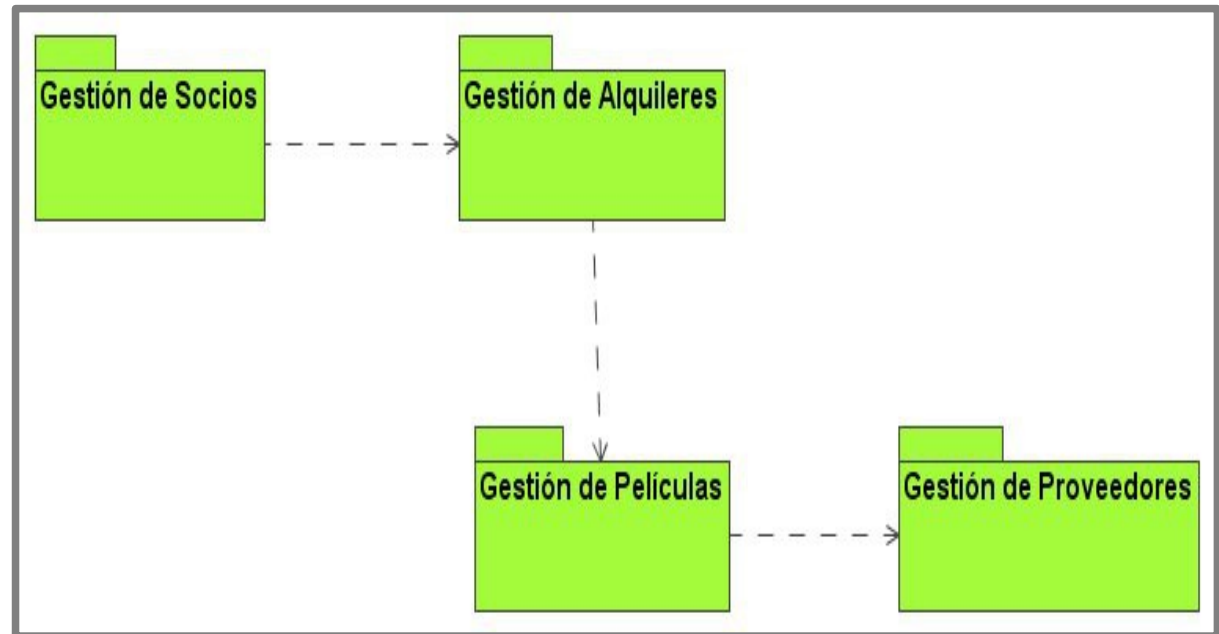
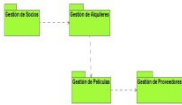


# Diagramas de paquetes





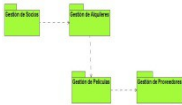
# Diagramas de paquetes

Diagrama de UML usado para describir la estructura de un sistema en base a agrupaciones lógicas y las dependencias entre éstas.

Elementos básicos de un diagrama de paquetes: paquetes y sus relaciones.

Se usan para:

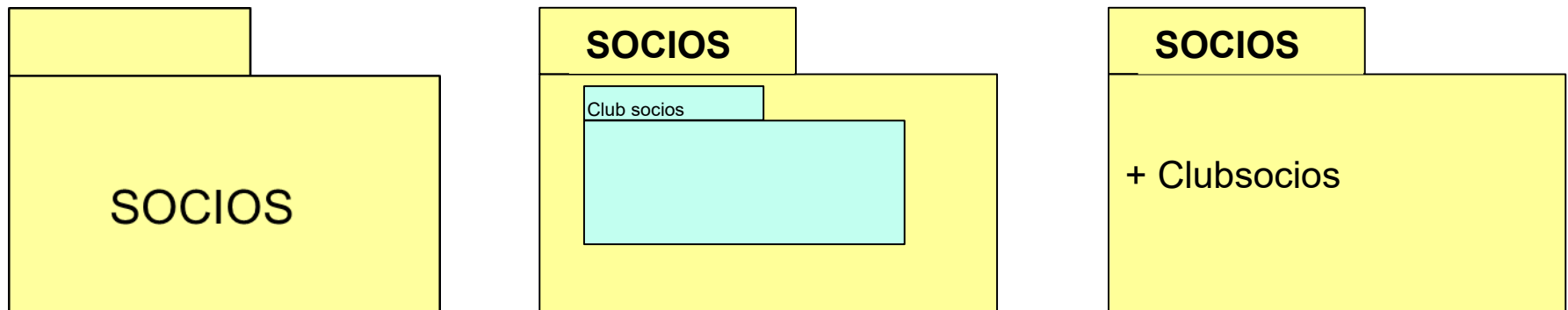
- Agrupar elementos relacionados semánticamente.
- Definir un “límite semántico” en el modelo.
- Proporcionar unidades para trabajo en paralelo y gestión de la configuración.

