

Diagrama de Paquetes

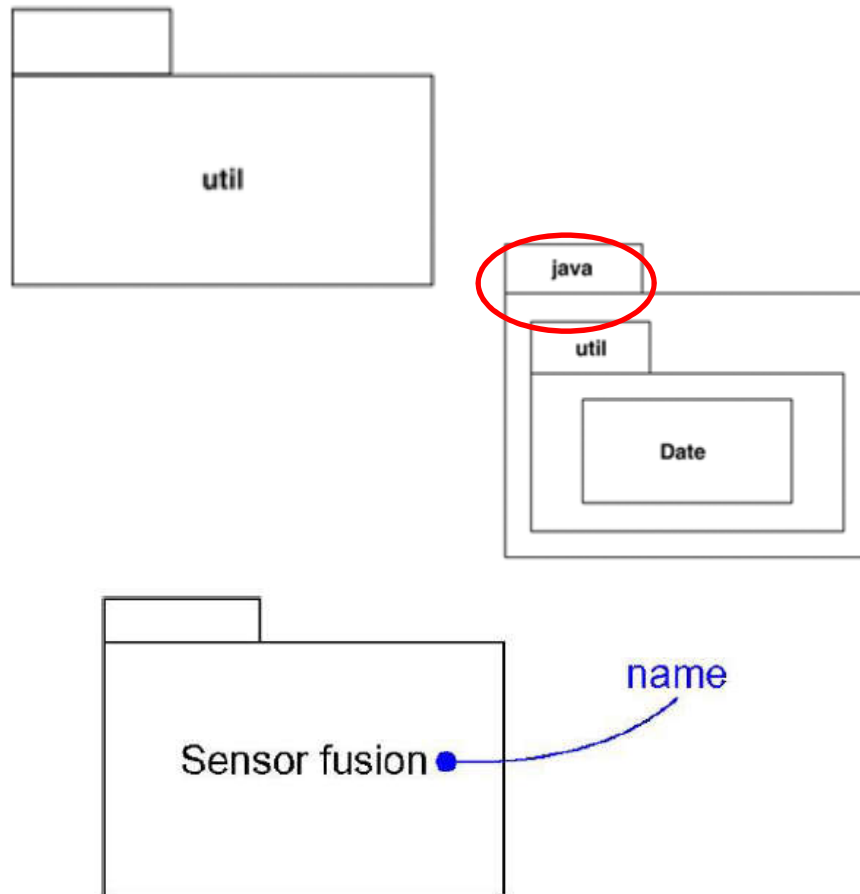


Paquetes

- “Un paquete es un mecanismo de propósito general para la organización en grupos de elementos”[Booch,2000]
- Organización Elementos



Notación UML

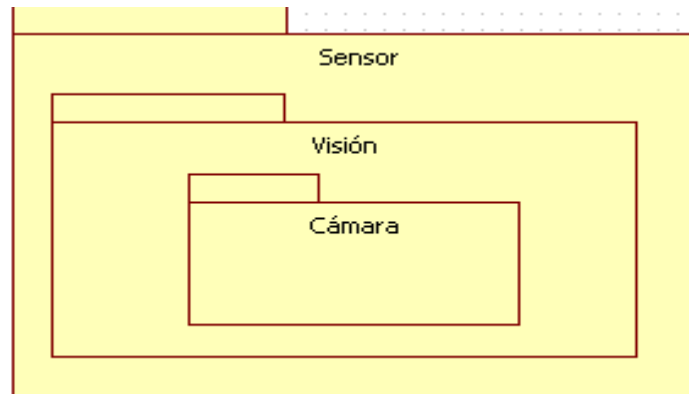


- Carpetas con etiquetas.
- Los paquetes subordinados se incluyen en el interior
- El nombre se encuentra en la etiqueta o en el centro de la carpeta



Nombres de Paquetes

- Cada paquete representa un namespace
- Se utilizan nombres calificados
 - Ej: La clase *Cámara* que vive en el paquete *Visión* que vive en el paquete *Sensor*. El nombre completo de la clase sería *Sensor::Visión::Cámara*



