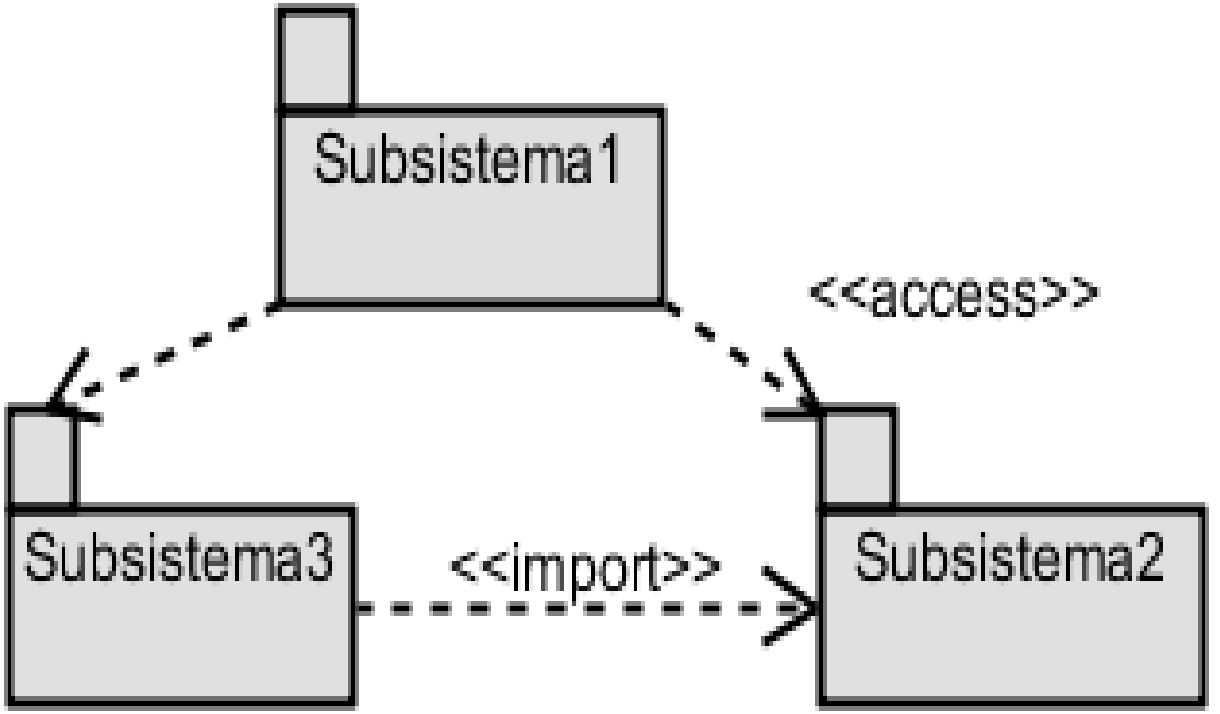
Tema 3.4:Diseño de la arquitectura



Contenido

Tema 3.4: Diseño de la arquitectura

- Conceptos de diseño de la arquitectura.
- Herramientas para su representación.
- Estilos arquitectónico.
- Actividades del diseño arquitectónico.

Bibliografía: **[PRES13 capítulo 7]**
[SOMM11 capítulo 6]
[ARLO05 capítulo 19]

Conceptos de diseño de la arquitectura

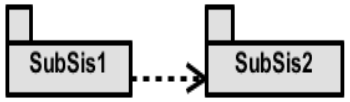
El **diseño de la arquitectura** va a proporcionar una imagen global de la estructura del sistema software, esbozando los artefactos más importantes que lo componen, principalmente **subsistemas y sus relaciones (interfaces).**

La **determinación de la arquitectura del software** consiste en la toma de **decisiones** respecto a:

- Cómo se va a dividir el sistema en subsistemas.
- Cómo deben interactuar dichos subsistemas.
- Cuál va a ser la interfaz de cada uno de estos subsistemas.
- Cuál va a ser el estilo arquitectónico usado.

Importancia de la arquitectura

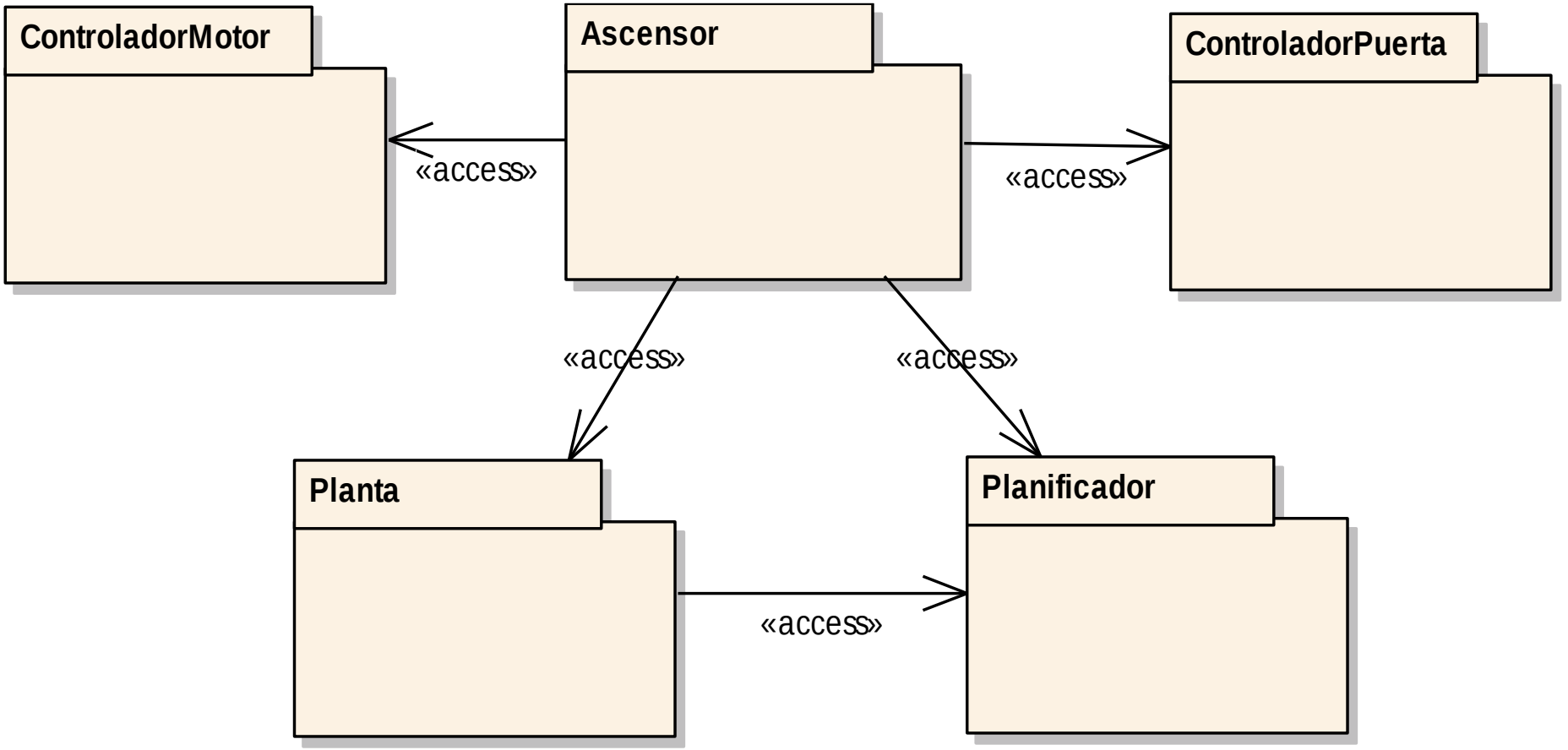
- Facilita la comprensión de la estructura global del sistema.
- Permite trabajar en los subsistemas de forma independiente.
- Facilita las posibles extensiones del sistema.
- Facilita la reutilización de los distintos subsistemas.

