



# UCA

Programación Orientada a Objetos I - 2023

## UML – PAQUETES

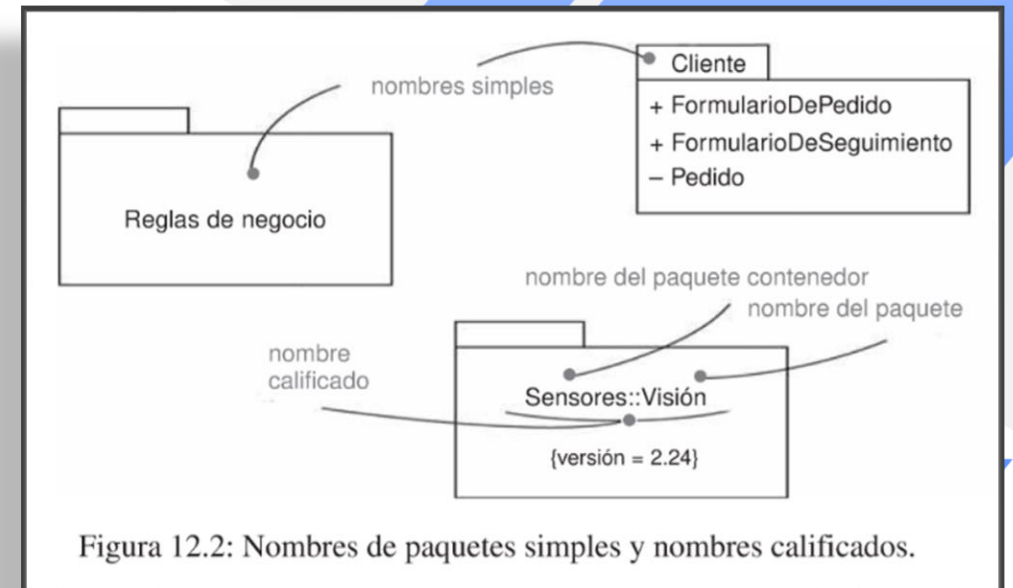
# Paquetes

- Los sistemas simples no necesitan mucha estructura (quizás por eso los llamamos simples).
- Los sistemas complejos necesitan estructurar y jerarquizar sus componentes para mantener el orden
- Un paquete permite organizar elementos en grupos
- Permiten una más fácil comprensión
- Permiten controlar el acceso de los contenidos



# Paquetes

- Es un mecanismo de propósito general para organizar el propio modelo de manera jerárquica
- Gráficamente se representa como una carpeta con una pestaña o solapa
- Tiene un nombre que lo distingue de otros paquetes
  - Simple: solo tiene su nombre
  - Calificado: si se antepone el nombre del paquete contenedor separado por dos símbolos dos puntos (::)
- Si se muestra su contenido, el nombre va en la pestaña o solapa
- Si no se muestra su contenido, el nombre puede estar dentro de la carpeta
- El nombre puede ser cualquier texto en UML, pero en la práctica terminará siendo un nombre corto como los de el resto de elementos

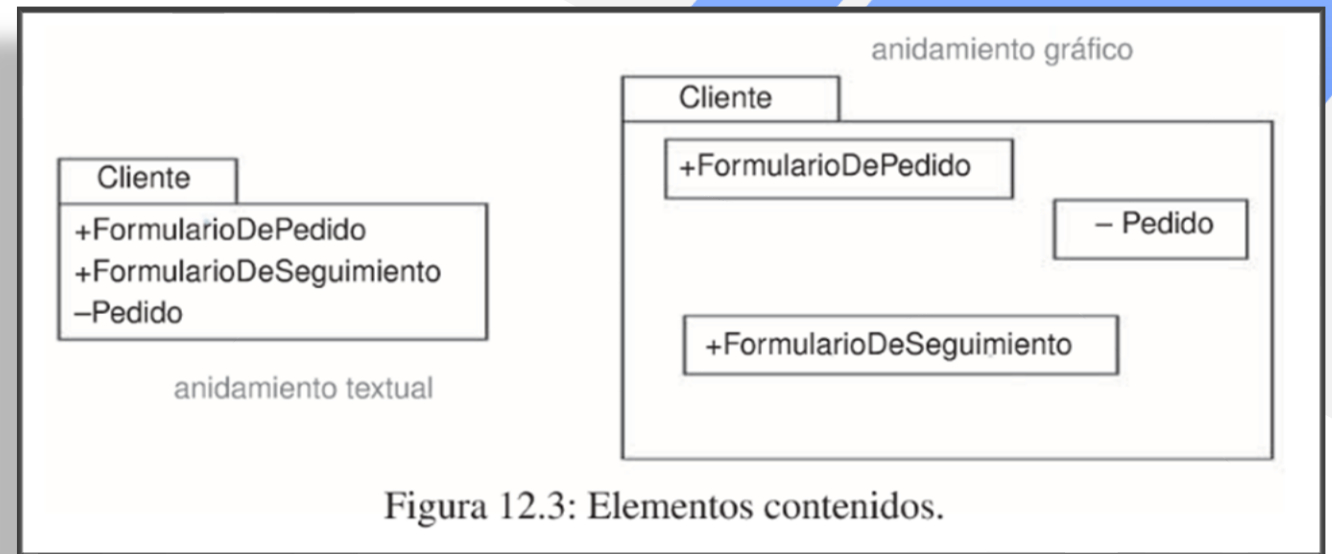


# Paquetes

- Un paquete puede contener otros elementos:
  - Clases, componentes, casos de uso, diagramas, etc.
  - Incluso otros paquetes
- Se trata de una relación compuesto, es decir que si un paquete se destruye, también se destruye su contenido
- Cada elemento solo puede pertenecer a un paquete.
- Los paquetes deben tener un nombre único dentro de su paquete contenedor
- Como los paquetes pueden contener otros paquetes, podemos decir que hay un anidamiento o estructura jerárquica de paquetes.
- Si bien no hay un límite explícito, en la práctica no deberían exceder los dos o tres niveles para mantener una visión comprensiva
- Los paquetes evitan tener modelos planos, donde todos los nombres deberían ser únicos

# Paquetes

- UML asume que hay un paquete raíz anónimo por lo que no hace falta especificarlo
- Se puede controlar la visibilidad de los paquetes, es decir que podemos modelar que componentes del paquete pueden ser accedidos desde fuera de si mismo
  - Normalmente los contenidos serán públicos, pero no obligatoriamente
  - Se pueden utilizar la misma visibilidad que para las clases (público, privado o protegido)
  - Se muestra como siempre con los símbolos + - #
- Se puede mostrar el contenido de un paquete en forma textual o en forma gráfica



# Paquetes

- Diferencia entre paquetes y clases:
  - Las clase son abstracciones de cosas encontradas en el problema o la solución
  - Los paquetes son mecanismos para organizar elementos del modelo
- La mayoría de las veces agrupan el mismo tipo de elementos (ejemplo, paquetes de clases)
- Pero pueden incluir diferentes elementos (clases, casos de uso, especificaciones, etc)
- Lo importante es que tengan:
  - Alta cohesión: Que entre si sirvan a una misma tarea, que estén **relacionados conceptualmente** entre si
  - Bajo acoplamiento: Que sean **lo más independiente posible**, dentro de sí mismo y con el entorno