

Modelado Avanzado con Casos de Uso

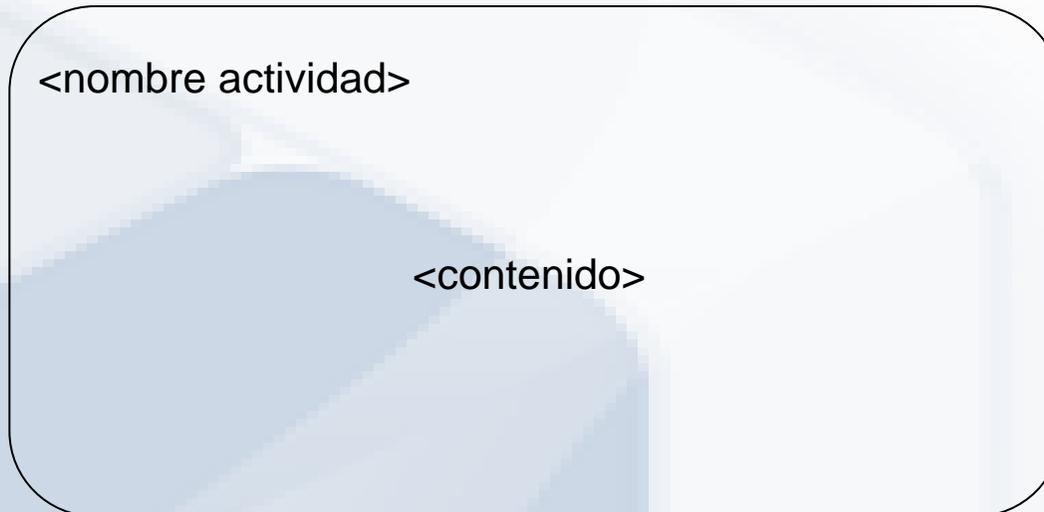


Especificación Gráfica de Casos de Uso

- Una simple secuencia de acciones no puede describir adecuadamente la **riqueza de situaciones** que se pueden presentar en un caso de uso (alternativas, excepciones...).
- Posibles soluciones:
 - Complicar la **descripción textual** de la secuencia de acciones.
 - Emplear una **descripción gráfica** mediante diagramas de actividad.
- Un **diagrama de actividad** representa:
 - El **flujo de acciones** (secuencial, alternativo o concurrente)
 - de un **caso de uso** o de una **operación** de una clase.
- Elementos principales:
 - **Acciones**: cada una de las unidades en que se descompone la actividad.
 - **Transiciones**: conexión entre el fin de una acción y el comienzo de otra.
 - **Condiciones**: deben ser expresiones booleanas mutuamente exclusivas.

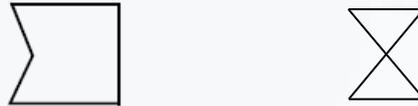
Actividades

- Especificación de comportamiento (parametrizado) cuyos elementos individuales son las acciones.
- El flujo de ejecución de una actividad se modela con nodos (**acciones y nodos de control**) conectados a través de ramas (**transiciones**).



Acciones

- Unidad abstracta fundamental de especificación de comportamiento.
- La ejecución de una acción representa alguna transformación o procesamiento en el sistema modelado.
- Tipos:
 - Aceptación de evento: acción que espera por la ocurrencia de un evento.



- Envío de señal: acción que crea una señal y la transmite.