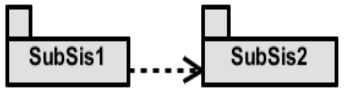


Herramientas para su representación

Diagrama de paquetes

Describe el sistema en torno a **agrupaciones lógicas** y proporciona una primera estructuración del sistema.

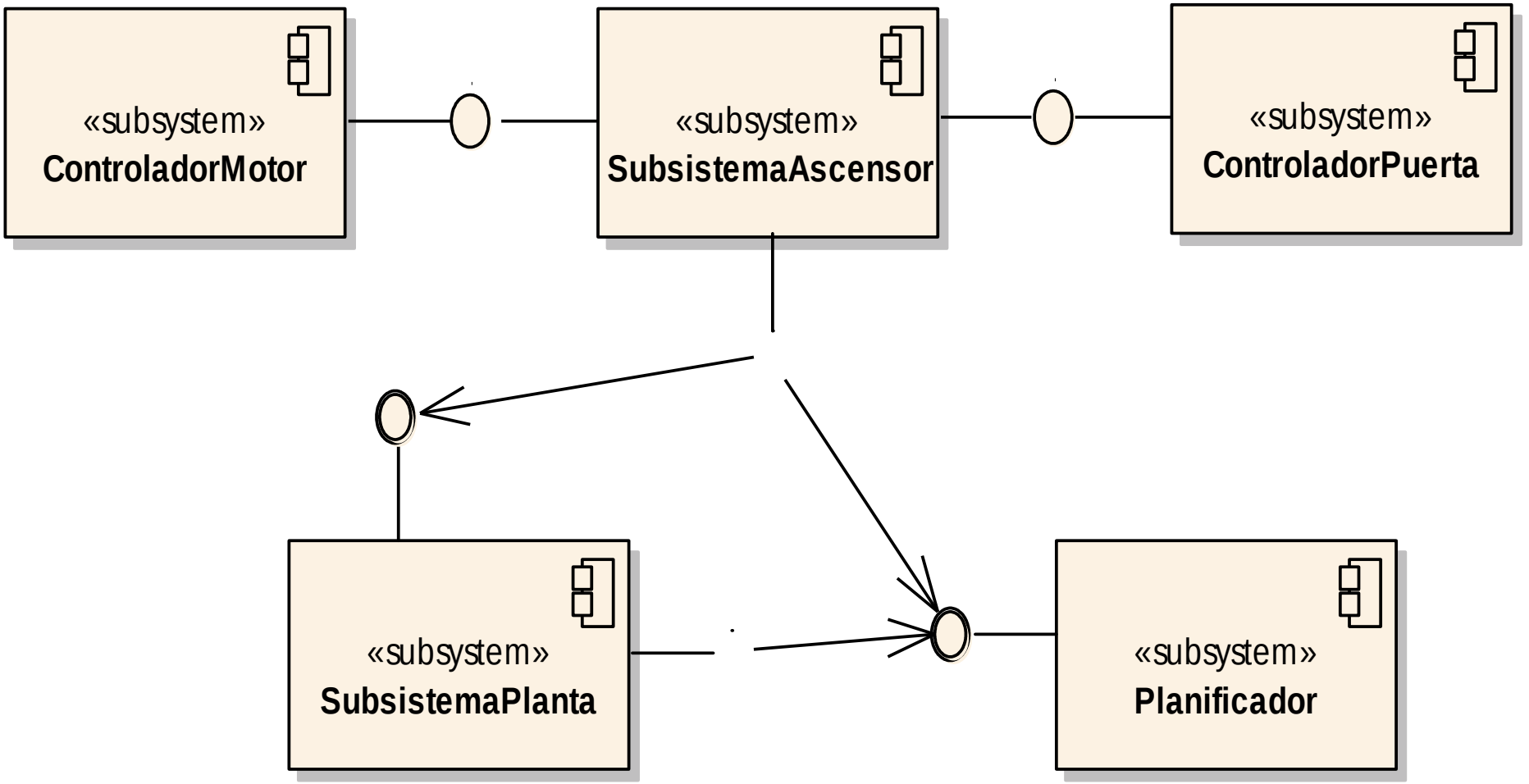




Herramientas para su representación

Diagrama de componentes

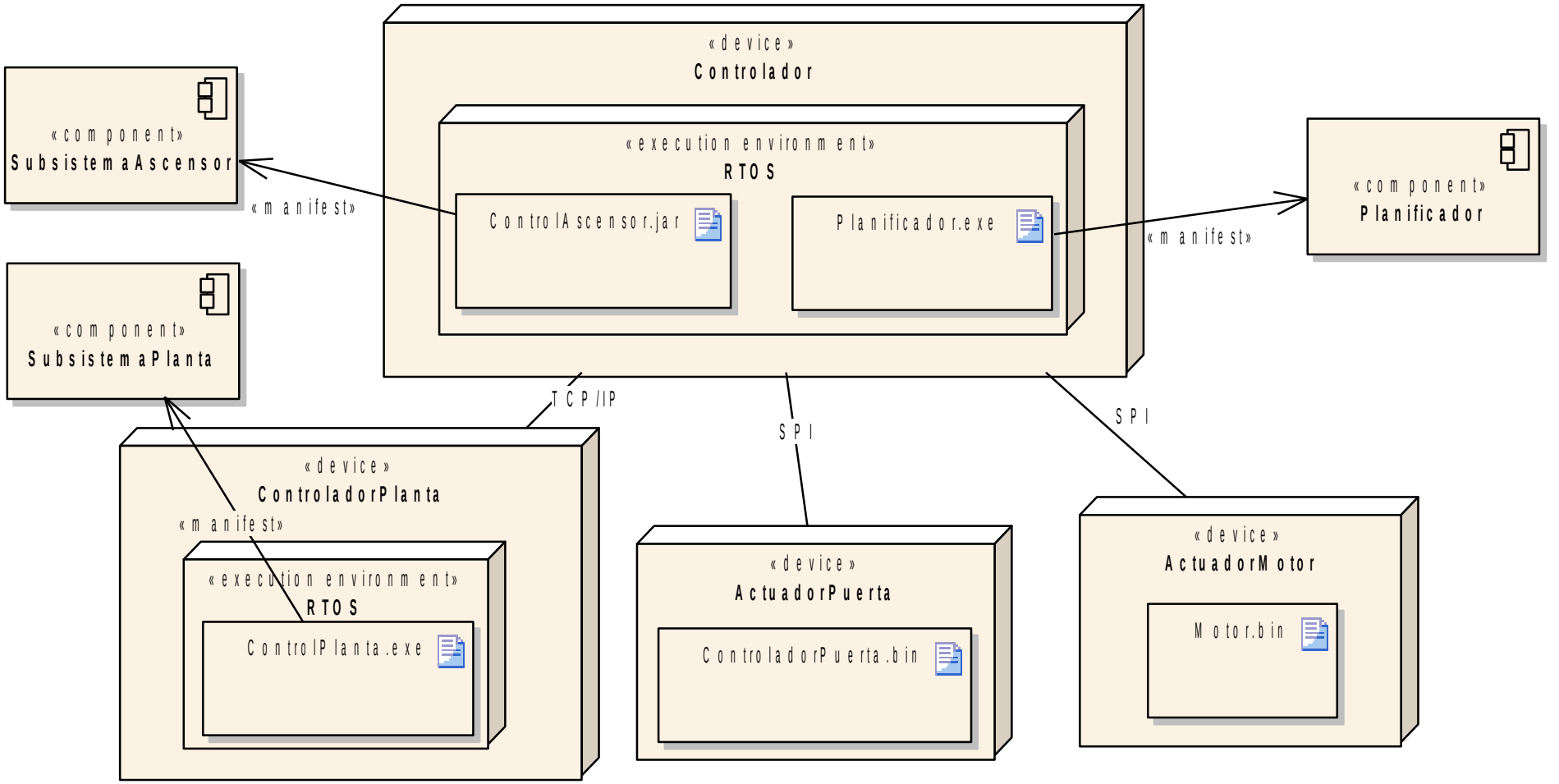
Representa una estructuración concreta del sistema a partir de los componentes software (sistemas) y su interrelación (interfaces).