Nombres de Paquetes

- Elementos del mismo tipo deben ser llamados de manera única dentro del contexto del paquete que lo incluye
- Ej:
 - No pueden existir 2 clases llamadas Cola en el mismo paquete
 - Puede existir una clase **cola** en el paquete P1 y una clase **cola** en el paquete P2

Diferenciados por
su nombre
completo

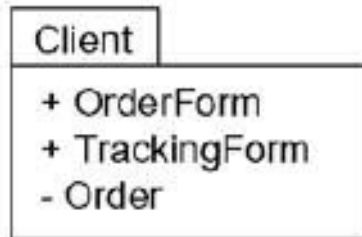


- P1::cola
- P2::cola

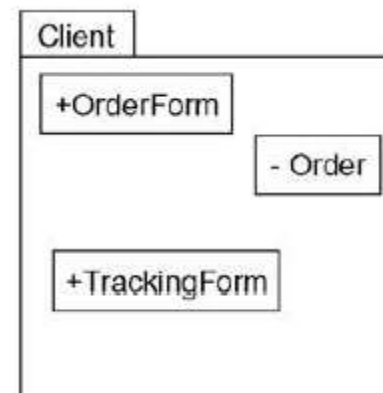


Elementos de Paquetes

- Un paquete es un conjunto de cualquier tipo de elementos: Clases, interfaces, componentes, otros paquetes etc.



