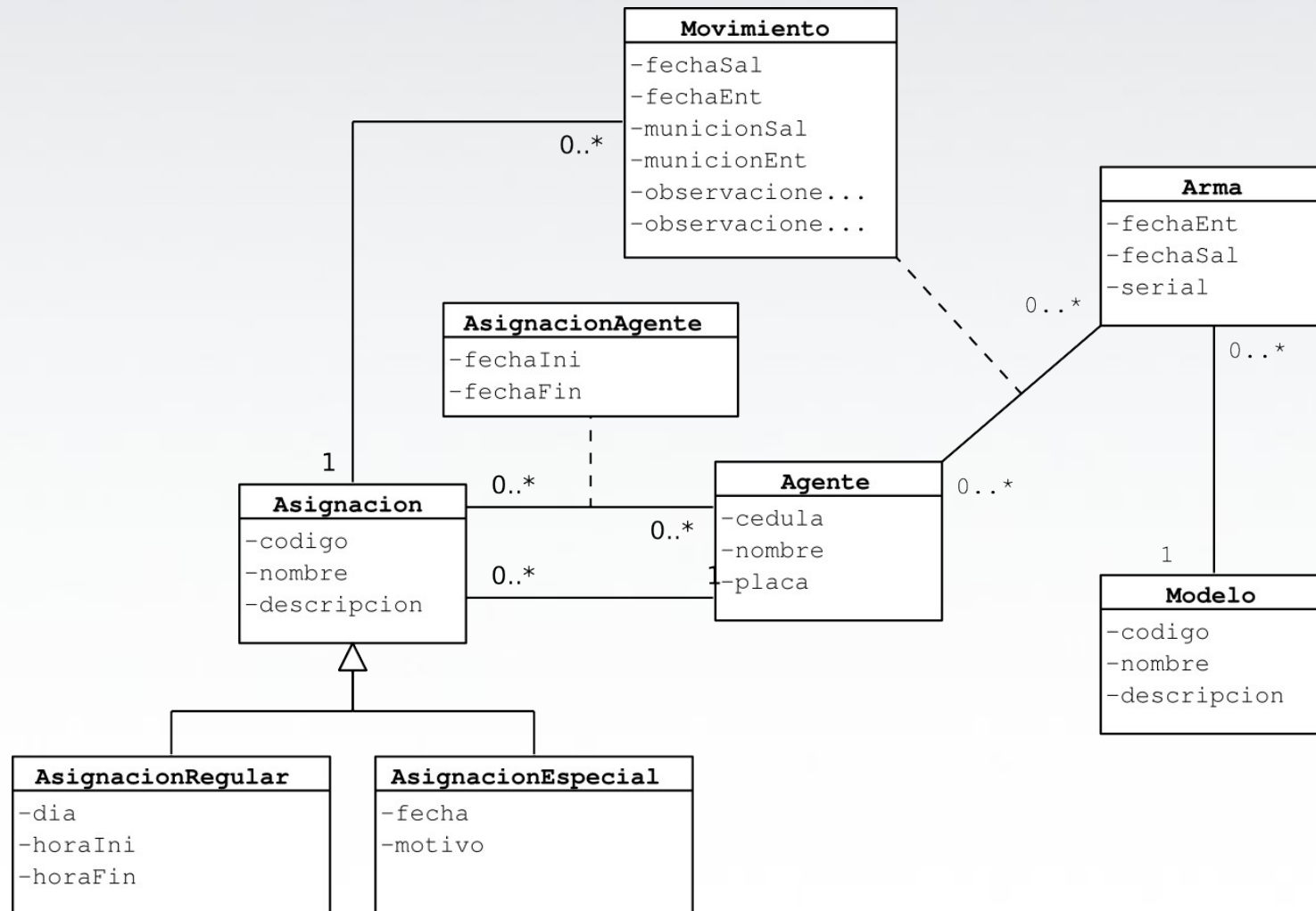
Describir las fuentes de los datos y sus procesos de
transformación

Establecer los atributos de calidad de diseño que deberá
satisfacer el sistema

Diseño de la Arquitectura: Usando ***Estilos Arquitectónicos, Patrones de Diseño, Frameworks*** (marcos)



Diseño de Archivos o Bases de Datos



La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus ***componentes***, las ***relaciones*** entre ellos y el ***ambiente y los principios que orientan*** su diseño y evolución.