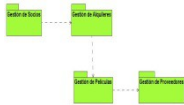


# Diagramas de paquetes

**Paquete:** Es un mecanismo de propósito genérico para agrupar y encapsular elementos del modelo (incluidos otros paquetes) y diagramas.

- Proporciona un espacio de nombres dentro del cual todos los nombres deben ser únicos.
- Cada elemento es exclusivo de un paquete.
- Cada paquete tiene un nombre que lo distingue de otros paquetes.
  - Nombre simple.
  - Nombre cualificado, formado por el nombre del paquete precedido por el nombre del paquete en el que se encuentra (**Socios::ClubSocios**).

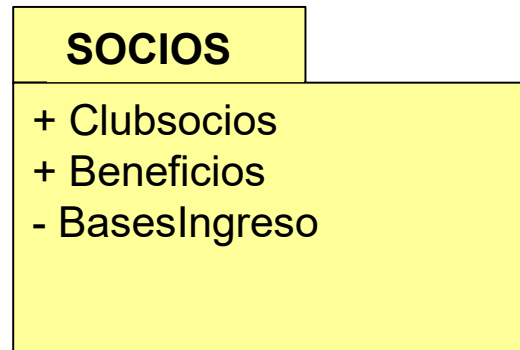


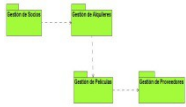


# Diagramas de paquetes

**Visibilidad del contenido del paquete:** La visibilidad de un elemento indica si son visibles o no a los clientes del paquete.

| VISIBILIDAD | SÍMBOLO | SEMÁNTICA   |
|-------------|---------|---|
| Público     | +       | Los elementos con visibilidad pública son visibles a elementos fuera del paquete; se exportan por el paquete. |
| Privado     | -       | Los elementos con una visibilidad privada sólo son visibles por otros elementos de ese paquete.               |

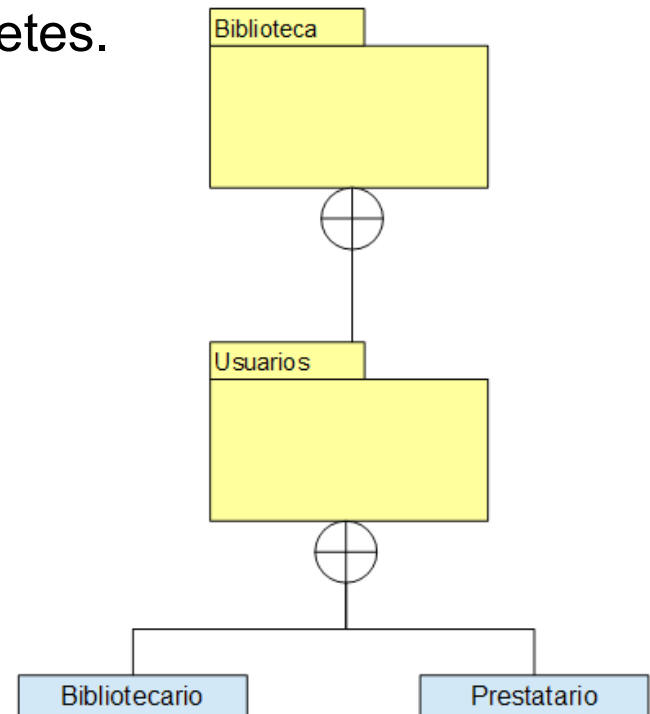
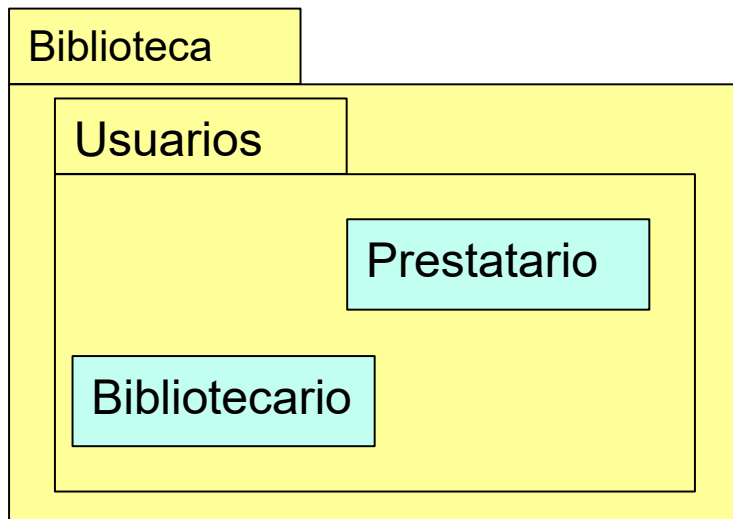