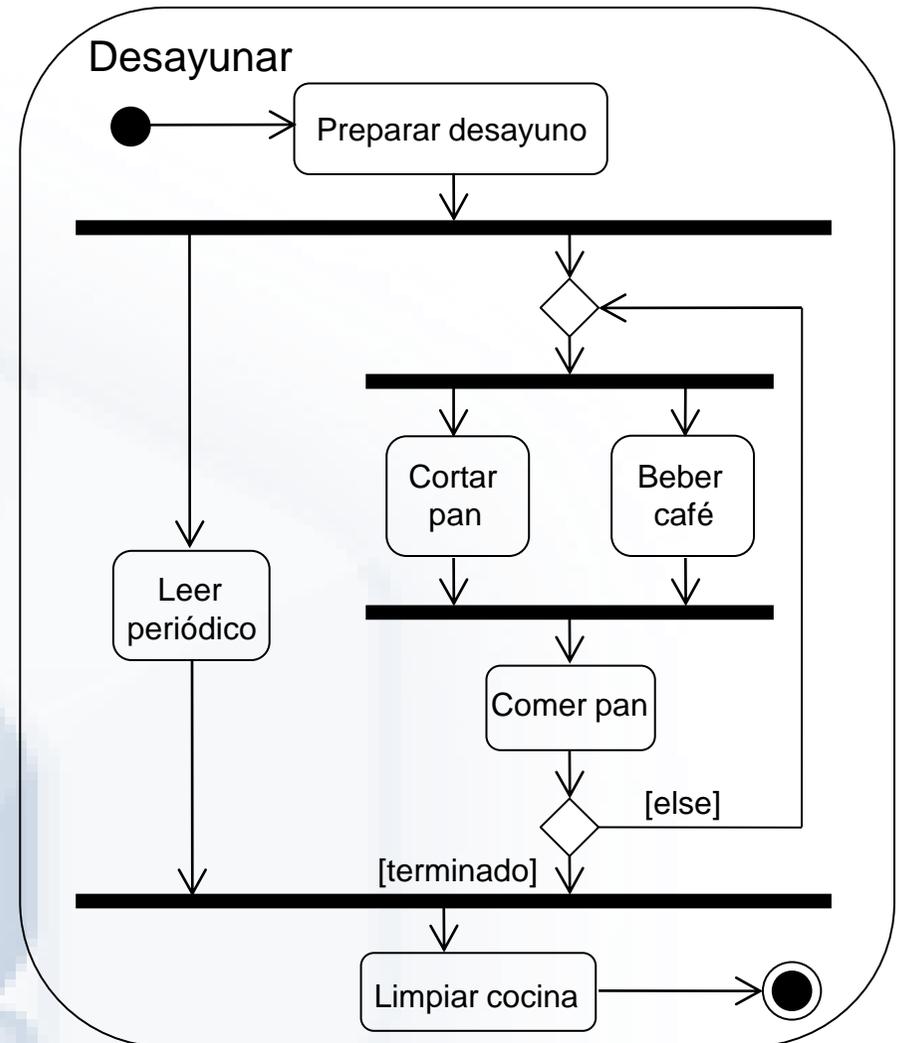
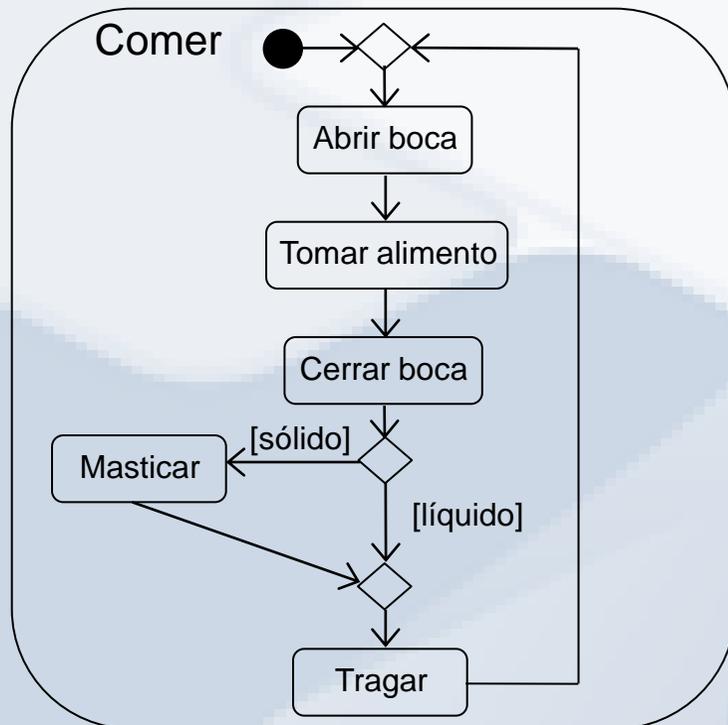


- Ejecución de comportamiento: especificación del comportamiento que debe ser llevado a cabo.