Anidamiento Textual



Anidamiento Gráfico

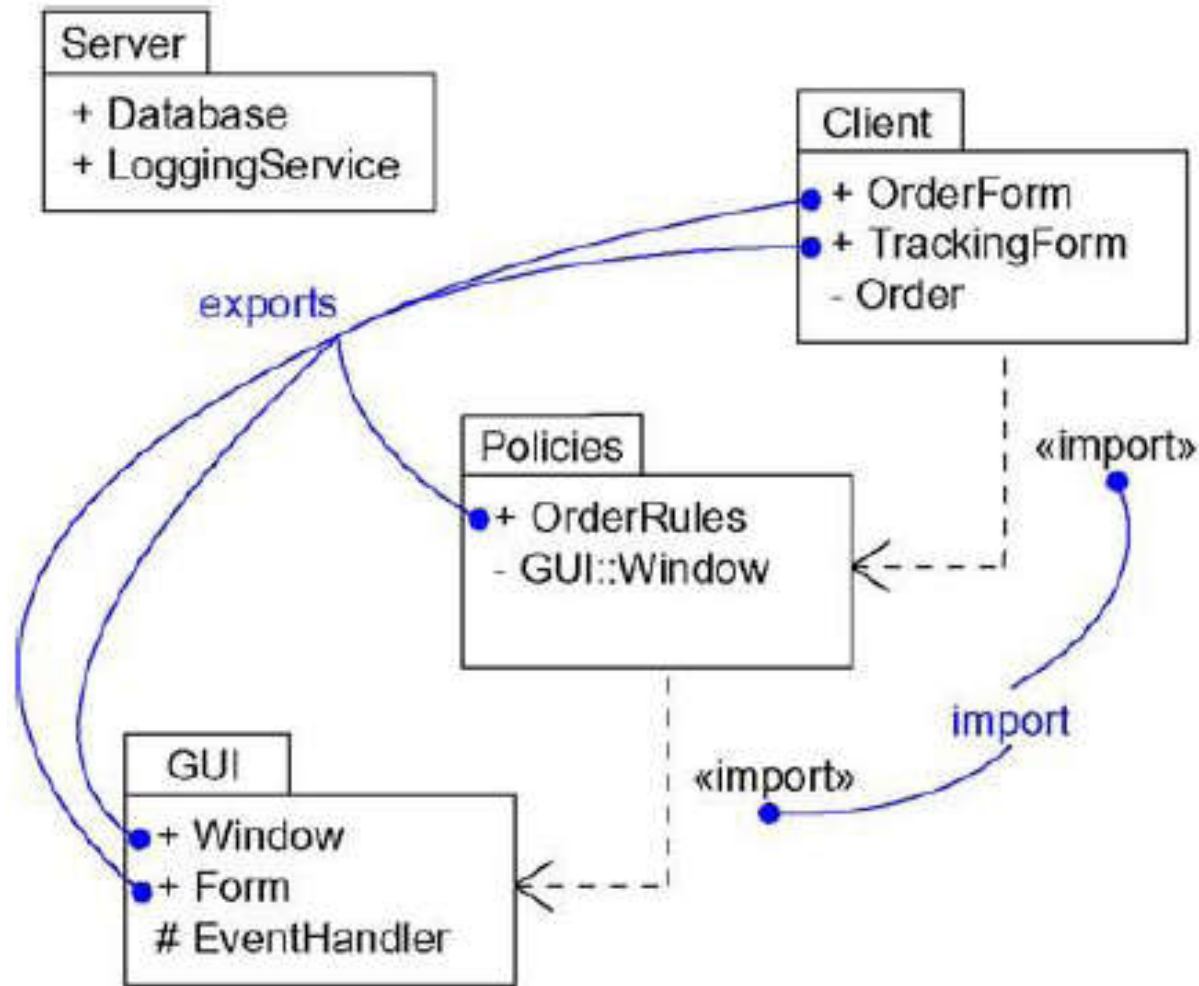


Visibilidad

- Los elementos públicos pueden ser vistos al contenido de cualquier paquete que incluya ese elemento
- Los Protegidos solo pueden ser vistos por sus hijos y no pueden ser vistos fuera del paquete en el que están declarados
- Los privados no pueden ser vistos fuera del paquete

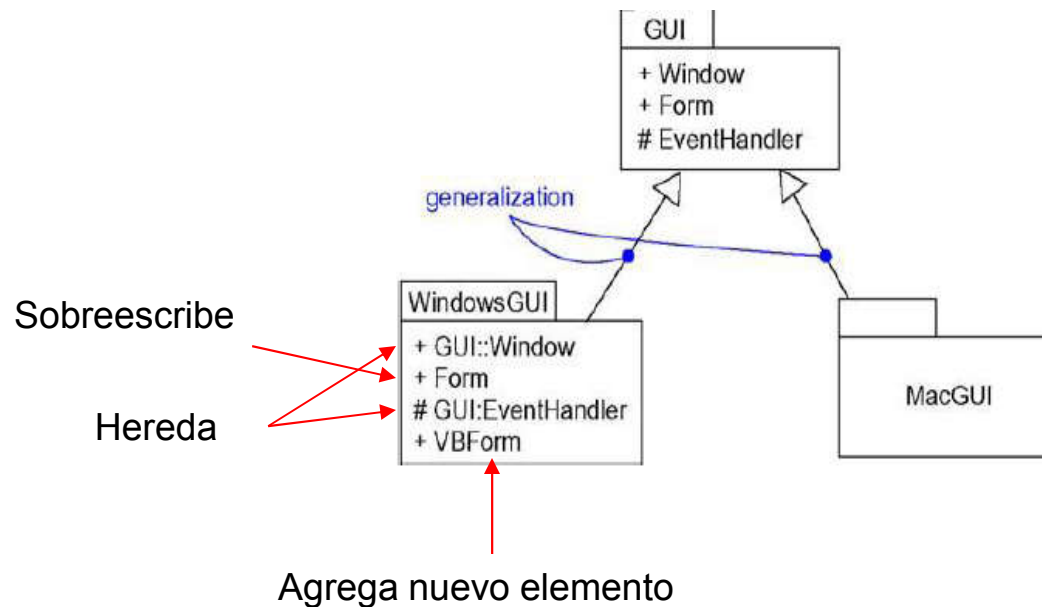


Organización por paquetes



Generalización

- o Similar a la generalización entre clases



Diferencias con clases

- Las clases son abstracciones de cosas encontradas en la solución del problema
- Los paquetes son mecanismos utilizados para organizar los elementos del modelo
- Los paquetes no tienen identidad (no se generan instancias de un paquete) son invisibles en el sistema ejecutándose
- Las clases tienen identidad (tienen instancias) son elementos en el sistema ejecutándose



Escoger las Clases de un paquete

- Principio común de encierro: Las clases de un paquete necesitan cambios por razones similares
- Principio común de la reutilización : Las clases de un paquete deben ser reutilizadas juntas