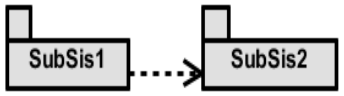


Herramientas para su representación

Diagrama de despliegue

Especifica el hardware físico sobre el que se ejecutará el sistema software y cómo cada uno de los subsistemas software se despliega en ese hardware.



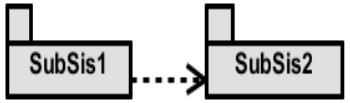


Estilos arquitectónicos

Un **estilo arquitectónico** proporciona un conjunto de subsistemas predefinidos, especificando sus responsabilidades e incluyendo reglas y guías para organizar las relaciones entre ellos. No proporcionan la arquitectura del sistema, sino una guía para obtener dicha arquitectura.

Estilos arquitectónico más generalizados:

- Arquitectura multicapa (Microkernel).
- Arquitectura cliente-servidor.
- Arquitectura de repositorio (Repository).
- Arquitectura MVC (Model-View-ControllerVC).
- MDA (Model Driven Architecture)



Estilos arquitectónicos

Arquitectura multicapa

Distribuye los subsistemas de un sistema en capas, de tal forma que:

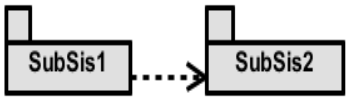
- Cada capa presta servicios a la capa inmediatamente superior y actúa como cliente sobre las capas más internas.
- El diseño incluye los protocolos que establecen cómo interactuará cada par de capas.
- Las capas más bajas proporcionan servicios de bajo nivel como protocolos de comunicación en red, acceso a base de datos, …

Ventaja:

- Cada capa puede diseñarse, implementarse y probarse de forma independiente
- Arquitectura fácilmente cambiable y portable:
  - Preservando la interfaz, una capa puede ser reemplazada por otra.
  - El cambio en la interfaz de una capa sólo afecta a la capa adyacente.

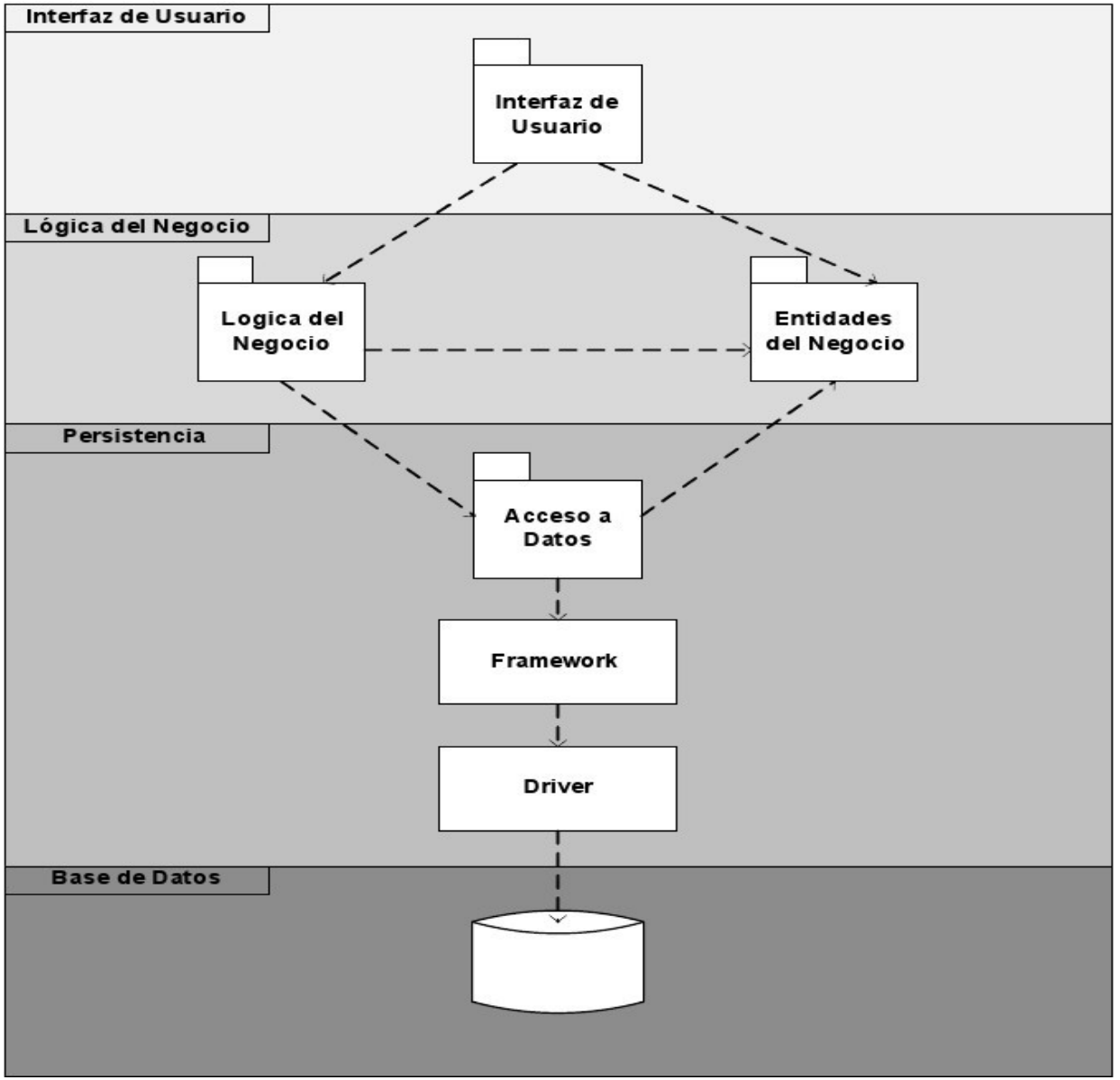
Desventajas:

- Dificultad para decidir cuáles son los servicios de cada capa.
- El rendimiento puede resultar afectado, debido a los múltiples niveles que hay que pasar para llegar a la que proporciona el servicio.