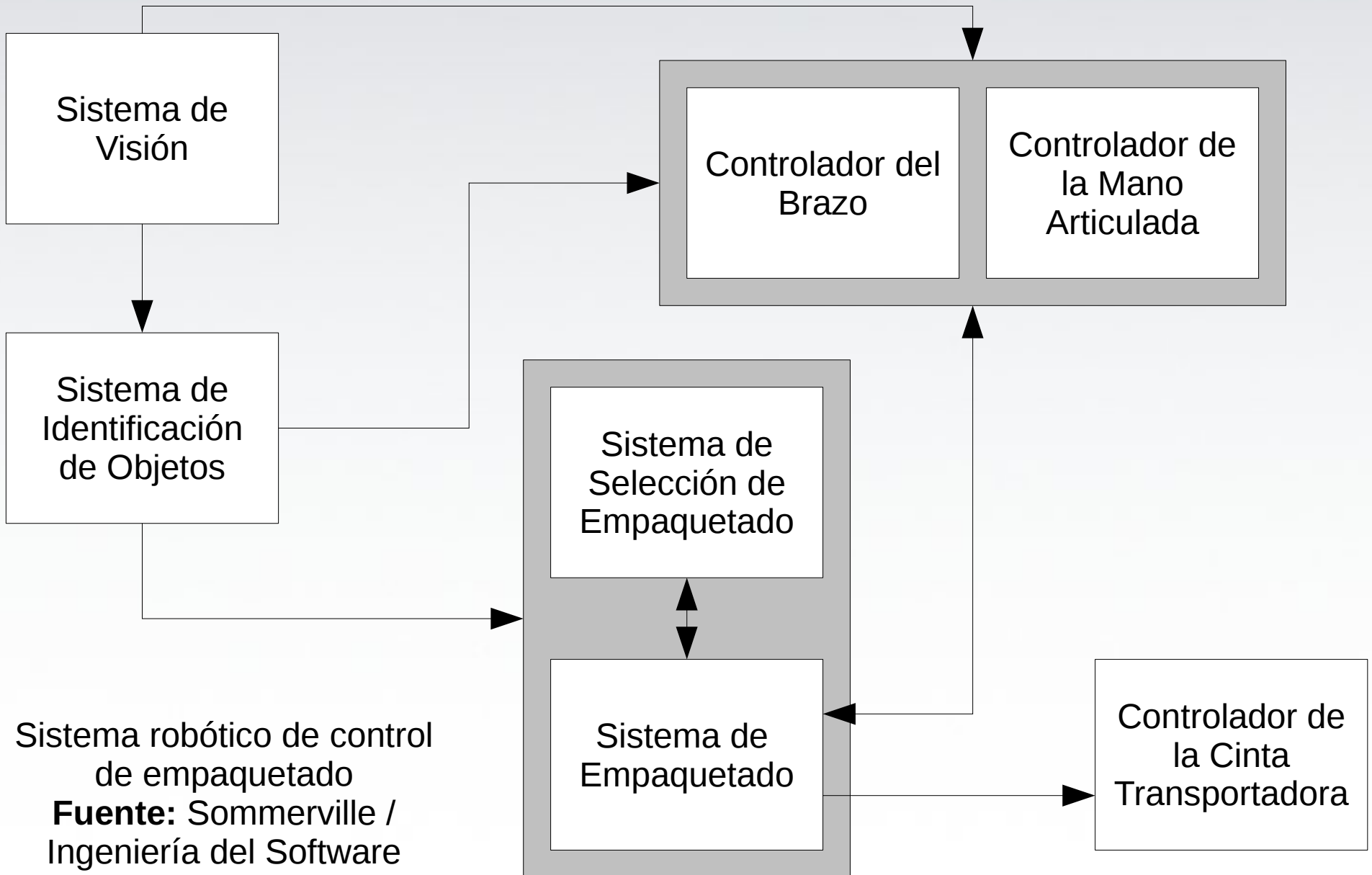
IEEE 1471-2000

La Arquitectura de Software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones

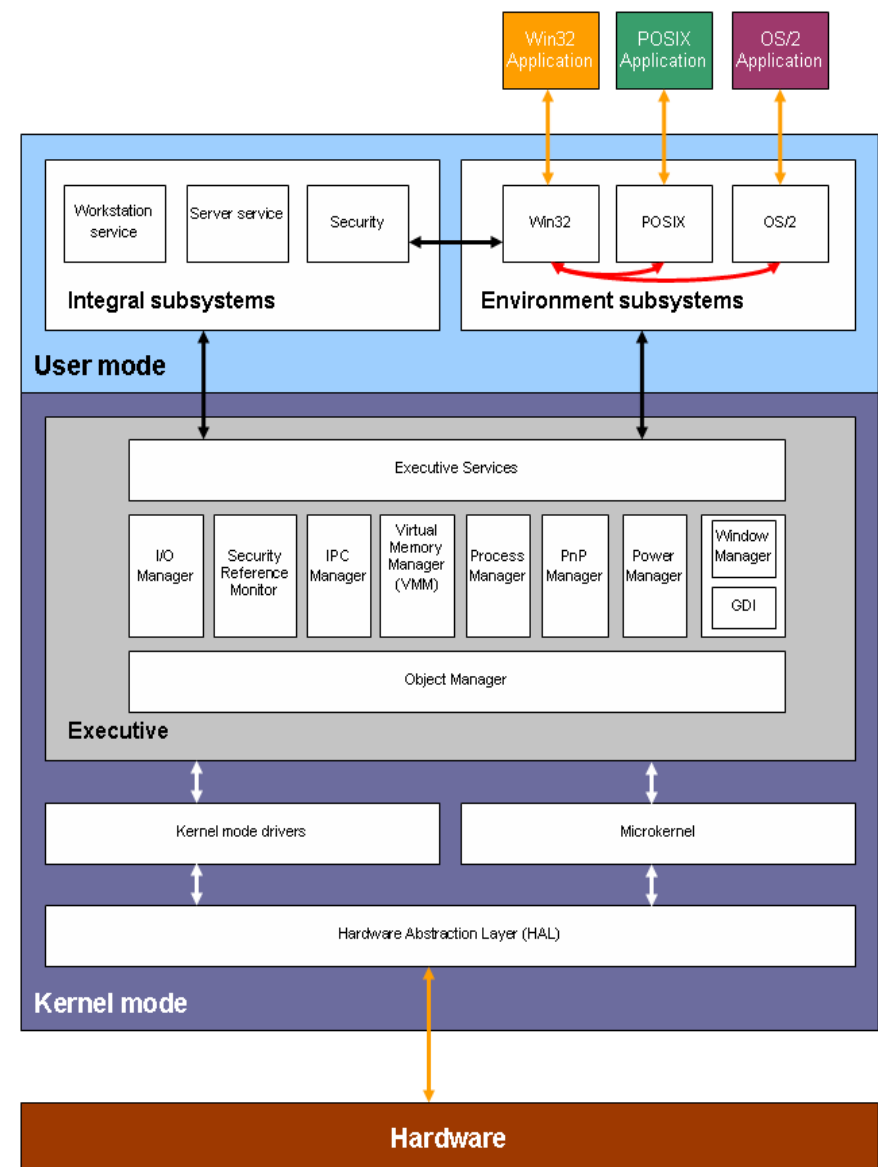
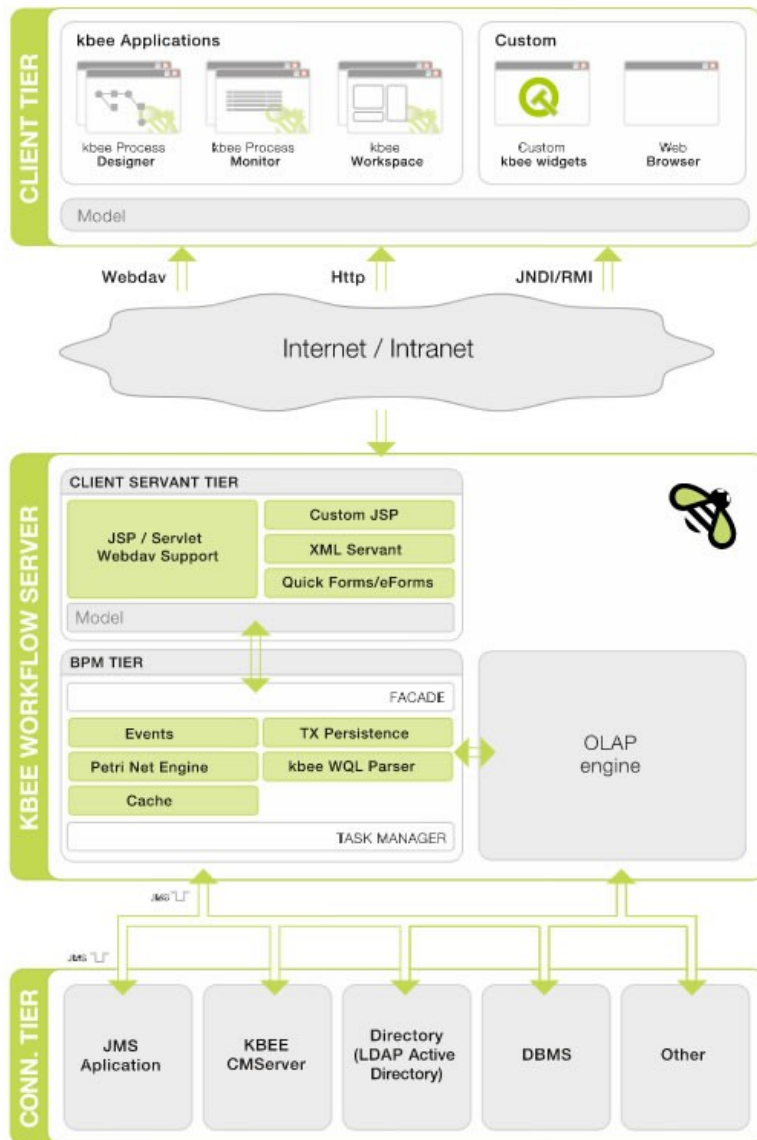
Paul Clements, 1996