Ejemplo

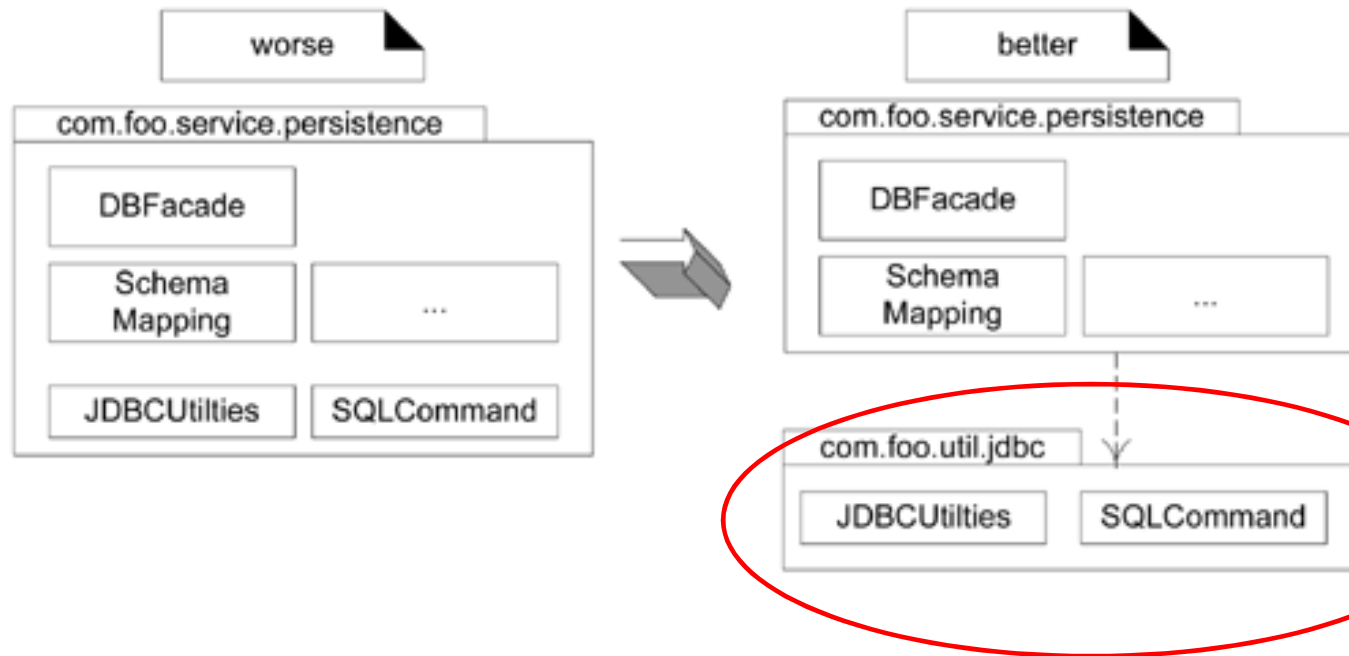
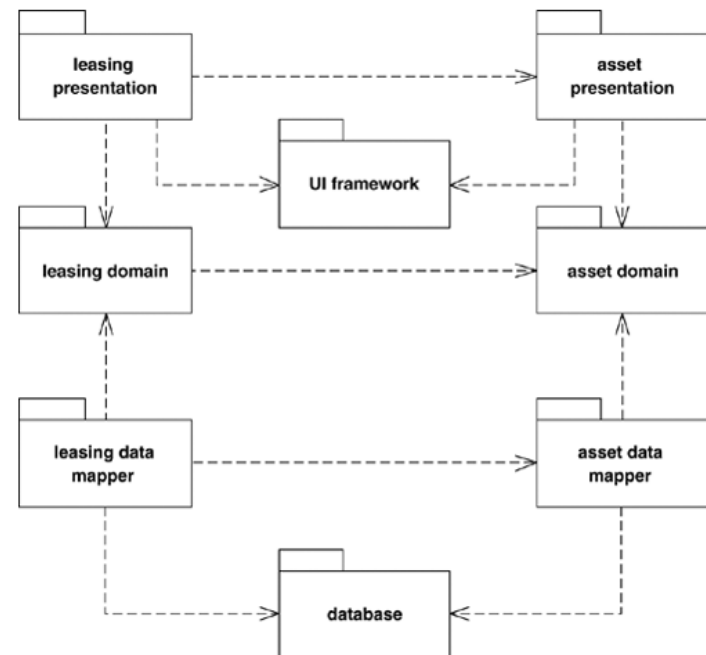