Estilos arquitectónicos

Arquitectura multicapa: Ejemplo



Estilos arquitectónicos

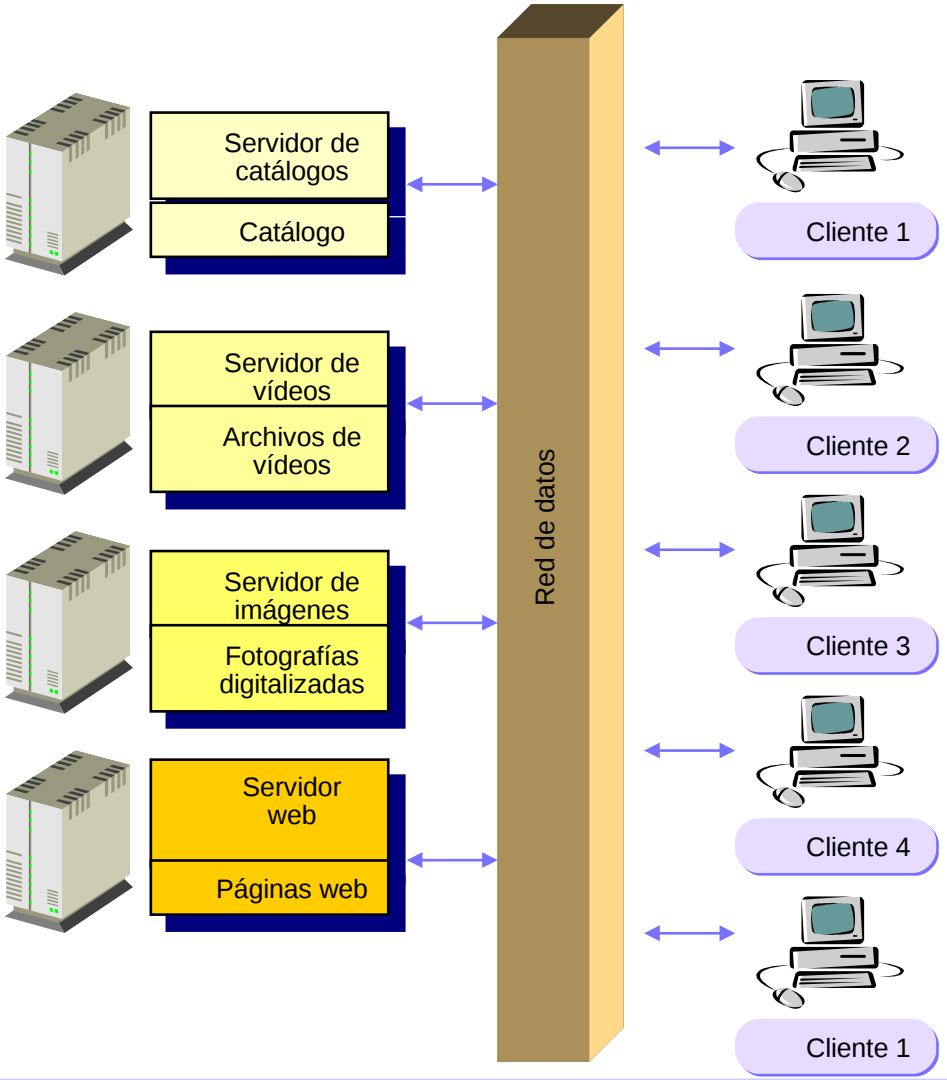
Arquitectura multicapa: Principios de diseño

- Las **capas** pueden diseñarse, construirse y probarse i**ndependientemente**.
- Una capa bien diseñada presenta **alta cohesión**.
- Las capas deben estar los mas **desacopladas** posible:
  - Dependencias en un sentido
  - Todas las dependencias expresadas en las interfaces.
- Una capa de un determinado nivel bien diseñada no tienen conocimiento de las capas superiores (buen **ocultamiento de información**).
- Las capas más inferiores deben diseñarse para prestar servicios de bajo nivel (**bajo acoplamiento**).

Estilos arquitectónicos

Arquitectura cliente-servidor

Subsistemas distribuidos que muestra cómo datos y procesamiento se distribuyen a lo largo de varios procesadores.



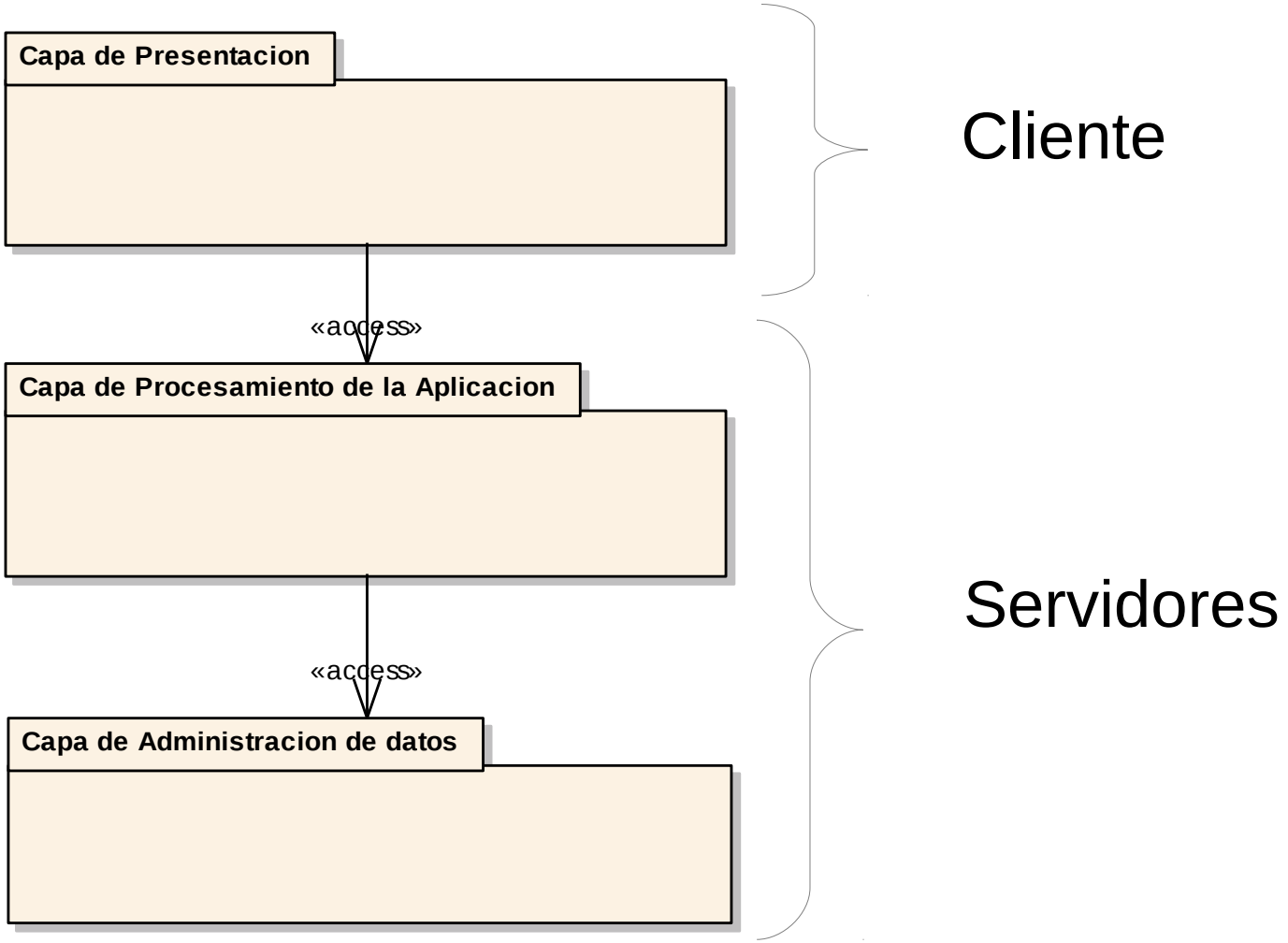
Componentes:

