



**SANTA ANA  
Y SAN RAFAEL**

Madrid

# DISEÑO ORIENTADO A OBJETOS

## UD 6 DIAGRAMAS DE COMPORTAMIENTO

# DISEÑO ORIENTADO A OBJETOS

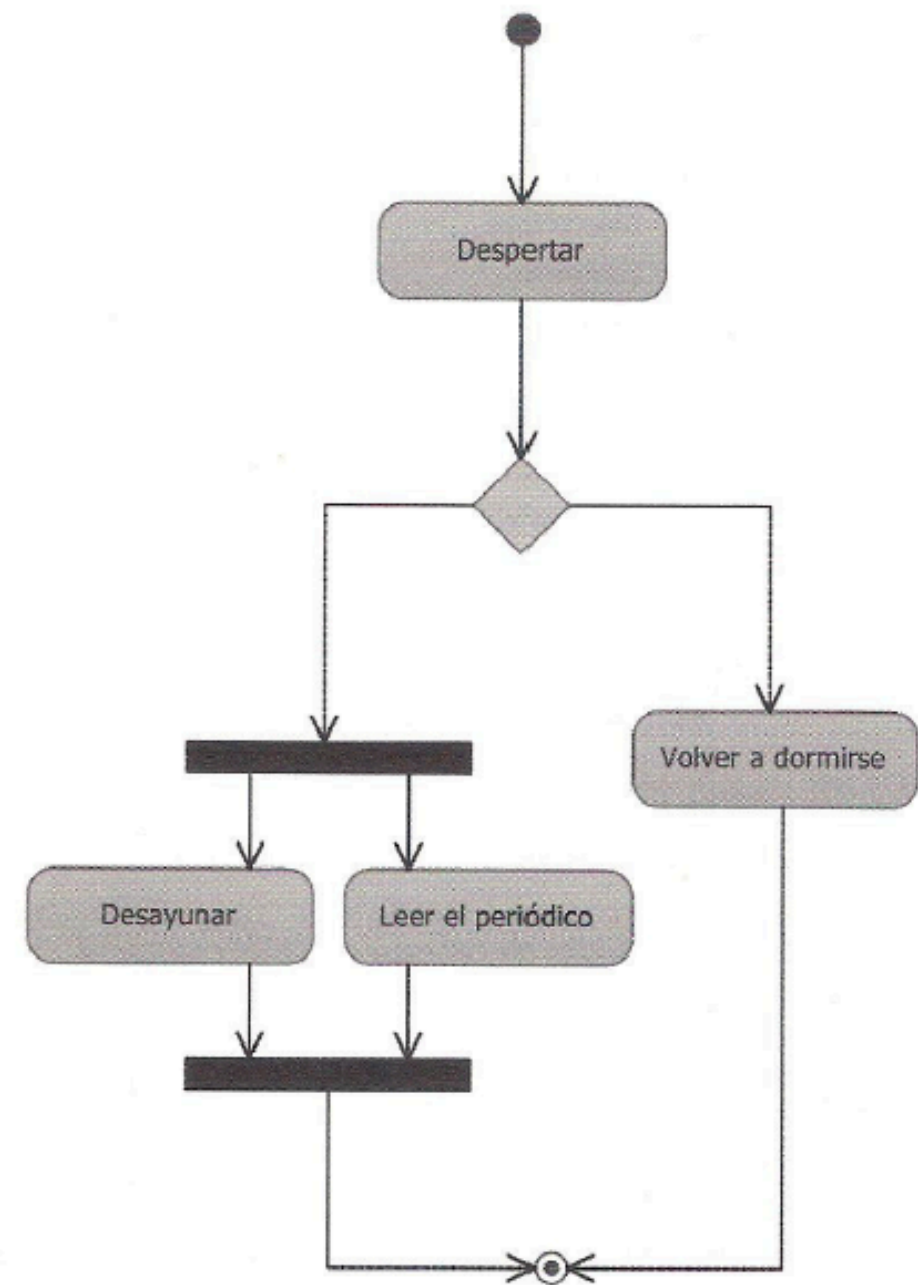
## DIAGRAMAS DE COMPORTAMIENTO

- ✓ Diagramas basados en el estándar unificado **UML**.
- ✓ Representan gráficamente los **procesos** y **formas de uso** de un programa.
- ✓ Están más orientados a representar los **aspectos dinámicos de un sistema**: flujo de mensajes en el tiempo, movimiento de componentes en una red, diferentes estados y operaciones que transcurren en el ciclo de vida de un programa.
- ✓ Los **diagramas de comportamiento** definen cómo interactúan a lo largo del tiempo las clases definidas en los diagramas de clase.
- ✓ Suelen seguir un **modelo jerárquico**, es decir, primero representan el funcionamiento del sistema en varias etapas para posteriormente ir detallando cada etapa o fase.

# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE ACTIVIDAD

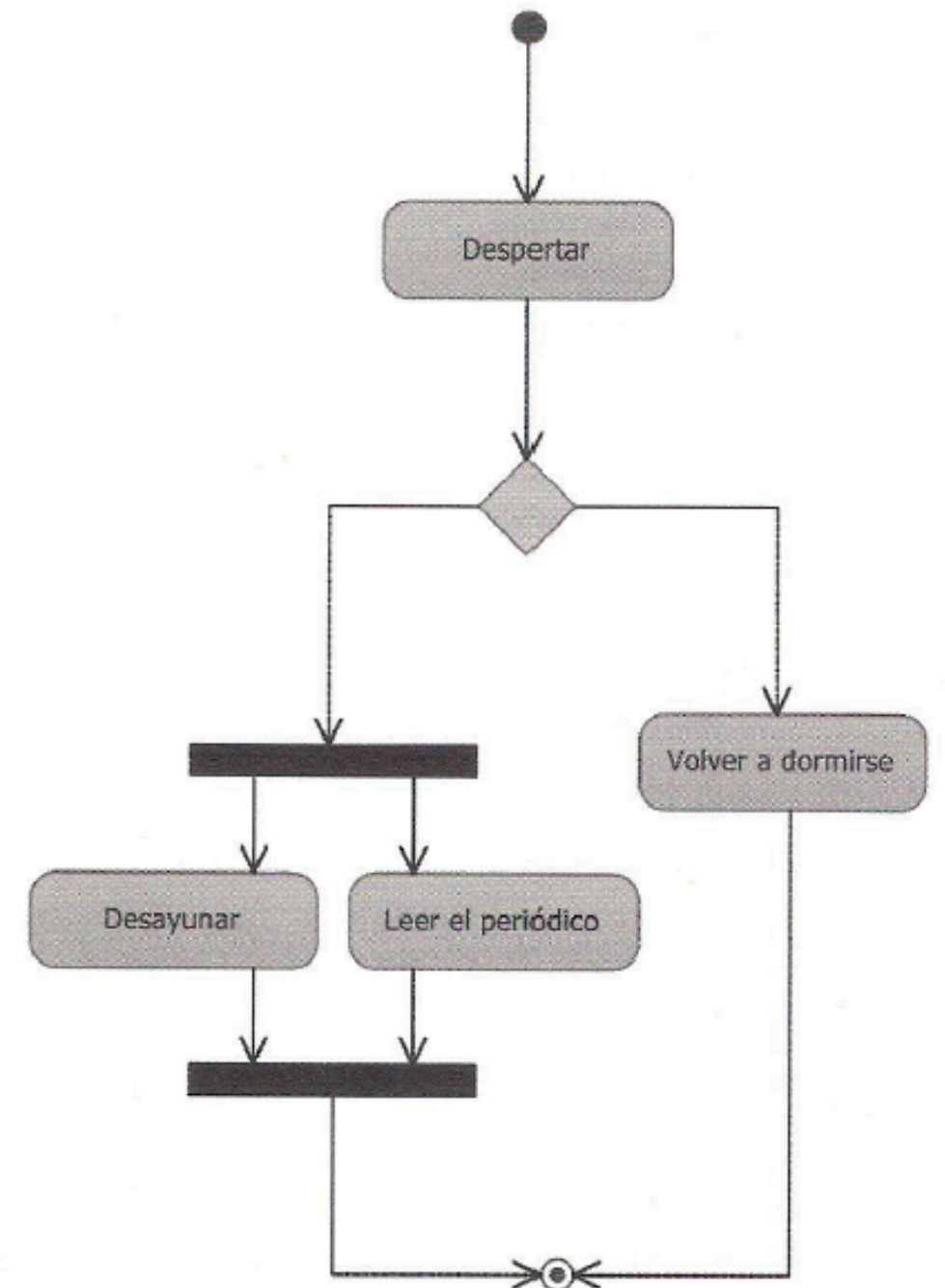
- ✓ Diagramas de comportamiento **UML** más sencillos.
- ✓ Representan **flujos de trabajo** del sistema desde su inicio hasta el fin, incluyendo sus operaciones y componentes.
- ✓ Tienen gran parecido a los diagramas de flujo conocidos.
- ✓ Se componen de:
  - **Estados**: etapas por las que pasa el sistema.
  - **Transiciones**: Líneas de conexión que enlazan estados en una dirección.
  - **Nodos**



# DISEÑO ORIENTADO A OBJETOS

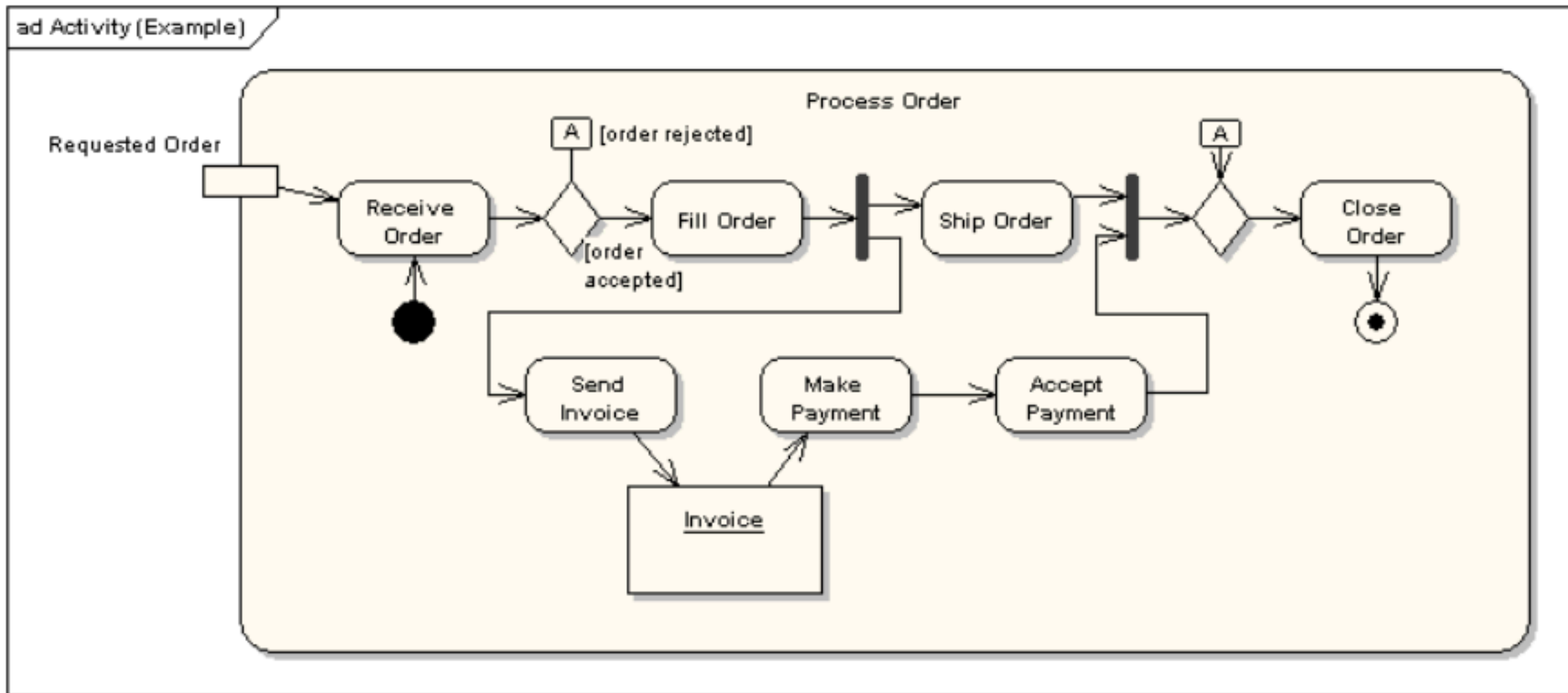
## DIAGRAMAS DE ACTIVIDAD: NODOS

- ✓ Existen diferentes tipos de nodos:
  - **Nodos de decisión:** definen caminos alternativos.
  - **Barras de sincronización:** definen actividades que ocurren de manera asíncrona.
  - **Nodos iniciales y finales** son únicos y deben existir en el diagrama. Indican estado inicial y final del **flujo de trabajo**.
- ✓ Herramientas para trabajar con diagramas de actividad: StartUML, Visual Studio ...