Diagrama de Paquetes

- Muestra los paquetes y sus dependencias
- Las dependencias entre paquetes resumen las dependencias entre los contenidos de los paquetes





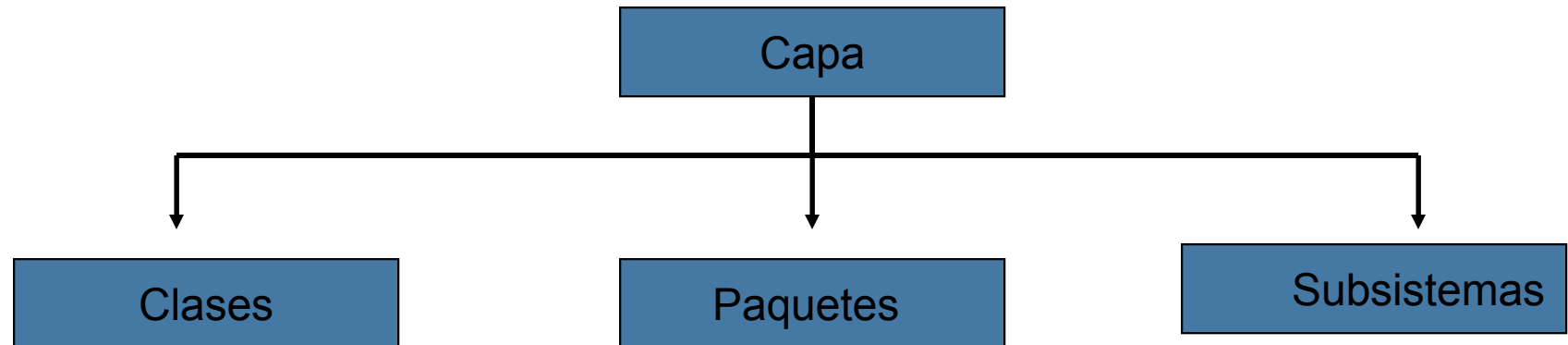
“Paquetes en la
arquitectura lógica”
.

Arquitectura Lógica

- Transición de Análisis a Diseño
- Diseño muy alto nivel
- Organización a gran escala de las clases de software en: [Larman]
 - Paquetes
 - Subsistemas
 - Capas