La Arquitectura de un sistema define la ***división y estructura*** de un sistema en ***subsistemas*** y establece un ***marco de control y comunicación*** entre los distintos subsistemas

¿Arquitectura?



¿Arquitectura?



Estructura y componentes* del sistema ***desde el punto de vista del programador***

Jerarquía y funciones de cada componente* de software

Estructuras de datos empleadas por cada componente*

Flujos de datos entre los componentes*

**** También se puede ver desde el punto de vista de Clases / Objetos***

Rendimiento: Operaciones críticas en un pequeño número de subsistemas / reducción de comunicación entre subsistemas

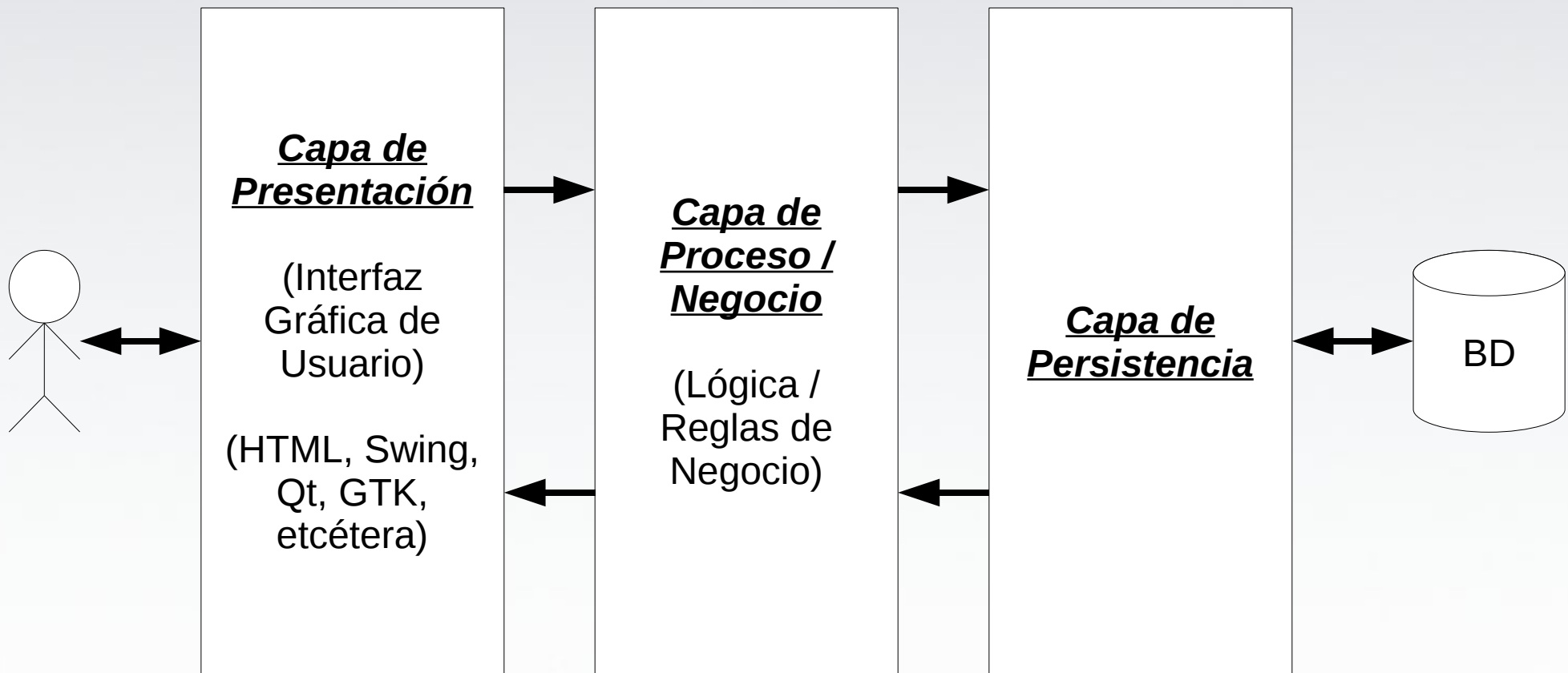
Protección: Es necesario proteger el acceso a ciertos recursos, se puede usar una arquitectura que esconda y limite el acceso y comunicación con los recursos a proteger