# DISEÑO ORIENTADO A OBJETOS

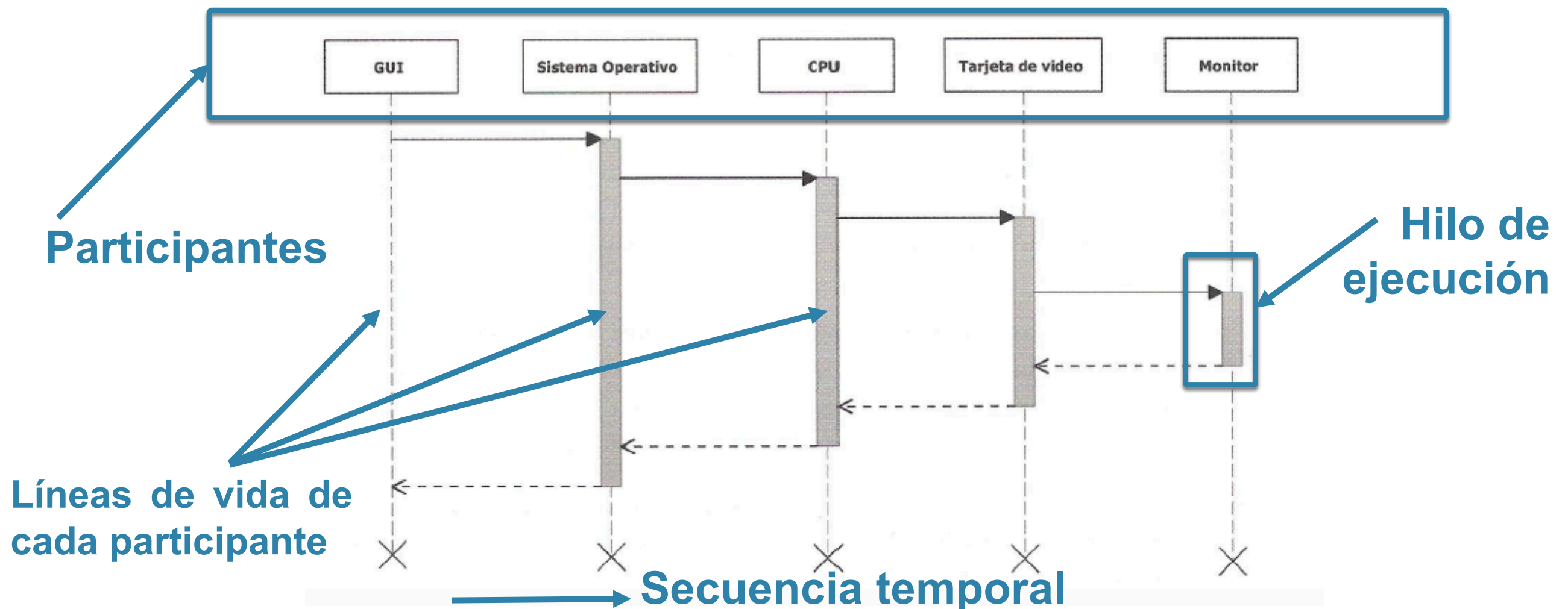
## DIAGRAMAS DE ACTIVIDAD: EJEMPLO



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE SECUENCIA

- ✓ Modelan la **secuencia lógica** a través del tiempo de los mensajes entre instancias.
- ✓ Se estructura mediante **líneas de vida** de cada **participante** en el transcurso secuencial y temporal del programa.



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE SECUENCIA

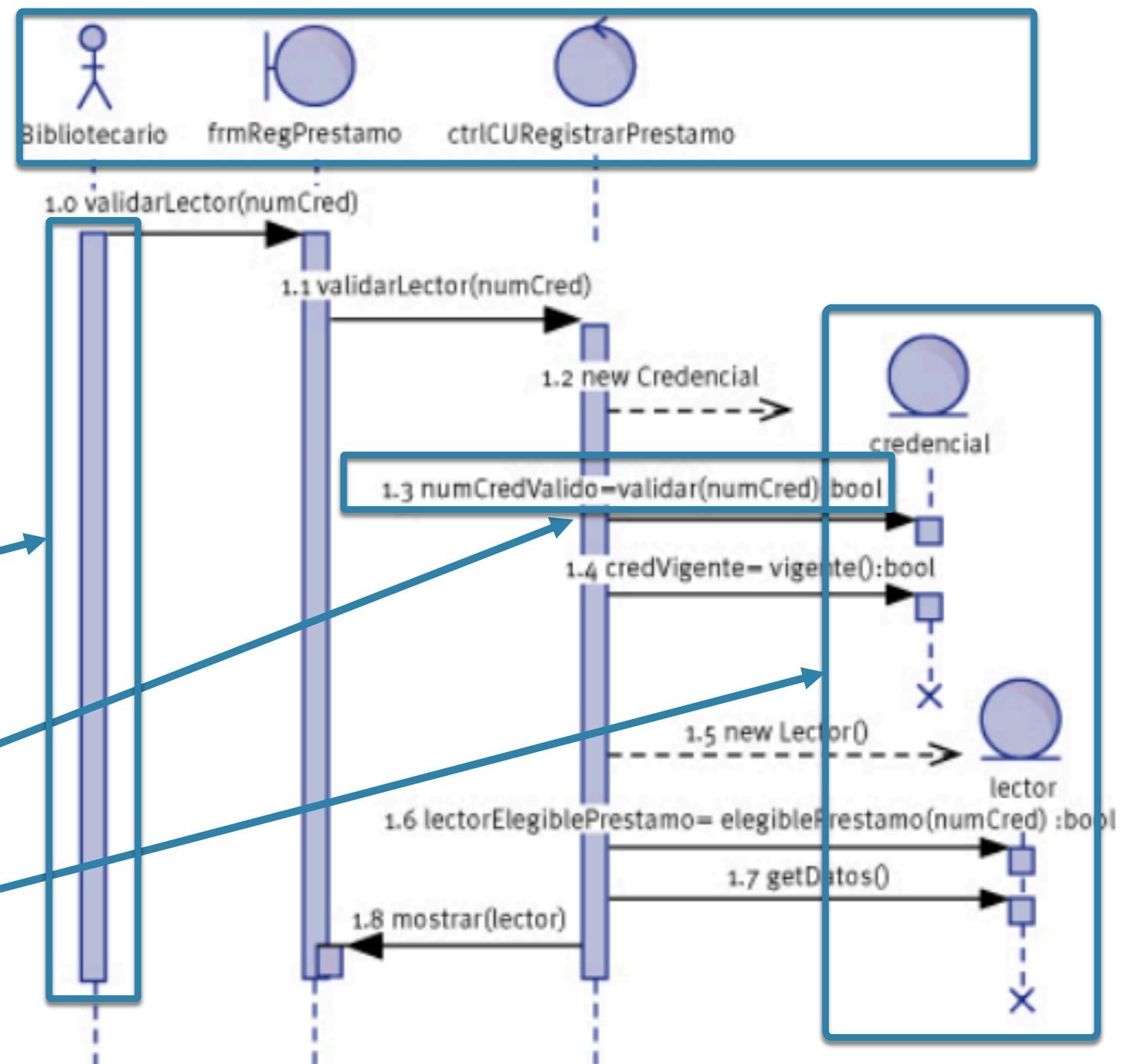
- ✓ Las **líneas de vida** nos permiten visualizar de manera rápida, la cronología de las interacciones, mensajes y participantes a lo largo de la vida del sistema.