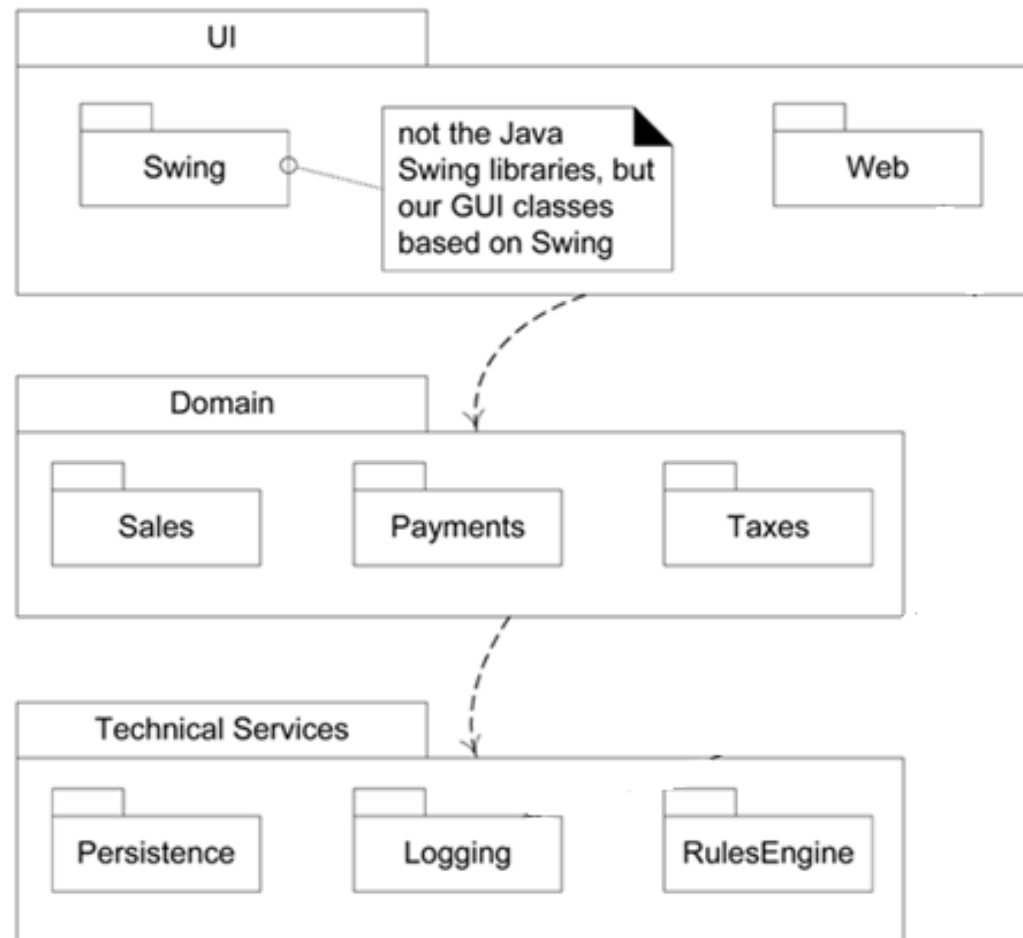
Capa



- Agrupa componentes Cohesivos
- Capas más comunes:
 - Interfaz de Usuario
 - Lógica de aplicación
 - Objetos de Dominio
 - Servicios Técnicos
- Una capa puede ser modelada con paquetes



Capas con notación de diagrama de paquetes



Arquitectura software

- La organización del software
- Selección elementos estructurales
- Interfaces
- Comportamiento
- Composición de los elementos estructurales y de comportamiento
- Estilo arquitectural, que guía la organización de los elementos y sus interfaces



Capas y particiones

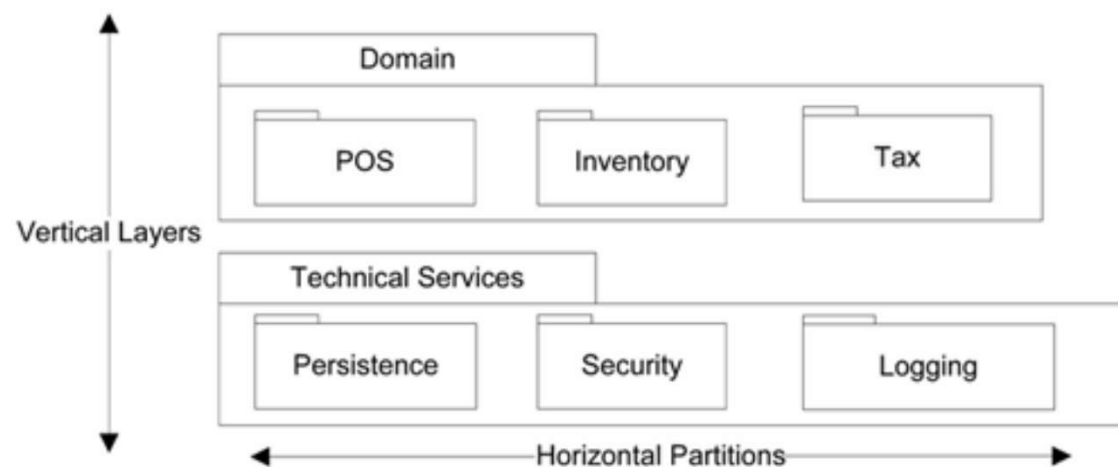
- **Capas**

- Divisiones verticales

- **Particiones**

- Divisiones horizontal de una capa

Figure 13.6. Layers and partitions.





“Fast, Cheap,
Good: Choose any
two.”