- **Servidores** independientes que ofrecen servicios a otros subsistemas.
- **Clientes** que pueden solicitar esos servicios.
- Una **red** que permite a los clientes acceder a los servicios ofrecidos por los servidores.

Estilos arquitectónicos

Arquitectura Cliente-servidor en capas

Se puede diseñar esta arquitectura por capas, lo normal son tres:



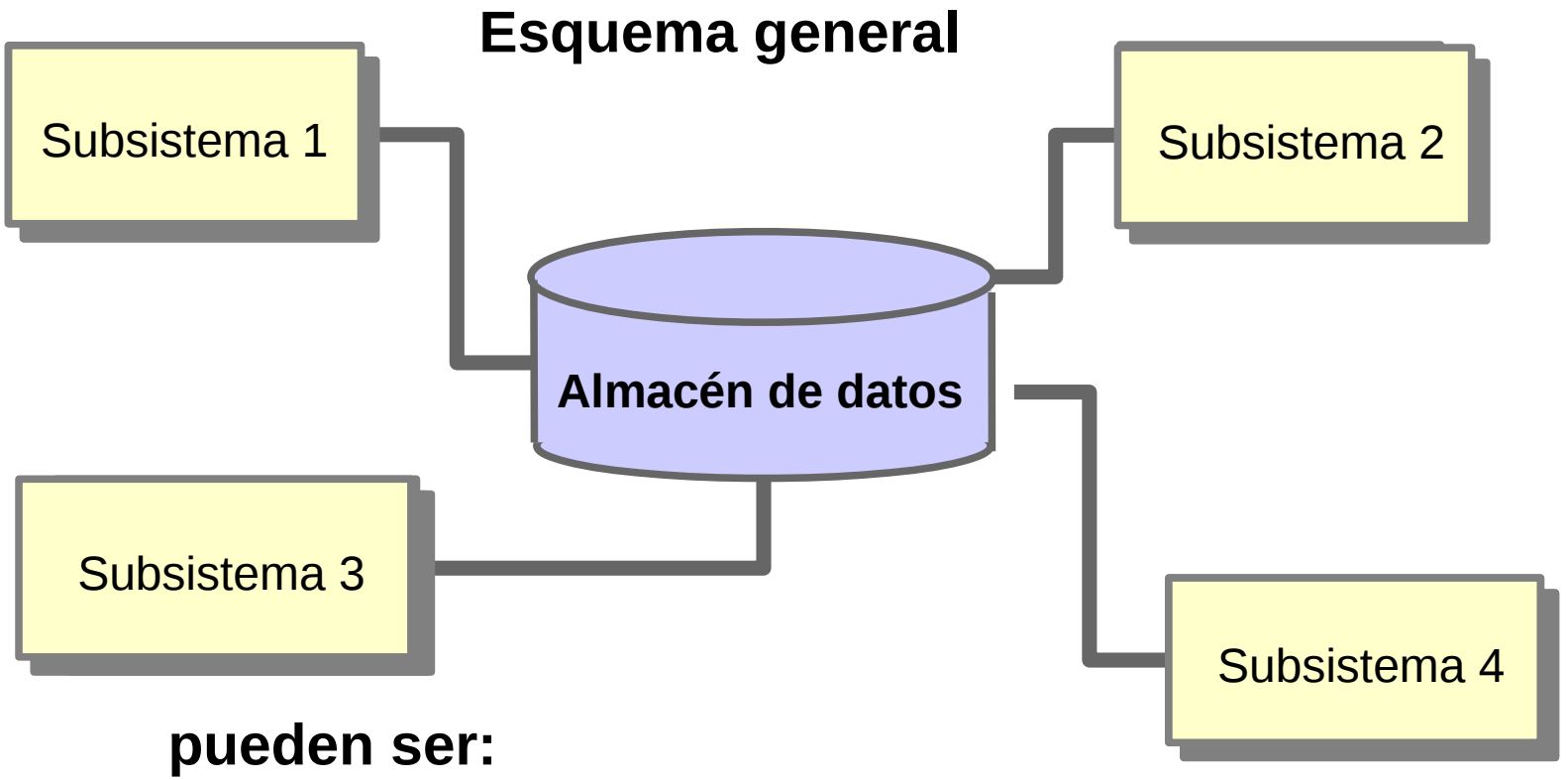
Arquitectura Cliente-servidor: principios de diseño

- La descomposición en clientes y servidores es una división fuerte del sistema que permite realizar el diseño, implementación y prueba de ambas partes **independientemente**.
- Mejora la **cohesión** de los subsistemas al proporcionar servicios específicos a los clientes.
- Reduce el **acoplamiento** al establecer un canal de comunicación para el intercambio de mensajes.
- Aumenta el nivel de **abstracción** al tener subsistemas o componentes distribuidos en nodos separados.
- Facilita la **reutilización** dado que se pueden usar marcos de desarrollo para sistemas distribuidos (CORBA, RMI, …).
- Los sistemas distribuidos pueden **reconfigurarse fácilmente** añadiendo servidores o clientes extras.
- Pueden escribirse clientes para nuevas plataformas sin necesidad de modificar a los servidores.

Estilos arquitectónicos

Arquitectura de repositorio

- Arquitectura en la que todos los datos se ubican en una base de datos central a la que acceden todos los subsistemas.



pueden ser:

Repositorio pasivo.
Repositorio activo

Estilos arquitectónicos

Arquitectura de Repositorio: ventajas/inconvenientes

Ventajas:

- Forma eficiente de compartir grandes cantidades de datos.
- Los subsistema productores de datos no necesitan saber de los subsistema consumidores de datos.
- Las actividades de seguridad, control de acceso y recuperación de datos están centralizadas.