Seguridad: Centralizar las operaciones relacionadas con la seguridad en un subsistema (o en un conjunto pequeño de subsistemas) para reducir costos y desarrollar los mecanismos de acceso adecuados

Disponibilidad: Incluir componentes redundantes y permitir reemplazar componentes sin necesidad de detener el sistema

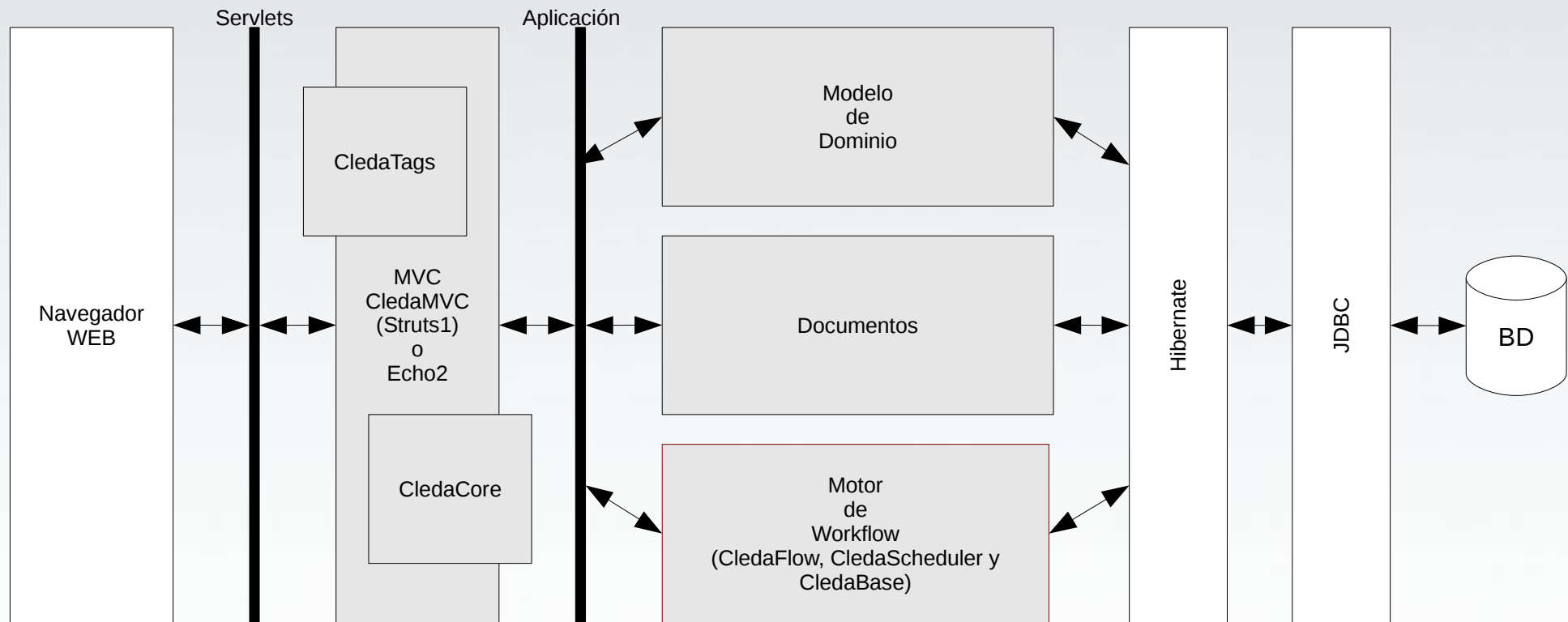
Mantenibilidad: Utilizar componentes independientes de grano fino que pueden modificarse con facilidad de forma independiente, separar productores de consumidores de información y evitar (o estandarizar) estructuras de datos compartidas