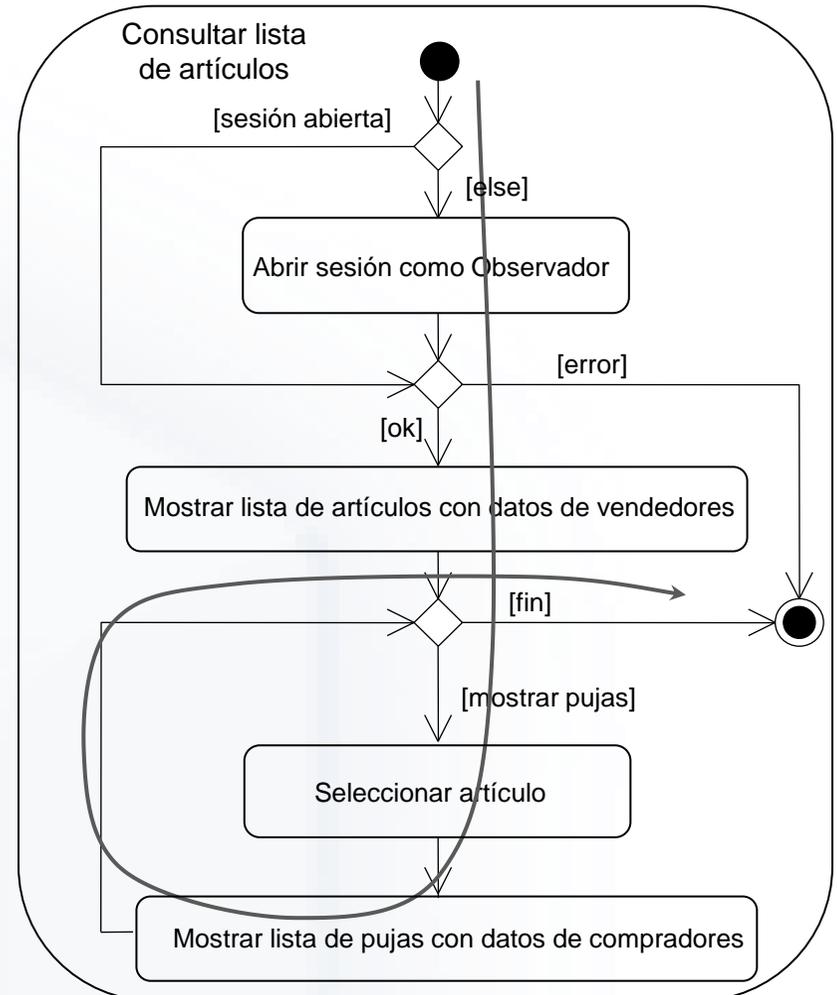
Diagramas de Actividad

- Acciones y transiciones.
- Estados inicial y final.
- Decisiones y ramas alternativas.
- Sincronización de ramas concurrentes.