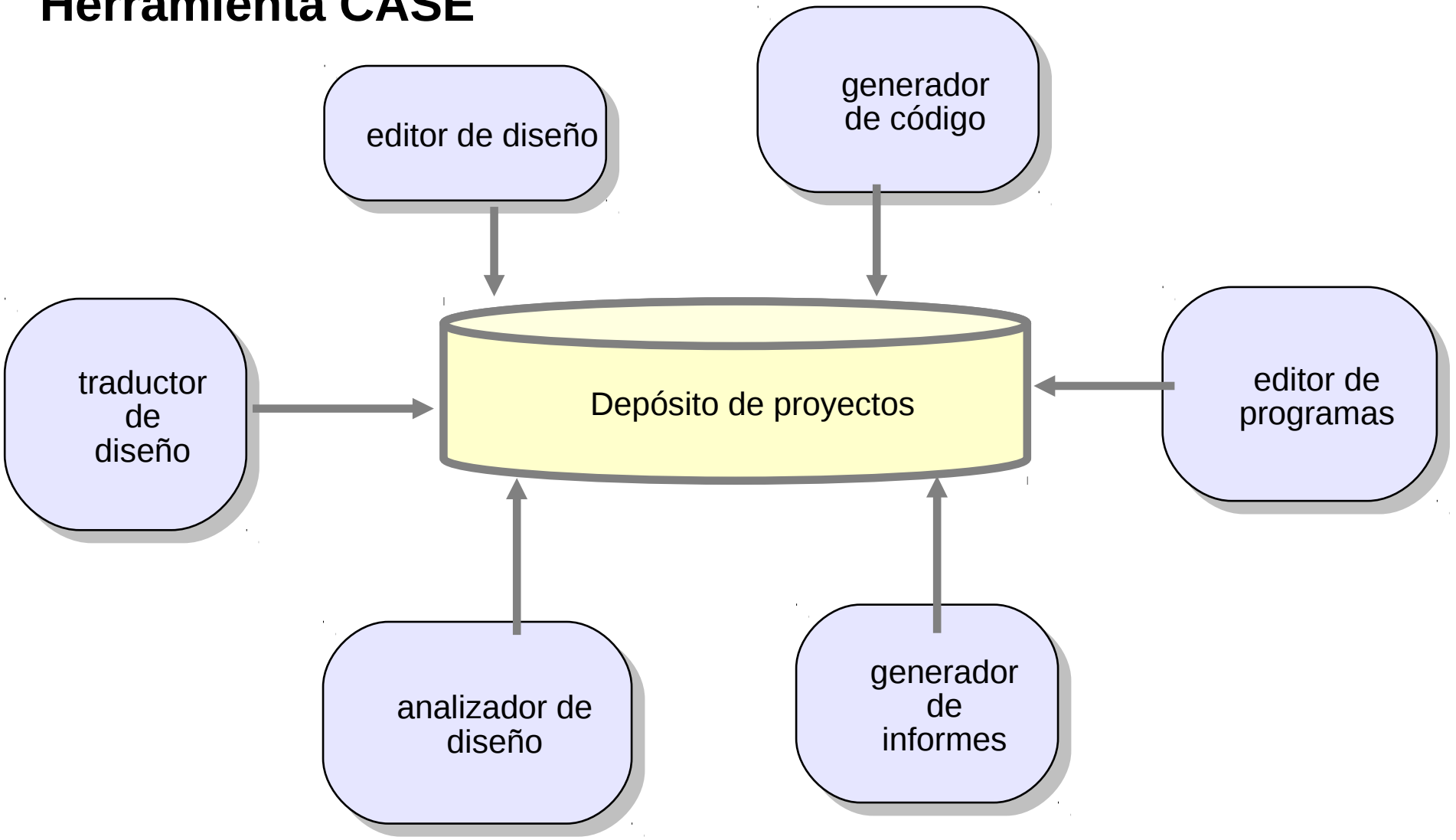
Inconvenientes:

- Todos los subsistemas deben estar acordes con el modelo de depósito. Dificultad para integrar nuevos subsistemas.
- Dificultad para evolucionar hacia otro modelo de datos.
- Fuerza la política de seguridad, control de acceso y recuperación de los subsistemas.

Estilos arquitectónicos

Arquitectura de repositorio: Ejemplo

Herramienta CASE



Arquitectura de Repositorio: Principios de diseño

- Se pueden diseñar **subsistemas independientes**, aunque deben conocer el esquema de los datos almacenados en el repositorio.
- Mejora la **cohesión** de los subsistemas, dado que cada subsistema tiene definida una funcionalidad específica para manipular la información
- El **acoplamiento** de los subsistemas con el repositorio es alto, haciendo difícil que un cambio en el repositorio no afecte a todos los subsistemas.

Estilos arquitectónicos

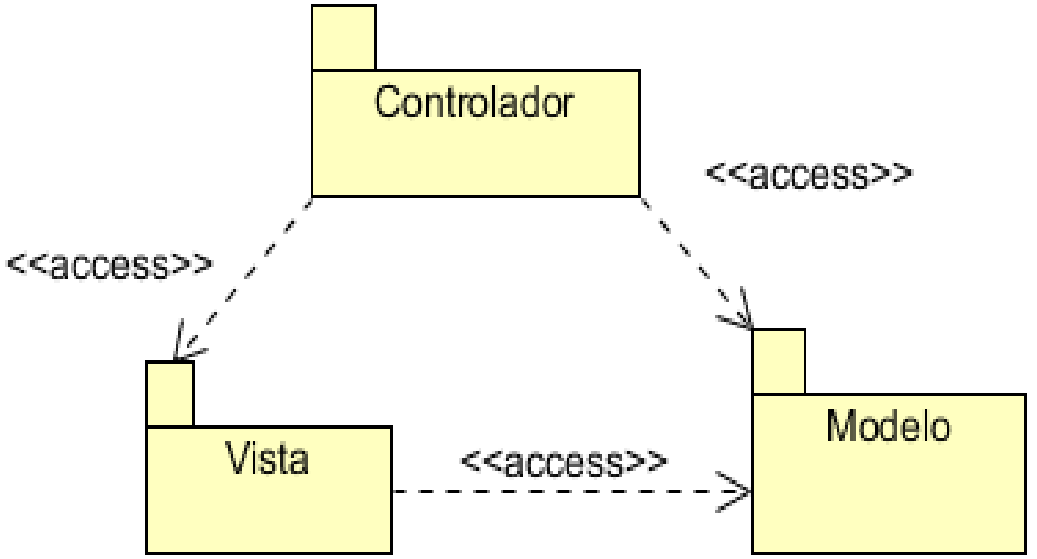
Arquitectura MVC

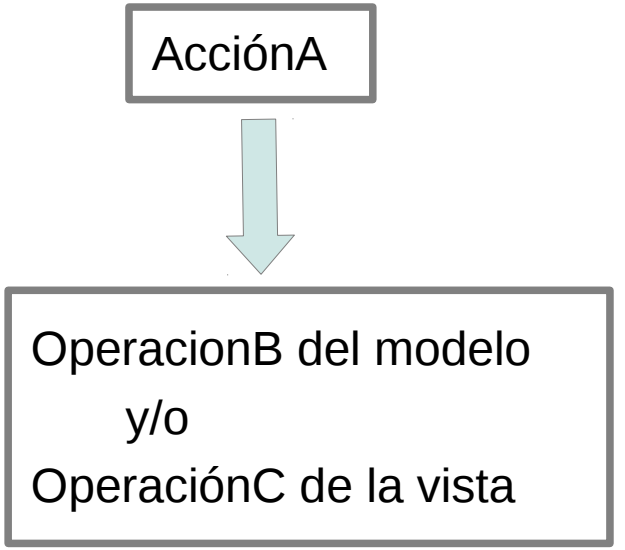
Arquitectura de tres componentes, donde cada una de ellos tiene definido su contenido, estos son:

Modelo: Responsable del conocimiento del dominio de aplicación.