Otros...



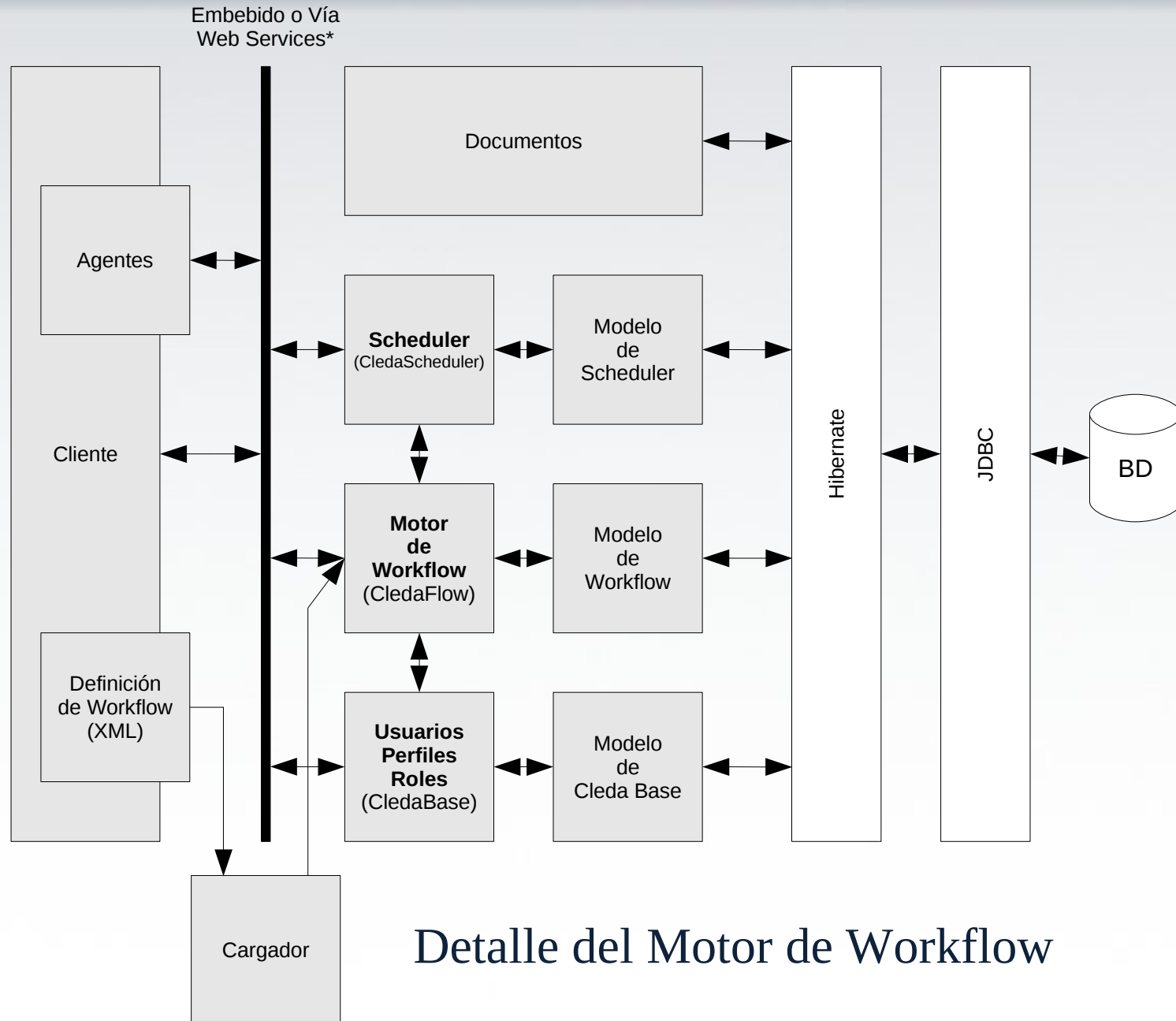
Arquitectura a 3 capas

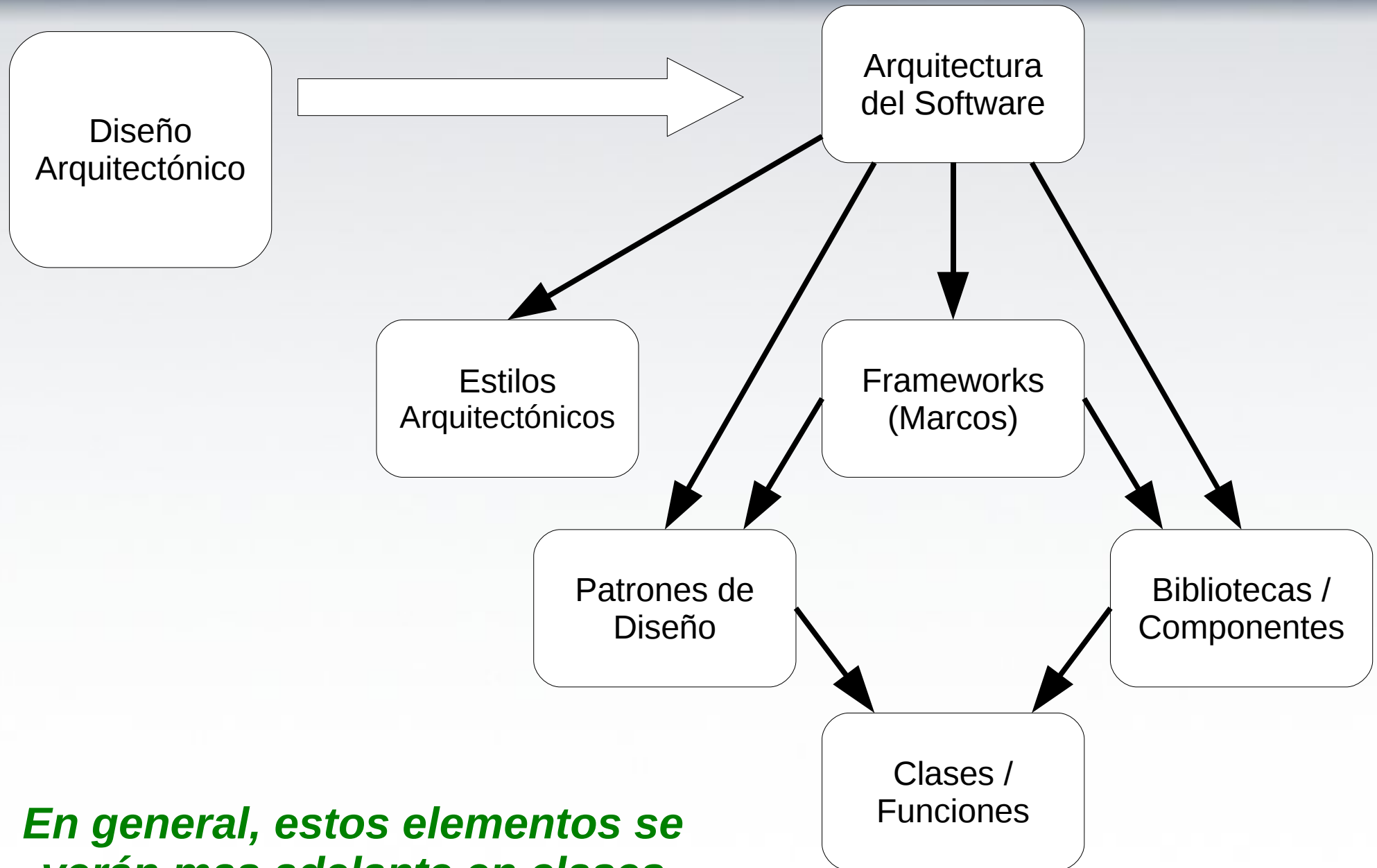
Arquitectura (Ejemplo con más detalle)



Arquitectura a 3 capas bien definida (Cleda)

Arquitectura (Ejemplo con más detalle)





En general, estos elementos se verán mas adelante en clases

Gracias

¡Gracias!