**Hilo de ejecución**

**Mensajes**

**Vida más corta de algunos Participantes**

### Participantes





# DISEÑO ORIENTADO A OBJETOS

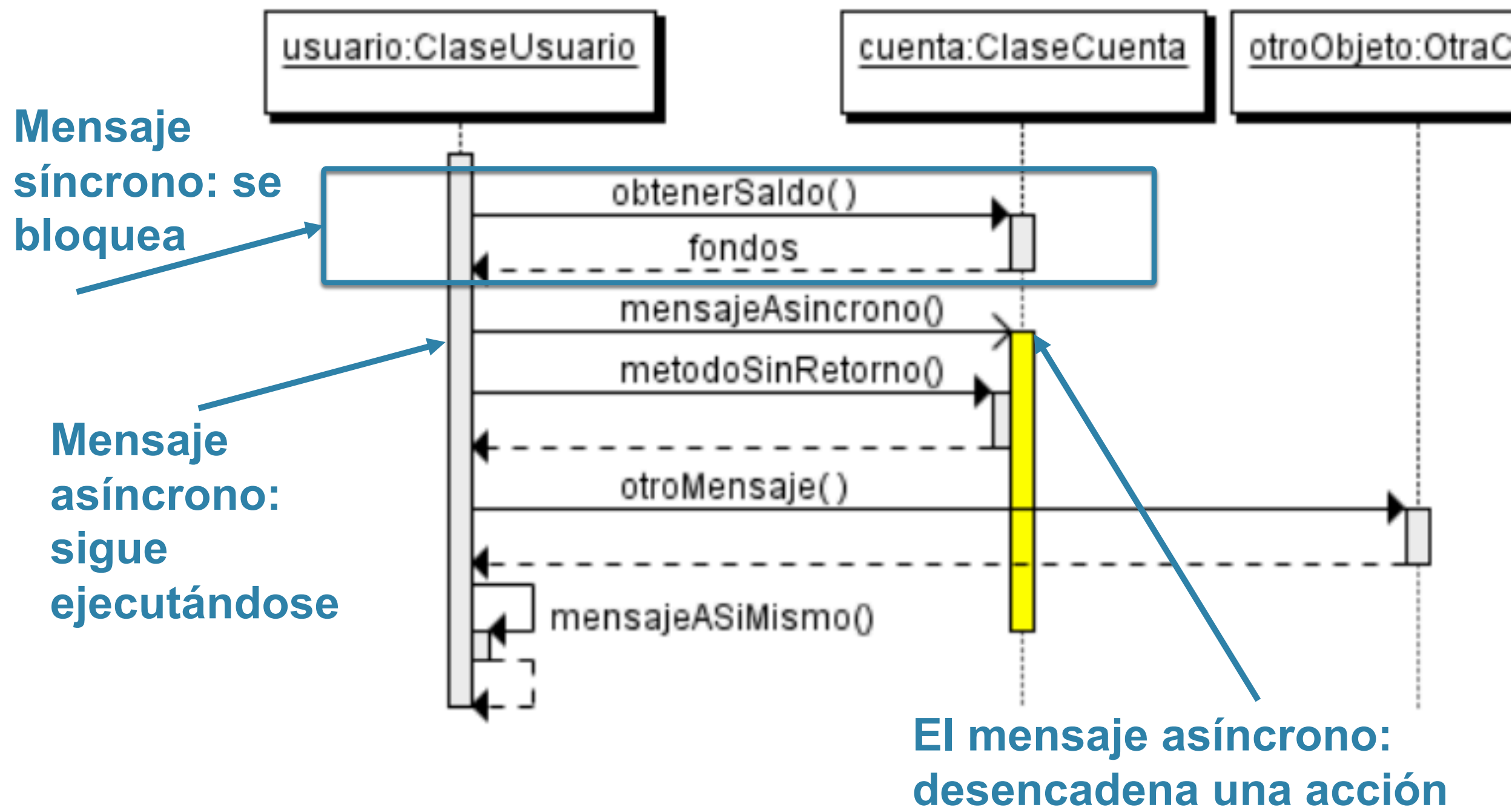
## DIAGRAMAS DE SECUENCIA: MENSAJES

- ✓ Los mensajes que se realizan entre los objetos pueden ser de dos tipos:
  - **Síncronos**: se corresponden con llamadas a métodos del objeto que recibe el mensaje. El objeto que envía el mensaje queda bloqueado hasta que termine la llamada (flechas con cabeza llena).
  - **Asíncronos**: estos mensajes terminan inmediatamente y crean un nuevo hilo de ejecución dentro de la secuencia (flechas con la cabeza abierta).
- ✓ Las **respuestas** a los mensajes se representan con líneas discontinuas.