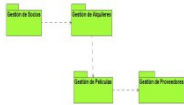


# Diagramas de paquetes

**Paquetes anidados:** Los paquetes pueden estar anidados dentro de otros paquetes en cualquier nivel.

- Los paquetes anidados tienen acceso al espacio de nombres de su paquete propietario.
- El paquete contenedor debe utilizar nombres cualificados para acceder a los contenidos de sus paquetes.

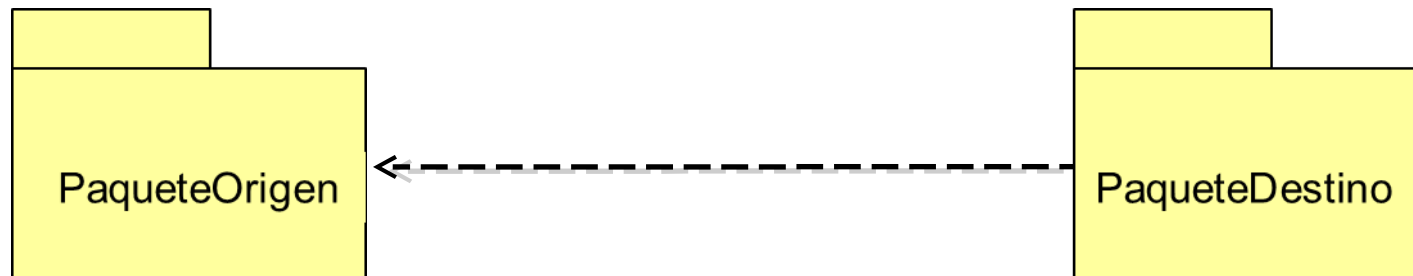




# Diagramas de paquetes

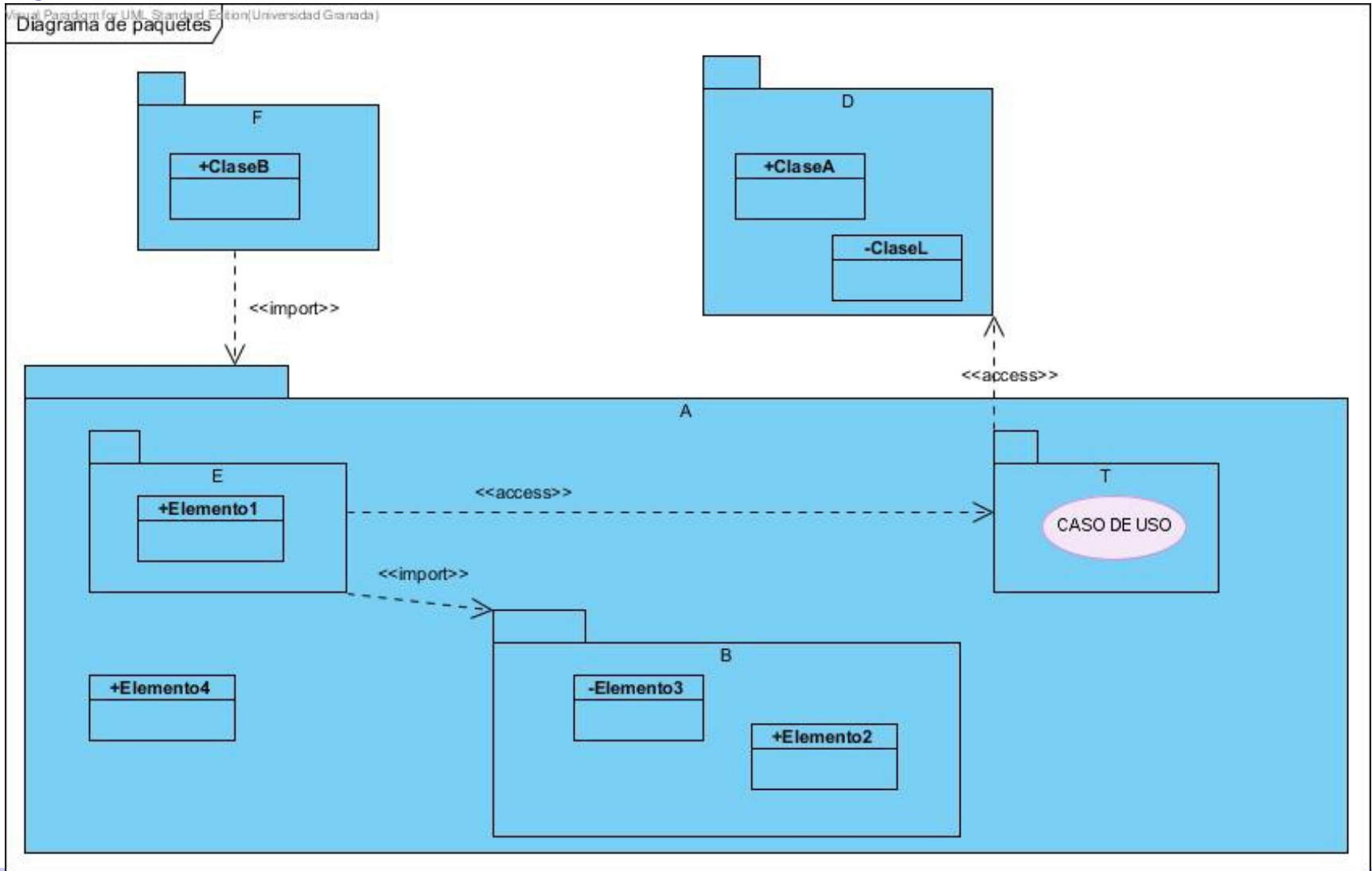
**Relaciones de dependencia:** Los paquetes pueden estar relacionados entre sí por una dependencia.

| Tipo dependencia                    | Semántica  |
|-------------------------------------|--|
| <code>&lt;&lt;use&gt;&gt;</code>    | El paquete destino utiliza un elemento público del paquete origen. Cuando no hay estereotipo hay una dependencia de este tipo por defecto. |
| <code>&lt;&lt;import&gt;&gt;</code> | Los elementos públicos del espacio de nombres del paquete origen se añaden como elementos públicos al espacio de nombres del destino.      |
| <code>&lt;&lt;access&gt;&gt;</code> | Los elementos públicos del espacio de nombres del paquete origen se añaden como elementos privados al espacio de nombres del destino.      |