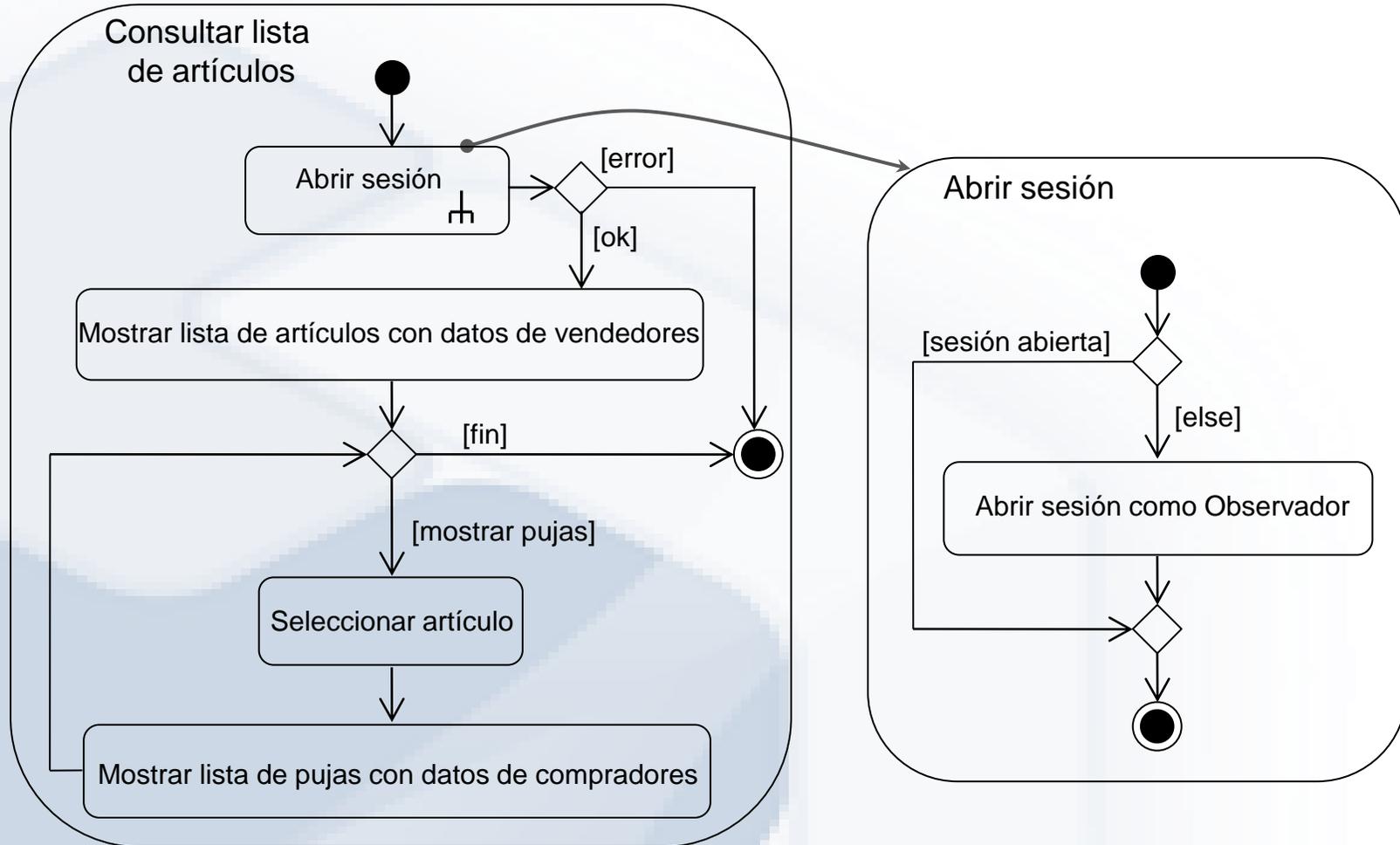


Correspondencia entre las Especificaciones Textual y Gráfica

- **Nombre:** Consultar lista de artículos
- **Actores:** Observador
- **Objetivo:** Obtener lista de artículos con datos de vendedores, y lista de pujas de un artículo con datos de compradores.
- **Precondiciones:**
- **Postcondiciones:**
- **Escenario básico:**
 - Abrir sesión como Observador.
 - Mostrar la lista de artículos con los datos de los vendedores.
 - Opcionalmente:
 - Seleccionar un artículo.
 - Mostrar la lista de pujas del artículo con los datos de los compradores.