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE SECUENCIA: MENSAJES



# DISEÑO ORIENTADO A OBJETOS

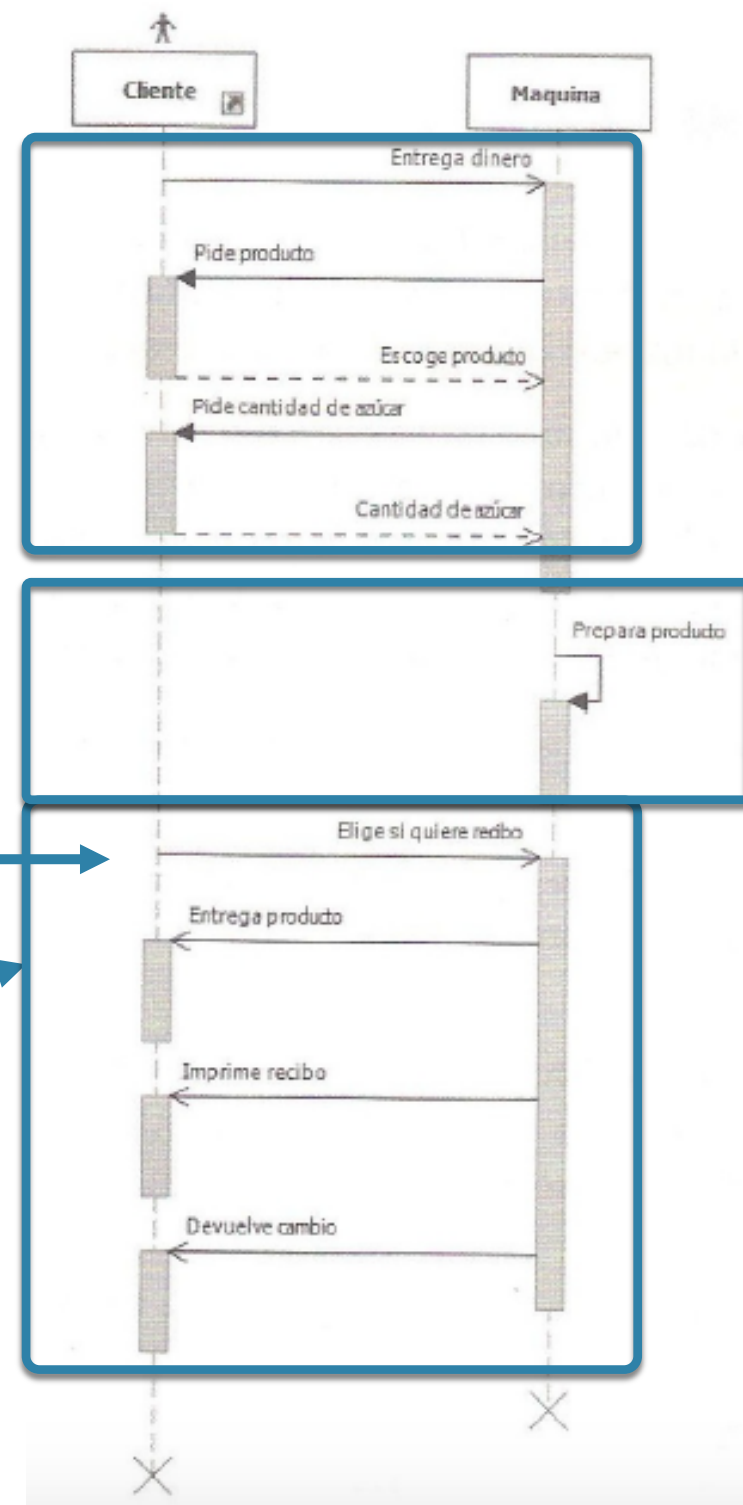
## DIAGRAMAS DE SECUENCIA: EJEMPLO MÁQUINA ESPENDEDORA

La máquina tiene tres periodos de proceso:

- (1) Pide los datos al cliente
- (2) Procesa la solicitud
- (3) Devuelve los resultados

No se representan las opciones para elegir imprimir recibo

Las opciones que no requieren respuesta son mensajes asíncronos



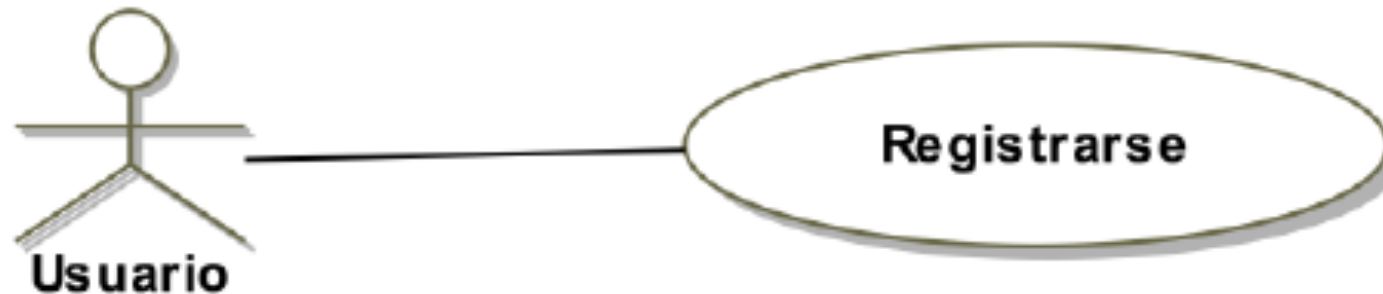
# DISEÑO ORIENTADO A OBJETOS

## CASOS DE USO

- ✓ Un **caso de uso** representa una interacción entre un agente externo (usuario, dispositivo externo o sistema externo), y un sistema informático.
- ✓ Se usan para representar los **requisitos funcionales del sistema**.
- ✓ Un **caso de uso** es un grafo con dos tipos de nodos y la relación entre ellos:
  - **Actor** - que representa cualquier elemento que intercambia información con el sistema, por lo que está fuera de él.
  - **Caso de uso** - Es una secuencia de intercambios en diálogo con el sistema que se encuentran relacionadas por su comportamiento
  - Las relaciones entre los actores y los casos de uso se denominan **arco de comunicación**.

