

# *Fundamentos de Ingeniería del Software*



*Capítulo 3. Análisis de Requisitos  
Introducción a los casos de uso*

# *Cap 3. Análisis de Requisitos Estructura*



1. Actividades iniciales.
2. Técnicas de recogida de la información.
3. Requisitos y análisis de requisitos.
4. Actividades generales de análisis de requisitos.
5. Documentos de especificación de requisitos.
6. Análisis estructurado.
7. Introducción a los casos de uso.
8. Prototipado.

# *7. Introducción a los casos de uso. Estructura*



7.1. Introducción

7.2. Diagramas de casos de uso

7.3. Actores

7.4. Casos de uso

- Descripción

- Relaciones entre casos de uso

- Granularidad de los casos de uso

7.5. Escenarios y casos de uso

7.6. Especificación de requisitos con casos de uso

7.7. Desarrollo dirigido por los casos de uso

# *Introducción a los casos de uso. Bibliografía*

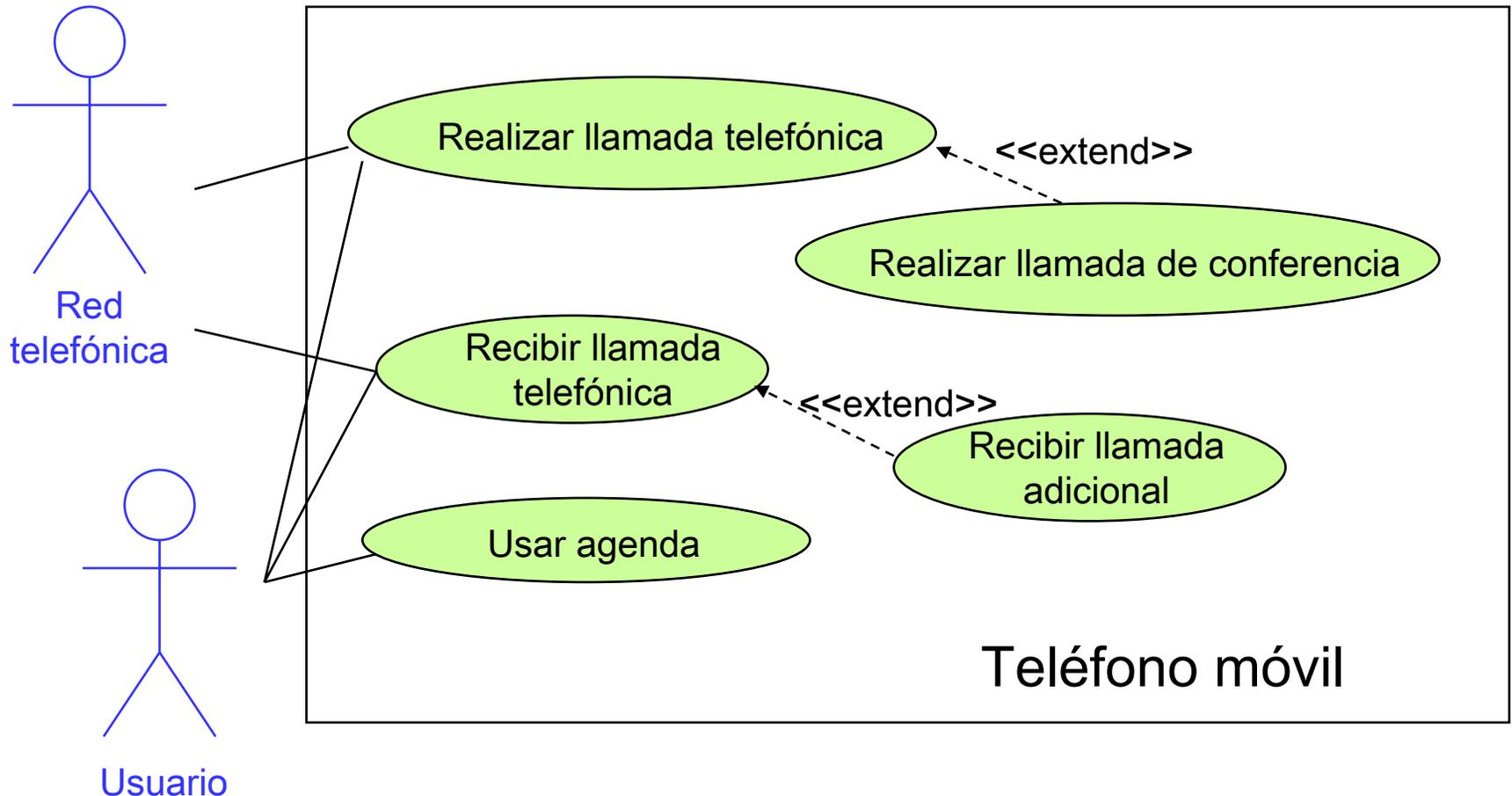
- "UML Gota a gota". M. Fowler, K. Scott. Ed. Addison-Wesley Longman. 1999. (Cap. 3)
- "UML y Patrones" (2ª Edición). C. Larman. Ed. Prentice-Hall. 2003. (Cap. 6)
- "El lenguaje unificado de modelado". G. Booch et al. Ed. Addison-Wesley. 1999. (Caps. 16 y 17)
- "Object-Oriented Software Engineering. A Use Case Driven Approach". I. Jacobson et al. Ed. Addison-Wesley. 1992.
- "Applying Use Cases". G. Schneider, J. P. Winters. Ed. Addison-Wesley. 1998.