Vista: Responsable de mostrar los objetos del dominio de aplicación al usuario.

Controlador: Responde a acciones del usuario, invocando peticiones al modelo y a la vista. Contiene reglas del tipo:





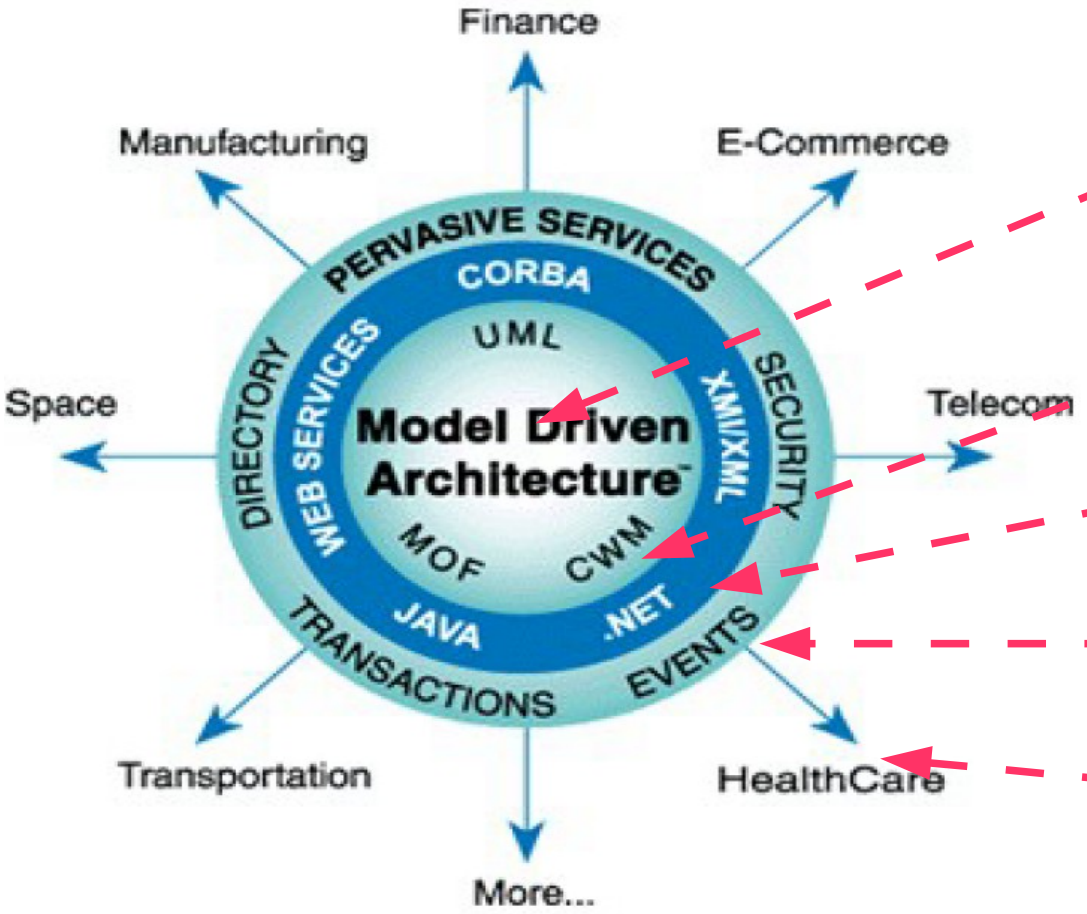
Arquitectura MVC: principios de diseño

- Cada uno de los subsistemas puede **diseñarse independientemente.**
- Se puede aumentar la **cohesión** de los subsistemas si la vista y el controlador se encuentran unidos en una capa de interfaz de usuario.
- Se reduce el **acoplamiento** dado que los canales de comunicación entre los subsistemas es mínimo.
- Los subsistemas vista y controlador hacen uso extensivo de componentes **reutilizables.**
- Es un sistema más **flexible** dado que se puede modificar la interfaz de usuario cambiando el subsistema vista, controlador o ambos.
- Se puede probar la aplicación separadamente de la interfaz de usuario.

Estilos arquitectónicos

MDA (Model Driven Architecture)

MDA es un framework para el desarrollo de software que nos permite organizar y gestionar las arquitecturas empresariales, basado en el uso de herramientas automatizadas para construir modelos independientes y transformarlos en implementaciones eficientes.



- Existen varios núcleos MDA (Informática de Empresa, tiempo real,...)
- Núcleos basados en estandares OMG
- Núcleos independientes de la plataforma
- Servicios esenciales de propósito general
- DTF (domain task for-ces) de estandarización de dominios

Estilos arquitectónicos

MDA : Modelos

Los modelos van a dirigir el proceso de comprensión, diseño, construcción, instalación, operación, mantenimiento y modificación. Éstos son, en función del nivel de abstracción:

CIM (Computation Independent Model))

- Modelado de negocio y requerimientos.

PIM (Platform Independent Model)