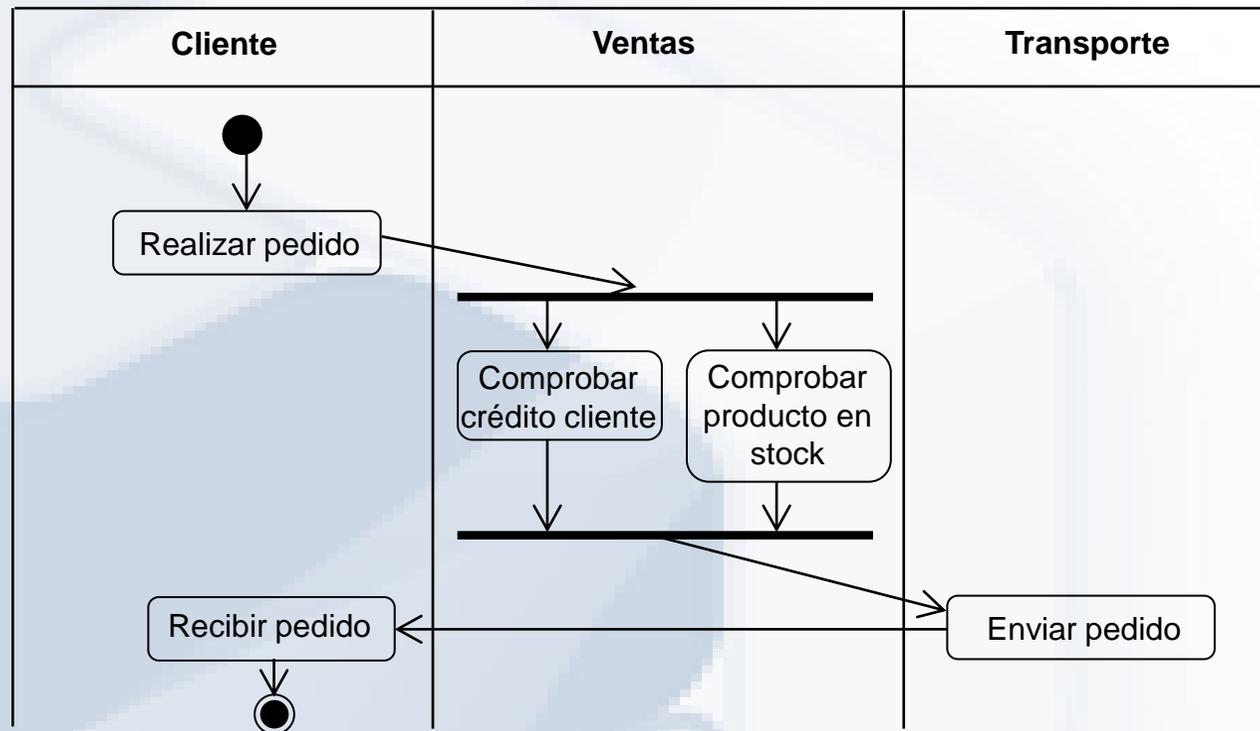


Subactividades



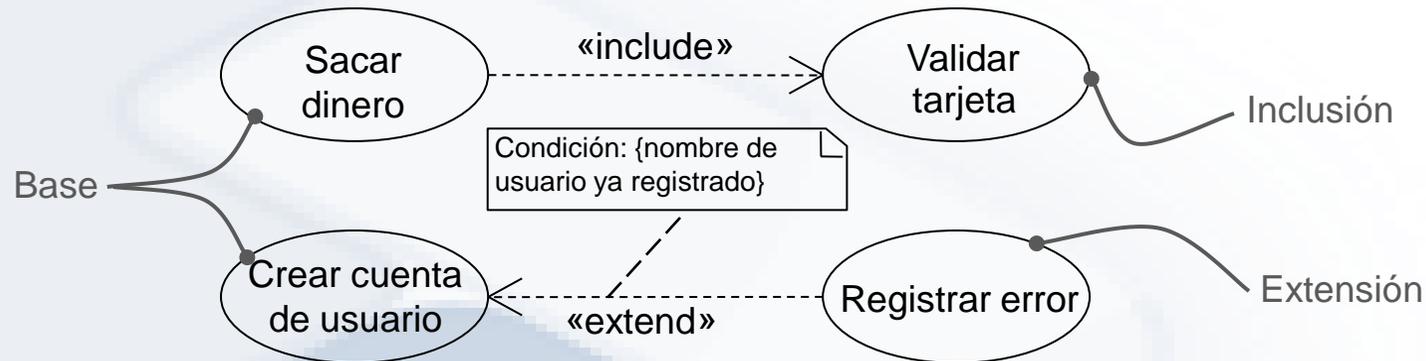
Swimlanes

- Partición con nombre que permite agrupar un conjunto de acciones.
- Permite especificar de manera visual qué recurso o rol es responsable de cada acción que forma parte del diagrama de actividad.