# 7.1. *Introducción*



- Técnica de recolección y especificación de requisitos.
- Fáciles de comprender/validar por los usuarios.
- Guían todo el proceso de desarrollo.
- Ayudan a la planificación/desarrollo incremental.
- Tradicionalmente ligados a la OO
  - pero no obligatorio
- Ayudan a determinar la interfaz de usuario.

# Diagrama de casos de uso. Ejemplo



# *Éxito de los casos de uso*

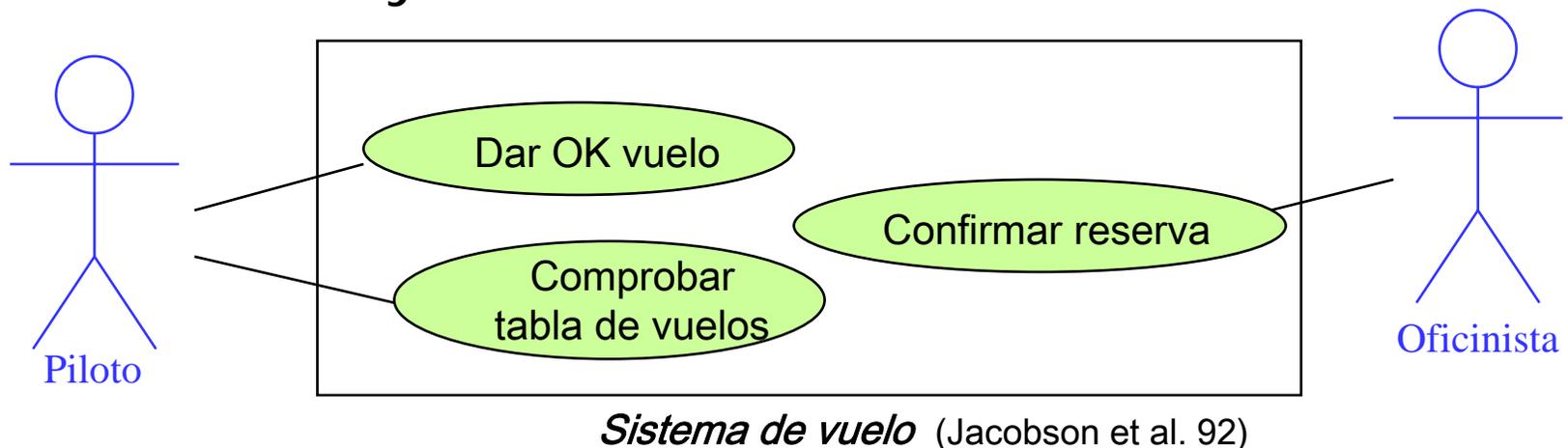


- Concebidos por I. Jacobson - Objectory/OOSE (Jacobson et al. 92)
- Se han asentado como una de las principales técnicas de especificación de requisitos.
- Presentes en casi cualquier nuevo método de desarrollo de software.
- Incluidos en UML y Métrica 3.

## 7.2. Diagramas de casos de uso

Elementos:

- **Actores:** roles que juegan los usuarios con respecto al sistema.
- **Casos de uso:** interacciones típicas entre usuarios y el sistema.