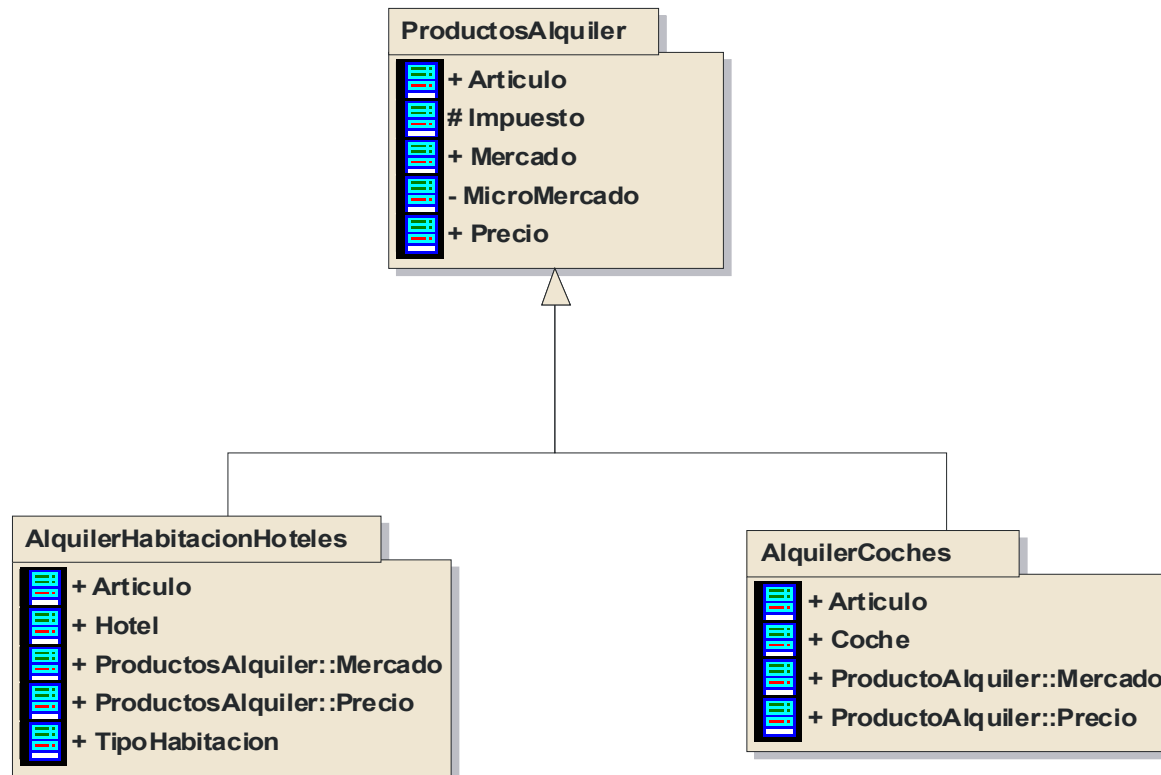
# Diagramas de paquetes

## Ejemplo de Relaciones de dependencia:

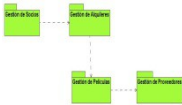


# Diagramas de paquetes

**Generalización de paquetes:** los paquetes hijo más especializados heredan los elementos públicos de su paquete padre. Los paquetes hijo pueden **añadir** nuevos elementos y pueden **anular** elementos en el paquete padre al proporcionar un elemento alternativo con el mismo nombre.



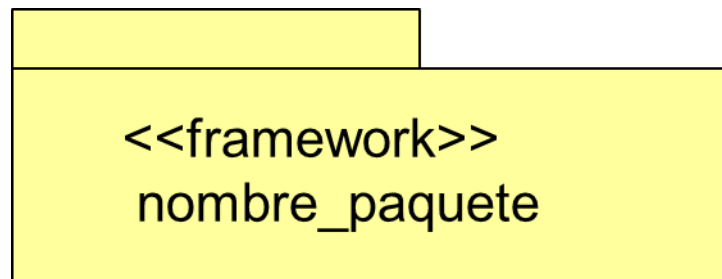


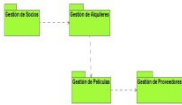


# Diagramas de paquetes

**Esteretipos:** UML proporciona dos estereotipos estándar para adaptar la semántica de los paquetes a fines específicos.

| Estereotipo      | Semántica   |
|------------------|---|
| <<framework>>    | Un paquete que contiene elementos de modelo que especifican una arquitectura reutilizable.    |
| <<modelLibrary>> | Un paquete que contiene elementos que están pensados para ser reutilizados por otros paquetes |





# Diagramas de paquetes

**Reglas para la elaboración de paquetes:** Un paquete bien estructurado debe:

- Ser lo más cohesivo posible.
- Estar poco acoplado con otros paquetes.
- Contener un grupo de clases relacionadas estrechamente.
- Poseer un conjunto equilibrado de elementos.
- Evitar dependencias cíclicas con otros paquetes