# DISEÑO ORIENTADO A OBJETOS

## CASO DE USO

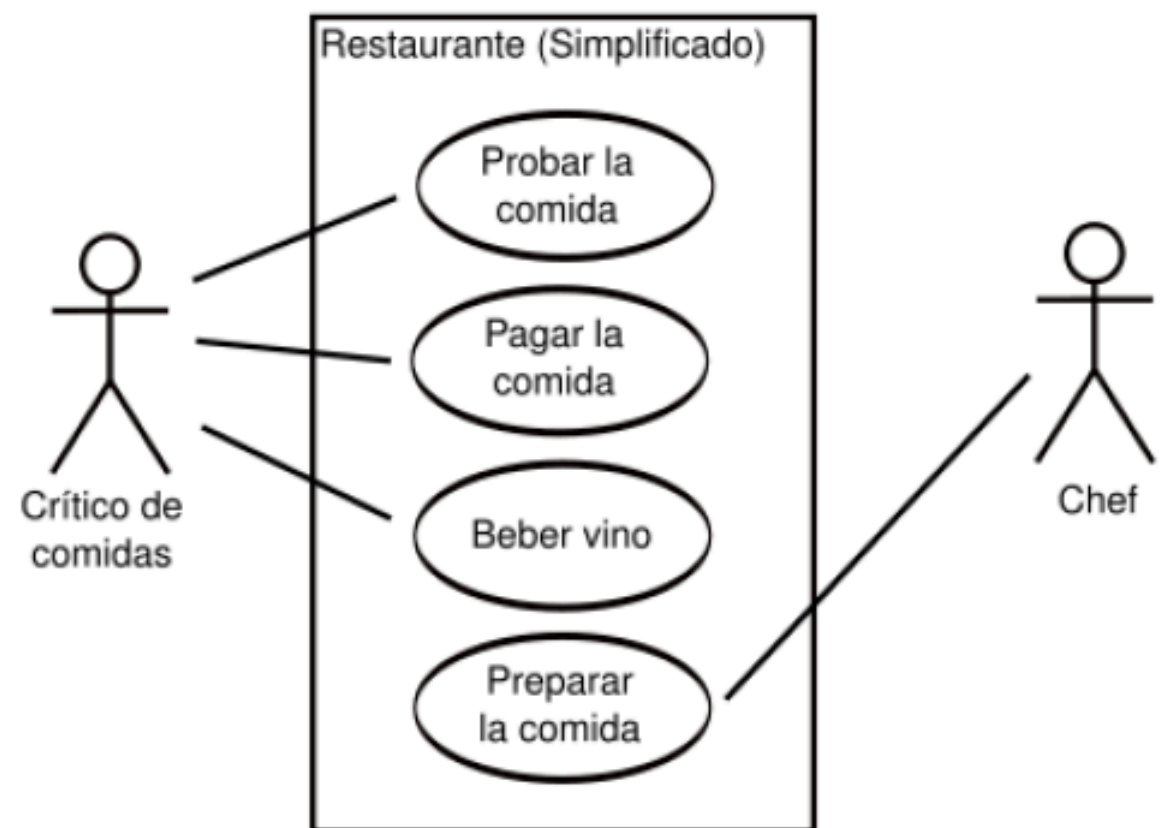


- ✓ *Registrarse* es un **caso de uso** que representa el conjunto de actividades que deben realizarse para llevar a cabo el proceso de registro.
- ✓ Cada **caso de uso** es una función que proporciona el sistema y que es de valor para los usuarios y otros agentes externos.
- ✓ Cada **caso de uso** es un **requisito funcional del sistema**.
- ✓ Cada **caso de uso** tiene una descripción informal en lenguaje natural o en un lenguaje estructurado, en un documento o plantilla destinado para ello.

# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO

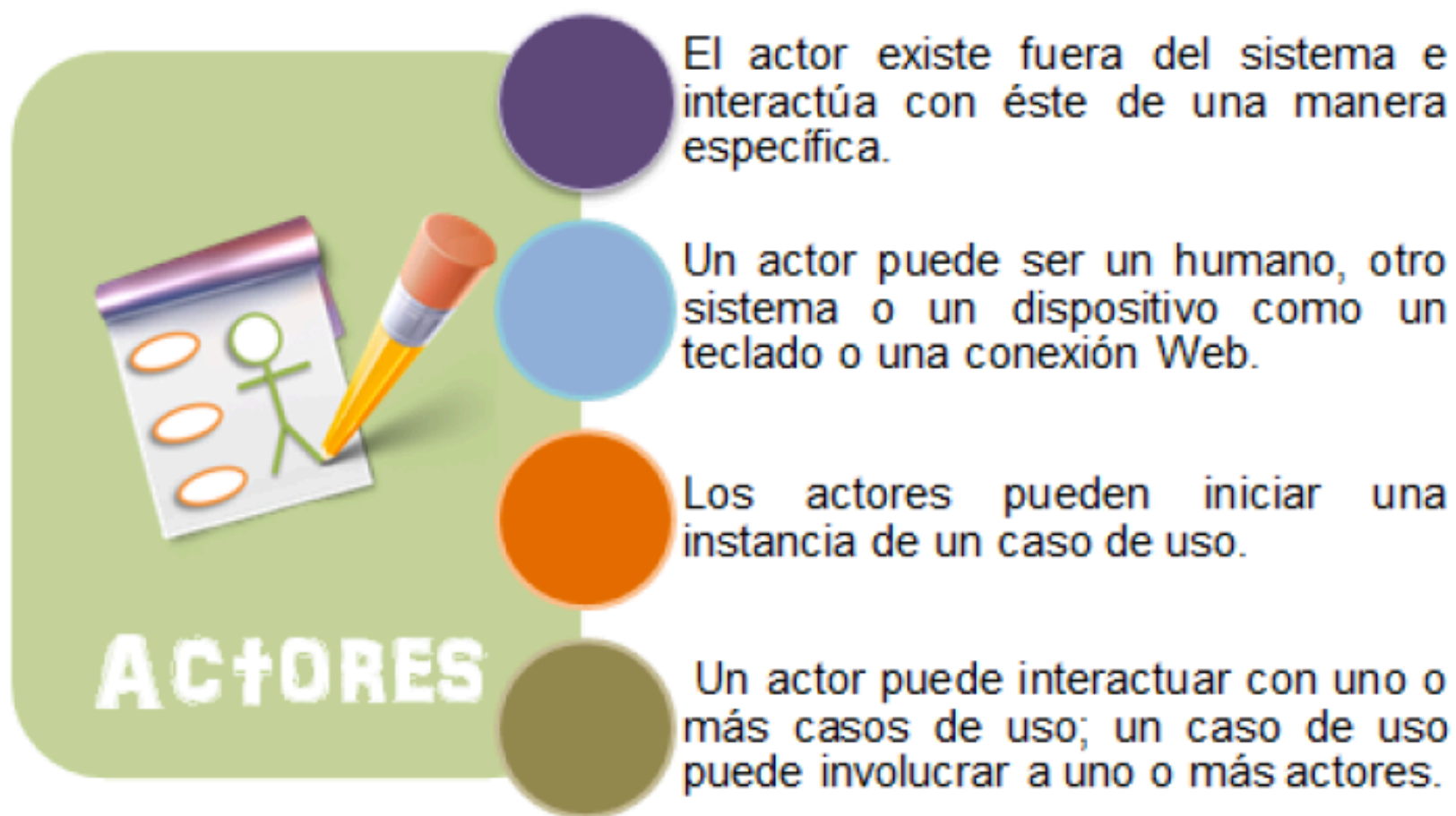
- ✓ Los **Diagramas de Casos de Uso** modelan la funcionalidad del sistema usando actores y los distintos casos de uso que componen toda la funcionalidad.
  - **Sistema/Escenario**: El rectángulo representa los límites del sistema que contiene los **casos de uso**. Los **actores** se ubican fuera de los límites del **Sistema**.
  - **Caso de uso**: se representan con óvalos. La etiqueta en el óvalo identifica la función del sistema, el requisito funcional del sistema.
  - **Actor**: entidades externas al **sistema** y que interactúan con él. El **actor** es un rol específico de un usuario. Puede ser un humano u otro sistema.