## 7.3. Actores

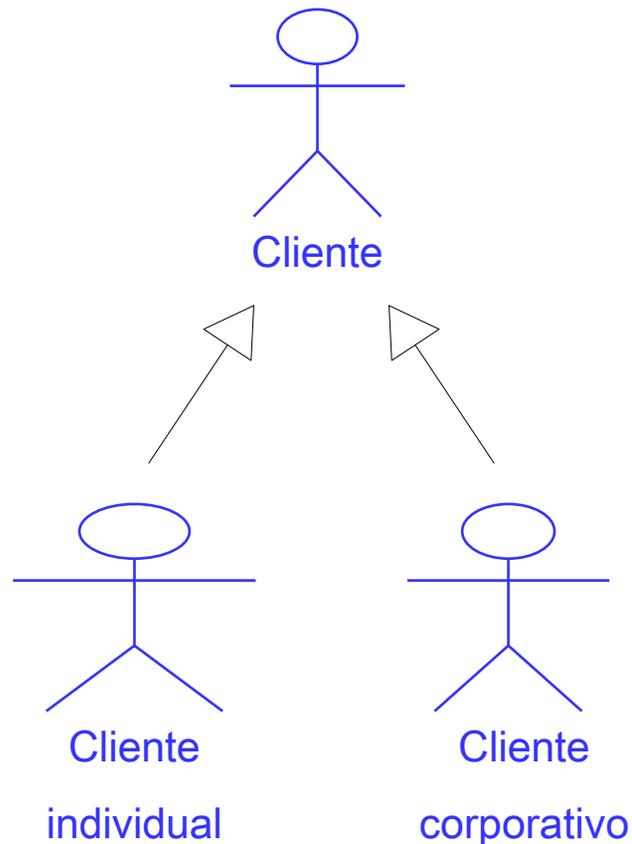


- Inician la ejecución de los casos de uso.
- No tienen que ser personas necesariamente.
- Un mismo rol puede ser jugado por más de un usuario.
- Un usuario puede jugar más de un rol.

# Actores (II)

■ En UML, se pueden generalizar actores.

■ p.ej.



# Actores (III)



- Ayudan a capturar los casos de uso  
*...aunque algunos casos de uso no tienen actores (de inicio) asociados...*  
p.e. "enviar factura" (*nadie la pide*) (Fowler 99)

Dos posibles soluciones: (Schneider Winters 98)

- que el sistema pueda iniciar el caso de uso (no permitido en UML, pero estos autores creen que sería útil)
- crear un actor como "Tiempo" (para un caso de uso que se debe iniciar automáticamente cada cierto intervalo de tiempo) o "Sistema"

# Actores (IV)



- Ayudan a configurar el sistema para cada usuario
  - crear *profiles* o perfiles de usuario
- Ayudan a tomar soluciones de compromiso durante el desarrollo
  - se observa quién (qué actor) necesita cada caso de uso.

# *Encontrar los actores*



- *¿Qué se considera un actor?*

⇒ podemos preguntarnos  
*¿Por qué se construye el sistema?*

- Los actores “ganan valor” con la ejecución del caso de uso (actor primario del caso de uso),

- o pueden sólo “participar” en él (actores secundarios del caso de uso)

## 7.4. Casos de uso



- Capturan una función visible para el usuario.
- Consiguen un objetivo para el usuario del sistema.
- Por cada caso de uso:
  - Un camino básico
  - Caminos alternativos (*describir tantos como sea posible para aumentar la robustez del sistema*)
- Caso de uso ↔ Descripción en lenguaje natural
- Descripciones breve, informal y completa (Larman 03)
- También con diagramas de actividad y de secuencia

# Casos de uso

## Descripción textual

- Un escenario primario y varios secundarios. P.ej.:

### Caso de uso Ordenar pedido

(Schneider Winters 98)

**Precondición:** Un usuario válido ha entrado en el sistema

#### Flujo de eventos:

##### *Camino básico*

1. El caso de uso comienza cuando el usuario selecciona Ordenar pedido.
2. El usuario introduce su nombre y dirección.
3. Si el usuario sólo introduce el código postal, el sistema mostrará la ciudad y provincia.
4. El usuario introduce los códigos de los productos deseados.
5. El sistema proporcionará la descripción y el precio de cada elemento.
6. El sistema mostrará el coste total de los elementos introducidos hasta el momento.
7. El usuario introduce información de pago mediante tarjeta de crédito.
8. El cliente seleccionará "Enviar".
9. El sistema verificará la información, guardará el pedido como "pendiente" y enviará la información de pago al sistema de contabilidad.
10. Cuando el pago es confirmado, el pedido se marca "Confirmado", un ID de pedido se manda al usuario y termina el caso de uso.