Include y Extend

- Es posible definir relaciones entre casos de uso:
 - «**include**»: para describir un comportamiento común reutilizable.
 - «**extend**»: para describir una variante del comportamiento base (es decir, sólo se inserta el comportamiento cuando se cumple una condición).



- Significado problemático en UML:
 - No está clara la **diferencia** entre ambas (reutilización / inserción).
 - No siempre encajan con la definición como “**unidad coherente de funcionalidad**”.
 - Pueden llevar por error al modelado de **procesamiento secuencial**.
- **En este curso se podrá utilizar «include», pero con mucho cuidado.**

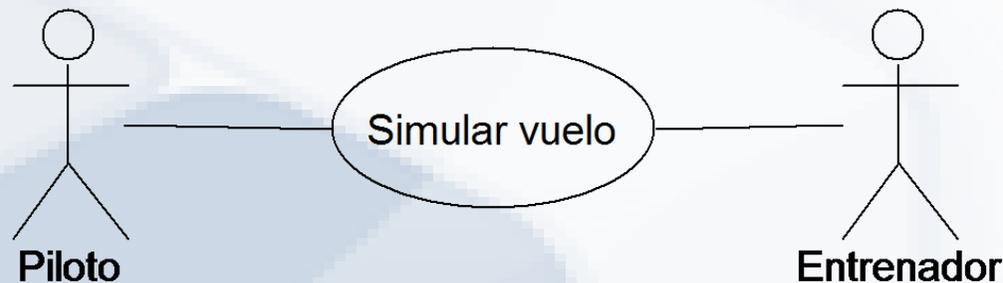
Acceso al Sistema

- Situaciones típicas que tienen en común una **secuencia de acciones previa** a la ejecución de los demás casos de uso:
 - Identificación o validación de usuarios.
 - Inicio (y cierre) de sesión.
 - Abrir un documento sobre el que se van a realizar diversas operaciones.
 - etc.
- Posibles soluciones (y problemas asociados):
 - Caso de uso aparte usado como «**include**»
 - Complica innecesariamente el diagrama de casos de uso.
 - Es parte de los demás casos de uso sólo la primera vez que se ejecutan.
 - Caso de uso aparte cuyo resultado es usado como **precondición para los demás**.
 - No representa una interacción “completa”, una funcionalidad en sí misma con un objetivo “independiente” de los objetivos de los demás casos de uso.

→ Secuencia opcional de acciones en el **diagrama de actividad** del caso de uso.
- **Puede reutilizarse definiéndola como subactividad.**

Actores Cooperativos

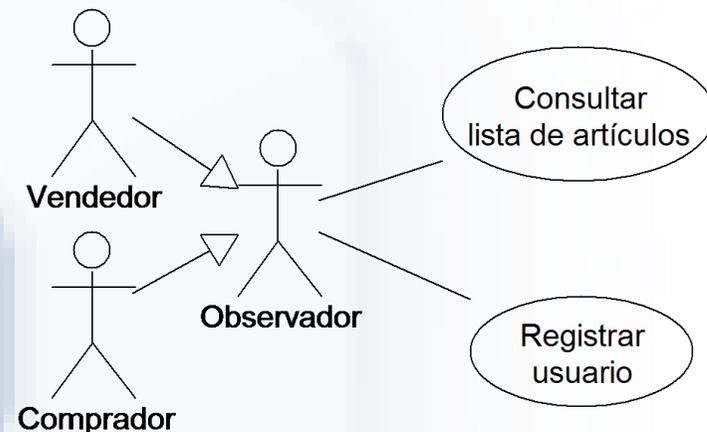
- ¿Qué significa conectar varios actores a un mismo caso de uso?
 - El caso de uso puede requerir la participación de **varios actores**, y
 - cada actor asociado a un caso de uso representa un **rol distinto**, y
 - uno de los actores será el **iniciador** del caso de uso, y
 - los actores **cooperan** entre todos para realizar el objetivo del caso de uso.



- **Incorrecto:** pretender que es “el mismo caso de uso” para distintos actores, que lo ejecutan de modo independiente.