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: ACTORES

- ✓ Un **Actor** puede ser un empleado, pero también puede ser un cliente en la tienda de la empresa. Incluso cuando es la misma persona en el mundo real, se representa como dos símbolos distintos en un **diagrama de caso de uso**, ya que la persona interactúa con el sistema en distintos roles.
- **Actores principales:** son los que emplean directamente el sistema llevando a cabo las tareas más importantes.
- **Actores secundarios :** existen para que los principales puedan utilizar el sistema.

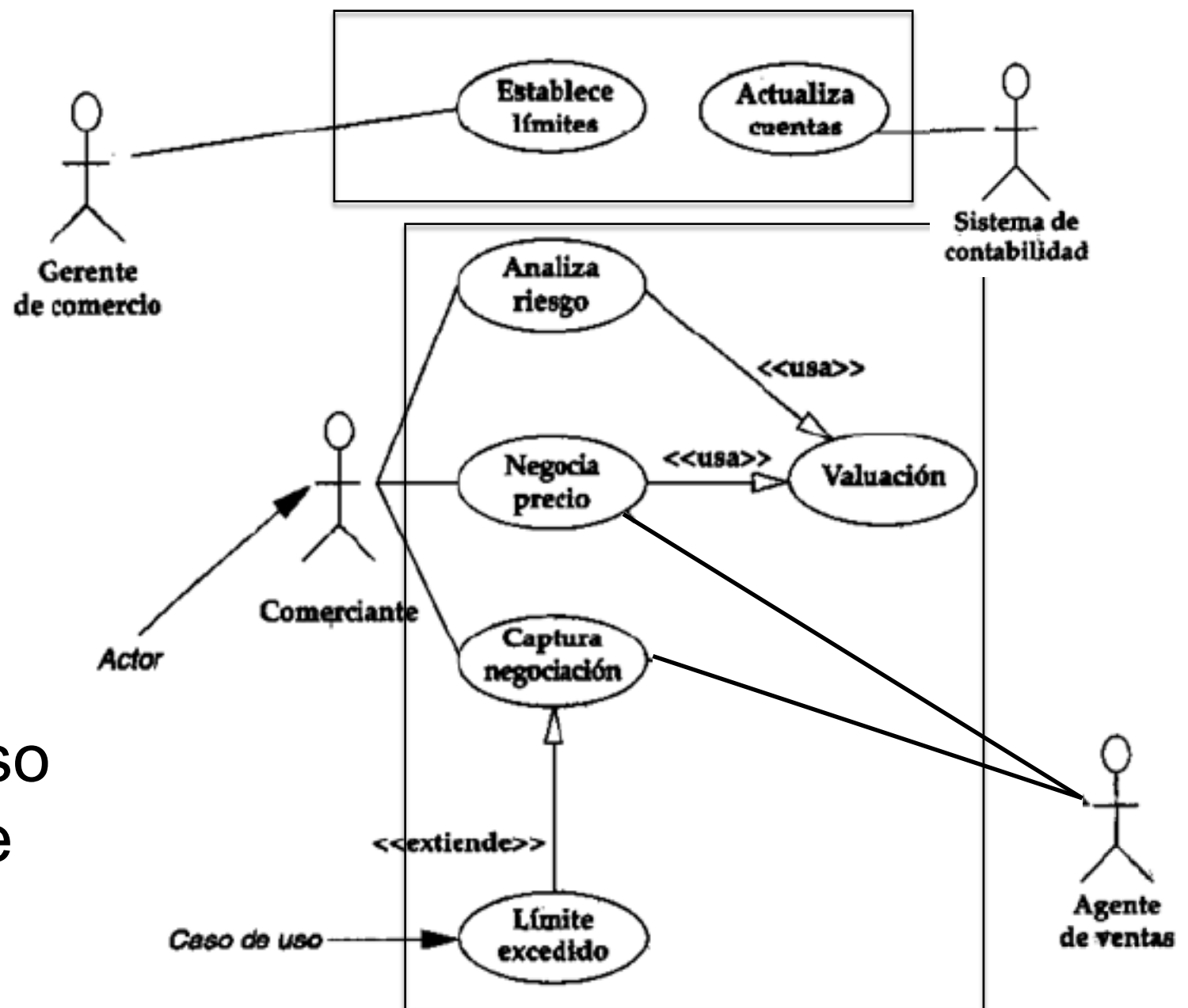




# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: RELACIONES

- ✓ Los **Diagramas de Casos de Uso** representan **relaciones** entre agentes externos (**actores**) y distintos **casos de uso** que componen toda la funcionalidad en el **sistema**.
- ✓ Podemos tener **relaciones** entre actores y casos de uso o entre los casos de uso de toda la funcionalidad en el sistema.