##### *Caminos alternativos*

En el paso 7, si cualquier información es incorrecta, el sistema pedirá al usuario corregir la información.

**Post-condición:** El pedido ha sido guardado en el sistema y marcado como "Confirmado".

# Casos de uso

## Descripción textual (II)

- Disponibles muchas plantillas  
P.ej., la plantilla de Coleman:

<b>Caso de Uso</b>	Ordenar Fabricación
<b>Descripción</b>	Se crearán órdenes de trabajo para cada producto solicitado en el pedido, y serán enviadas al jefe de producción para su planificación.
<b>Actores</b>	Jefe técnico
<b>Asunciones</b>	- Es viable la fabricación de cada producto solicitado en el pedido. - Existe una plantilla de fabricación para cada producto solicitado.
<b>Pasos</b>	1. REPETIR 1.1 Obtener un producto del pedido. 1.2 Buscar la plantilla de fabricación asociada al producto. 1.3 Crear la orden de trabajo. 1.4 Almacenar la orden de trabajo con el estado <i>pendiente</i> .
<b>Variaciones</b>	-
<b>Req. No Funcionales</b>	-
<b>Cuestiones</b>	-

# *Encontrar los casos de uso*



- Es útil encontrar primero los actores
- Se puede diferenciar entre (Fowler 99):
  - Objetivos del usuario:  
“formatear un documento”
  - Interacciones del sistema:  
“def. un estilo”, “mover un estilo a otro doc.”
- Guía útil: primero buscar los objetivos del usuario, y luego cubrir cada objetivo con interacciones del sistema.