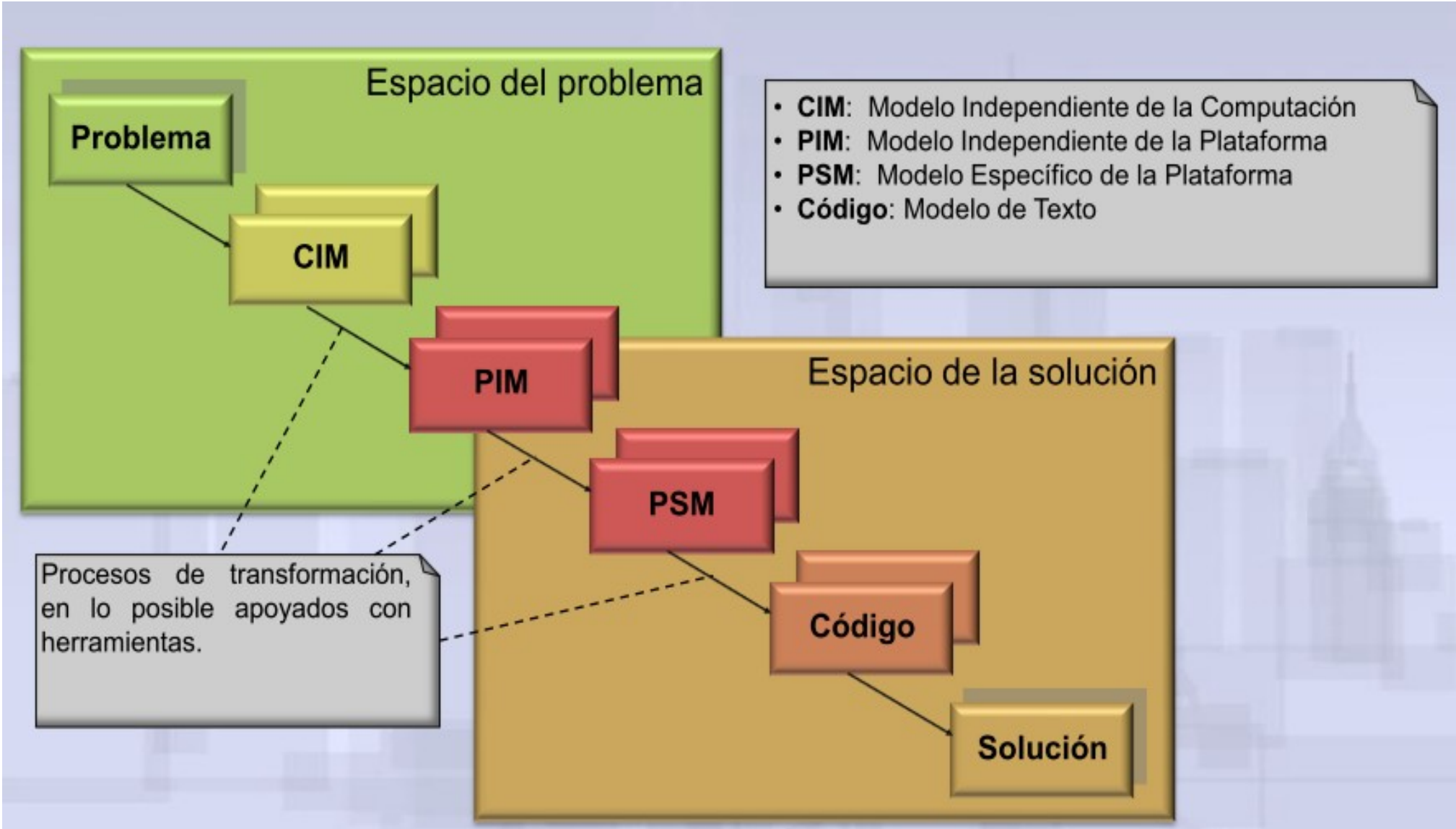
- Análisis y diseño independiente de la plataforma tecnológica .
- El analista realiza el PIM que describe el sistema.
- Un PIM se transforma en uno o mas PSM.

PSM (Platform Dependent Model)

- Diseño dependiente de la plataforma tecnológica.
- El arquitecto adapta el modelo para una implementación tecnológica específica.
- Cada PSM se transforma en código.

Estilos arquitectónicos

MDA : Modelos y transformaciones



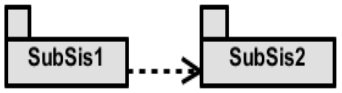
Tomado de los apuntes de Juan Bernardo Quintero

Estilos arquitectónicos

MDA: Principios de diseño

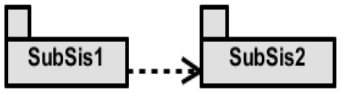
- **Representación Directa**: Enfocándose en el dominio del problema y no de la tecnología.
- **Automatización:** Herramientas que apoyen y facilitan las labores mecánicas.
- **Estándares abiertos:** Que eliminen la diversidad y formalicen el desarrollo en cada plataforma.

VER Seminario: MDA



Actividades del diseño arquitectónico

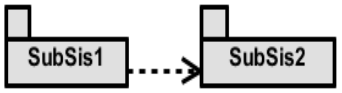
- Identificar los objetivos de diseño.
- Determinar la arquitectura software, y los correspondientes subsistemas.
- Modelar la arquitectura software.
- Refinar la descomposición en subsistemas.



Actividades del diseño arquitectónico

Identificar Objetivos

- Se identifican las cualidades deseables del sistema que van a determinar el diseño del mismo.
- Se obtienen a partir de los requisitos no funcionales o del dominio de la aplicación.
- Seleccionar un pequeño conjunto de objetivos de diseño que el sistema debe satisfacer necesariamente.
- Adquirir compromisos, dado que muchos objetivos de diseño son contrapuestos.
  - Por ejemplo, priorizando los objetivos de diseño.



Actividades del diseño arquitectónico

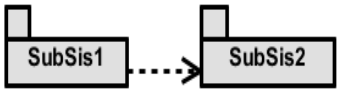
Determinar y modelar la arquitectura

Para **determinar** la arquitectura hay que:

- Seleccionar el estilo arquitectónico que mejor se adapta al sistema en desarrollo.
- Identificar subsistemas en el dominio del problema.
- Añadir subsistemas predefinidos de acuerdo al estilo arquitectónico seleccionado.

Para **modelar** la arquitectura hay que:

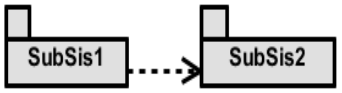
- Diseñar la arquitectura en un diagrama de paquetes.
- Elaborar el diagrama de componentes de la vista de arquitectura.
- Realizar el diagrama de despliegue.



Actividades del diseño arquitectónico

Modelar: Diagrama de paquetes

- Los diagramas de paquetes permiten modelar una primera estructuración o partición del sistema en base a uno o más paquetes:
  - Cada paquete agrupa a un conjunto de clases relacionadas semánticamente.
  - Los paquetes pueden guiarnos en la identificación de los subsistemas y/o componentes reutilizables del sistema.
- Al identificar los paquetes tener en cuenta los siguientes aspectos:
  - Tratar de identificar paquetes cohesivos y con poca interacción con el resto de paquetes de acuerdo a la arquitectura planteada.
  - Procurar diseñar paquetes para que puedan ser reutilizados en otros proyectos.
  - También pueden integrarse subsistemas de proyectos anteriores.
  - Evitar las dependencias cíclicas entre paquetes.

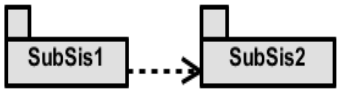


Actividades del diseño arquitectónico

Modelar: Diagrama de componentes

- Los diagramas de componentes permiten estructurar el sistema en subsistemas en base a componentes que pueden ser reemplazables.

- A partir del diagrama de paquetes se debe determinar qué partes pueden definir a componentes. Para ello habrá que:
  - Definir interfaces proporcionadas de cada componente para determinar las operaciones que pueden ser utilizadas por otros componentes.
  - Especificar las interfaces requeridas en cada componente para poder desarrollar su funcionalidad.
  - Construir el componente de modo que su contenido interno sea independiente del exterior, salvo a través de las interfaces proporcionadas y requeridas.



Actividades del diseño arquitectónico

Refinar la arquitectura

- Refinar los subsistemas hasta satisfacer los objetivos de diseño.
- Establecer los siguientes aspectos genéricos:
  - Correspondencia Hardware-Software.
  - Administración de datos persistentes.
  - Especificación de una política de control de accesos.
  - Diseño del flujo de control.
  - Diseño de la concurrencia y sincronización.
  - Definición de las condiciones del entorno.

VER Seminario: Ejemplo de Arquitectura