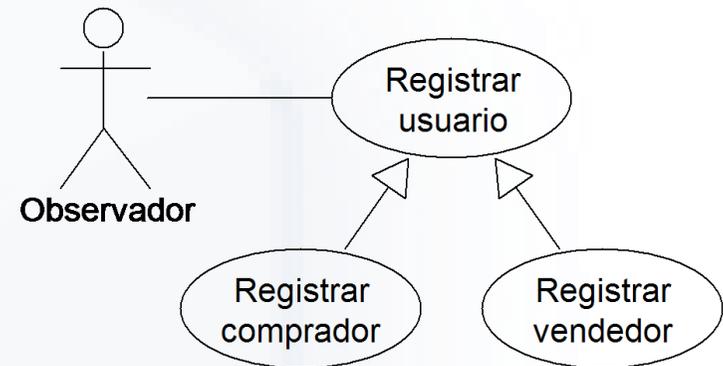
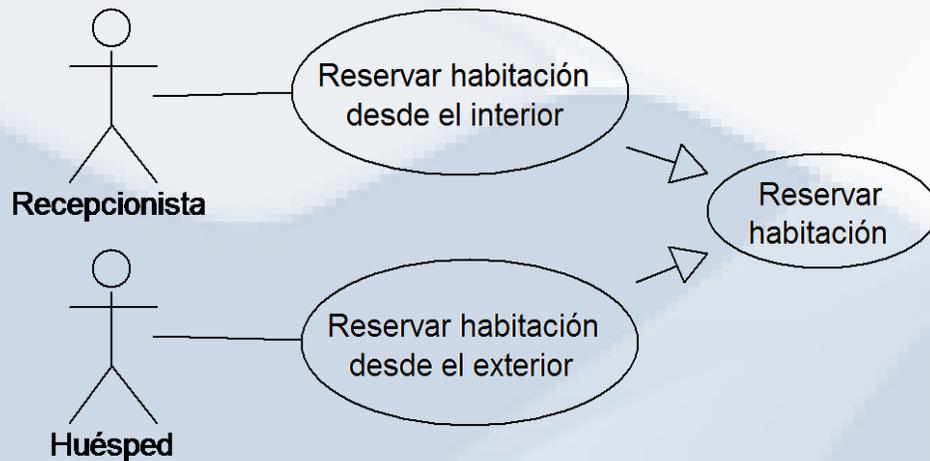
Generalización de Actores

- Problema: Categoría de usuarios A requiere el mismo servicio que otra categoría de usuarios B.
- Posibles soluciones:
 - Hay usuarios concretos que adoptan los dos roles (**clasificación multiple**).
 - El actor A es subtipo del actor B (**todo A es un B**).
 - Los actores A y B son subtipos del actor C (problema: **proliferación de actores**).
- Posibles errores:
 - Asociar los dos actores con **el mismo caso de uso** (serían actores cooperativos).
 - Que no requieran exactamente **el mismo servicio** (no sería el mismo caso de uso).



Generalización de Casos de Uso

- Un caso de uso puede tener especializaciones con el fin de:
 - Añadir **asociaciones** con actores que requieran distintas variantes.
 - Añadir o sobrescribir **pre/postcondiciones** y **acciones**.
- Significado problemático en UML
- **No usar la generalización de casos de uso si se puede evitar**



Realización de Casos de Uso

- El modelo de casos de uso captura los **requisitos funcionales** del sistema y recoge el **vocabulario del dominio**.
- Los requisitos funcionales son estructurados y formalizados usando el mismo vocabulario en el **modelo de análisis**, que es refinado y completado en el **modelo de diseño** con soluciones tecnológicas que aportan aspectos de implementación.
- Para realizar un caso de uso con metodología orientada a objetos se requiere una **colaboración** (tanto en análisis como en diseño), es decir, un conjunto de objetos que cooperan para conseguir el objetivo del caso de uso:
 - **Modelo estático** de la colaboración: clases y asociaciones.
 - Estructura de datos requerida por el caso de uso.
 - **Modelo dinámico** de la colaboración: objetos y mensajes.
 - Interacciones entre objetos para cumplir el objetivo del caso de uso.