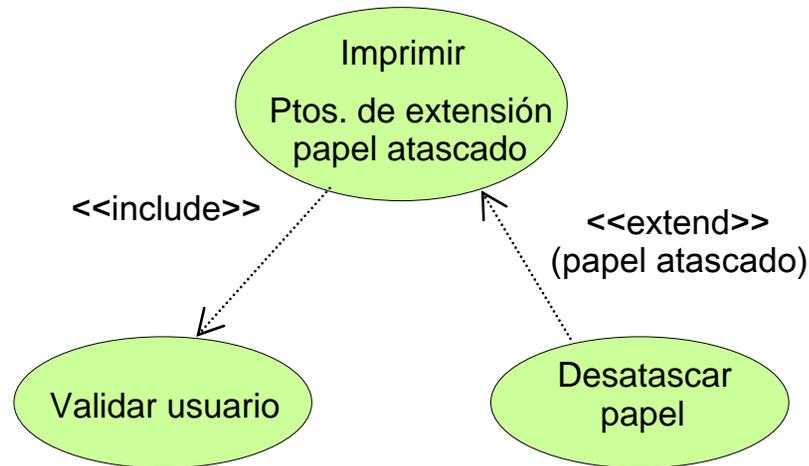
# *Relaciones entre casos de uso*

*(en UML, Booch et al. 99)*



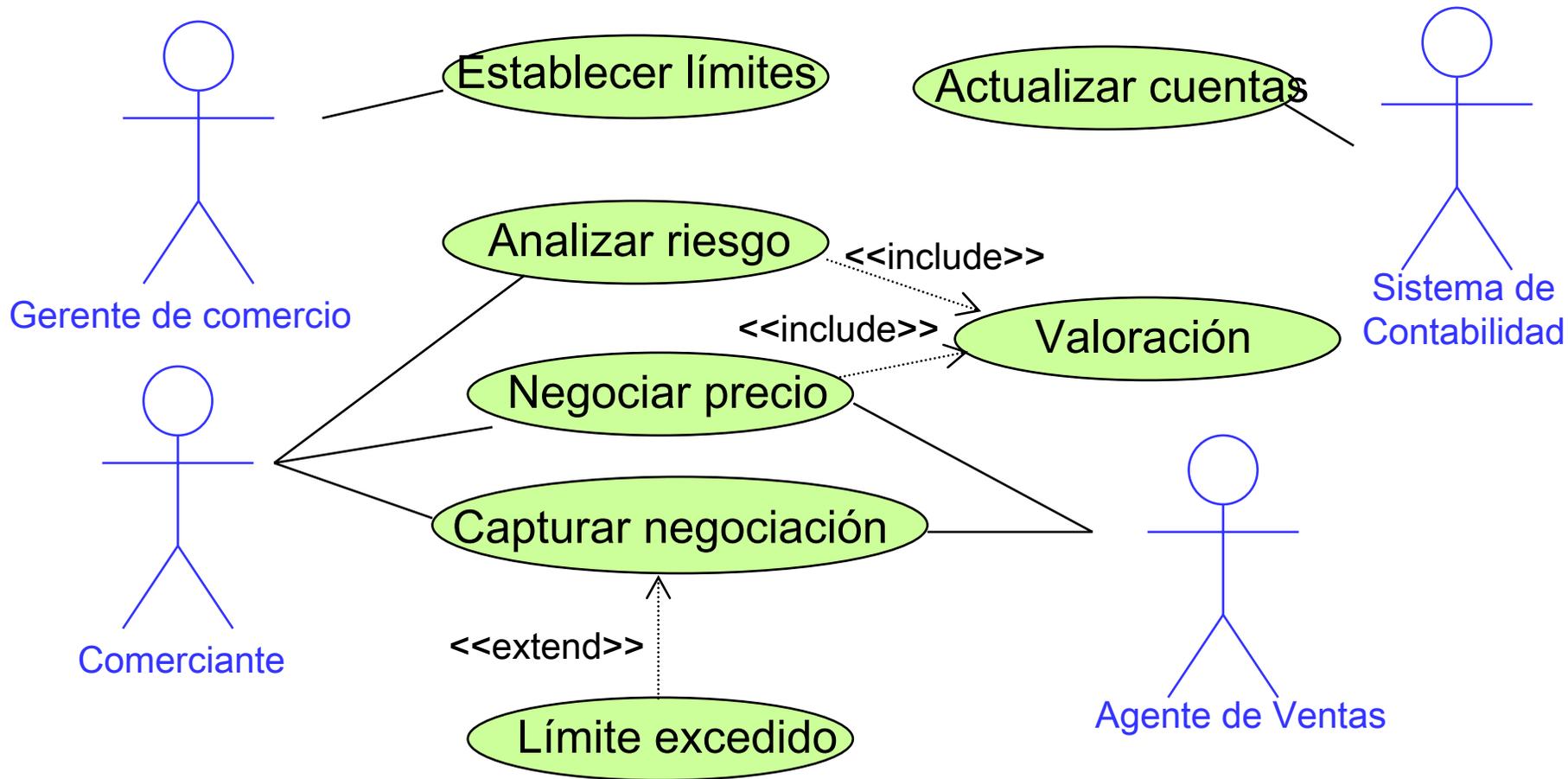
- **«*extend*»**: el caso de uso que extiende realiza una acción en un punto del caso de uso extendido, si se cumple una condición.  
Denota un “punto de extensión”.  
Sirve para representar las condiciones de error y poco usuales, que podrían oscurecer el caso de uso base.
- **«*include*»**: se factorizan acciones que se utilizan en más de un caso de uso (se incluye siempre).

# Relaciones entre casos de uso (en UML, Booch et al. 99) (II)



⇒ En UML también existe la generalización

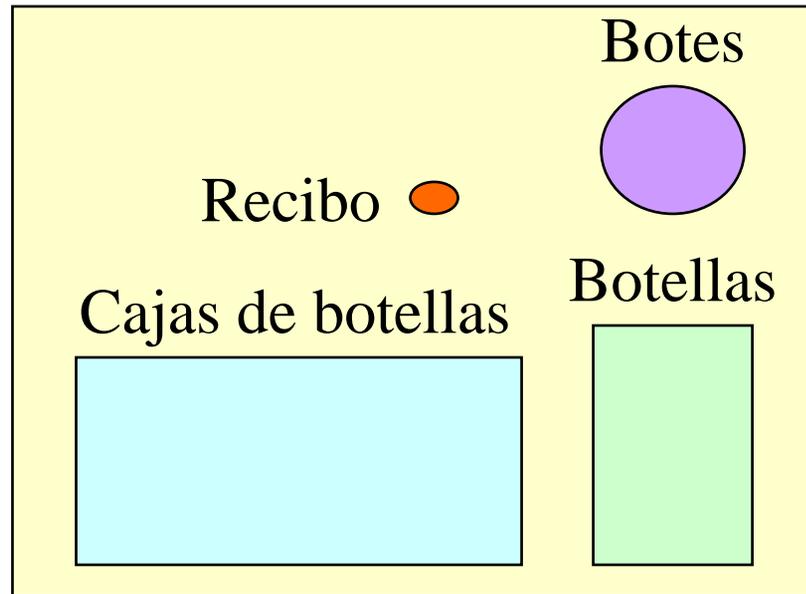
# *include y extend. Ejemplo*



(Fowler 99)

# Casos de uso. Ejemplo

## ■ Máquina de reciclado:



Extraído de (Jacobson et al. 92) p.150

# *Casos de uso. Ejemplo (II)*



## Máquina de reciclado de botes, botellas y envases de plástico

- Como cada ítem tiene precios y dimensiones distintas el sistema debe identificar el tipo de ítem que acaba de recibir
- El sistema registra el número de ítems recibidos y, si el cliente pide un recibo, imprime el número de ítems recibidos, su tipo, los precios parciales y el total que será devuelto al cliente en la caja, al presentar ese recibo impreso
- Un operador puede, al final del día, solicitar un listado de todos los ítems recuperados ese día
- El operador también puede cambiar la información del sistema (precios, tipos, etc.)



# *Granularidad de los casos de USO*



## ■ Pueden ser grandes o pequeños:

### ■ p.ej. en un procesador de textos...

- “poner texto en negrita” •
- “crear tabla de contenidos” ●
- “formatear el documento” ●●

## ■ *¿Qué criterio adoptar?*

### ■ Proyecto 10 personas/año $\Rightarrow$ Valores esperados :

- Según I. Jacobson: 20 casos de uso.
- Según M. Fowler: 100 casos de uso.

## 7.5. *Escenarios y Casos de Uso*



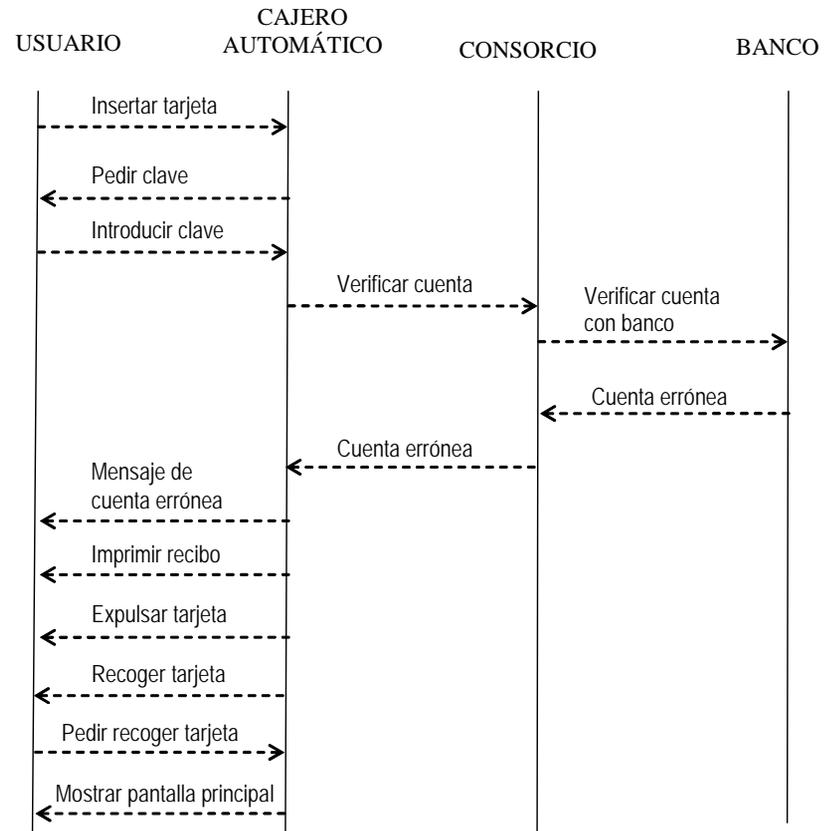
- “Interacción típica entre el usuario y el sistema”
- Distintas acepciones:
  - (a veces) Escenario = Caso de uso
  - En UML,  
escenario = “camino dentro de un caso de uso”  
(una combinación especial de condiciones dentro de un caso de uso)

# Escenarios y Casos de Uso (II)



Interactuar con cajero automático

CASO DE USO



ESCENARIO

## 7.6. Especificación de requisitos con casos de uso

- El SRS (*Software Requirements Specification*) puede estar formado por:
  - Diagrama de casos de uso
  - Modelo del dominio
  - (Para cada caso de uso)
    - Descripción textual (usando una plantilla)
    - Descripciones de las interfaces de usuario
  - Especificación suplementaria
    - Requisitos no funcionales
    - Reglas de negocio
    - Otros requisitos funcionales (no incluidos en los casos de uso)

# 7.7. Desarrollo dirigido por los casos de uso (Jacobson et al. 92)

## Diagrama de casos de uso