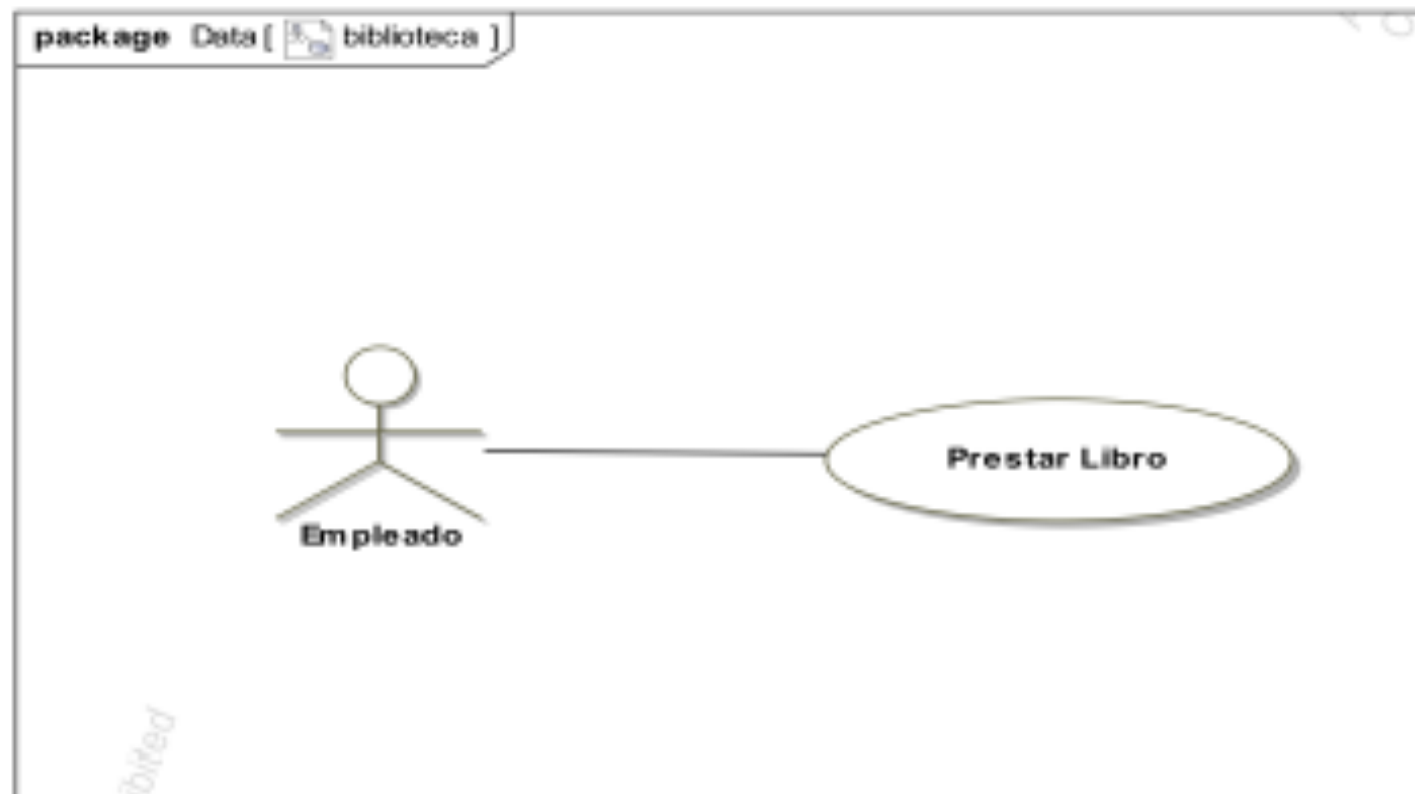




# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: RELACIONES

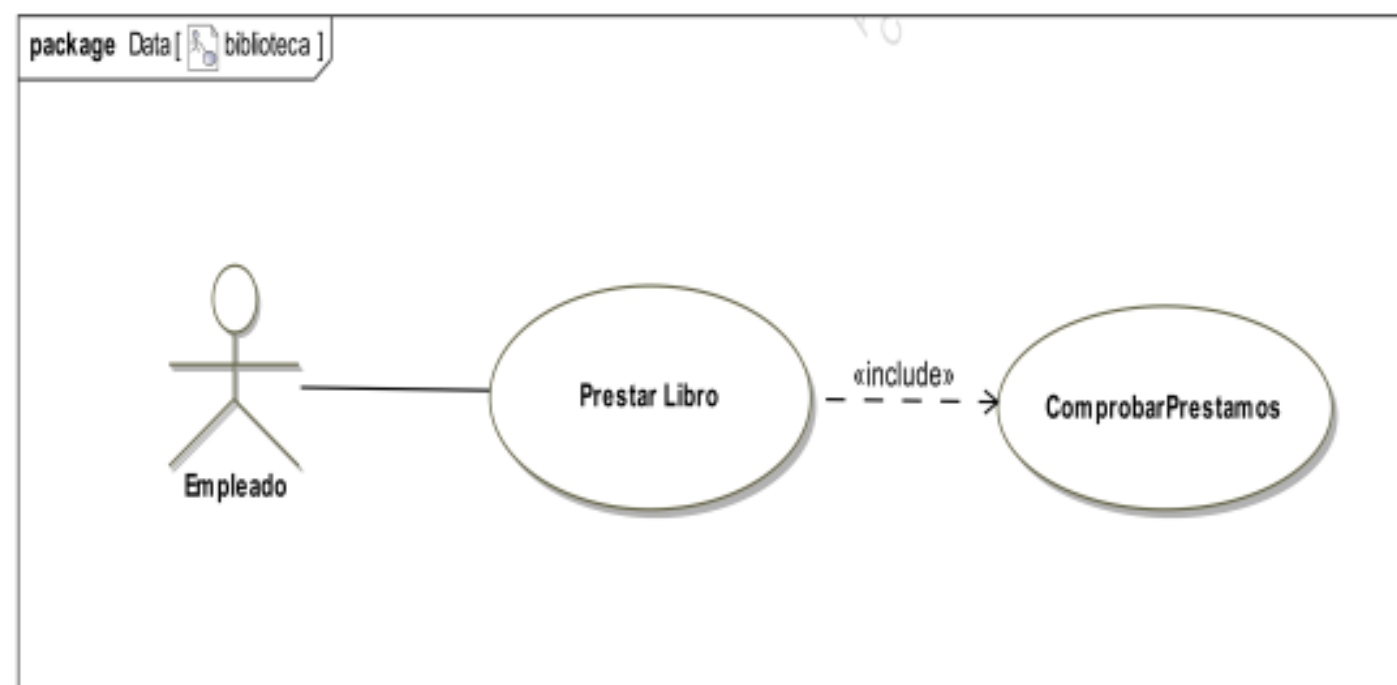
- ✓ **Relación de Asociación:** La **asociación** sólo es entre actores y casos de uso. Denota la participación de ese actor en ese caso de uso.



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: RELACIONES

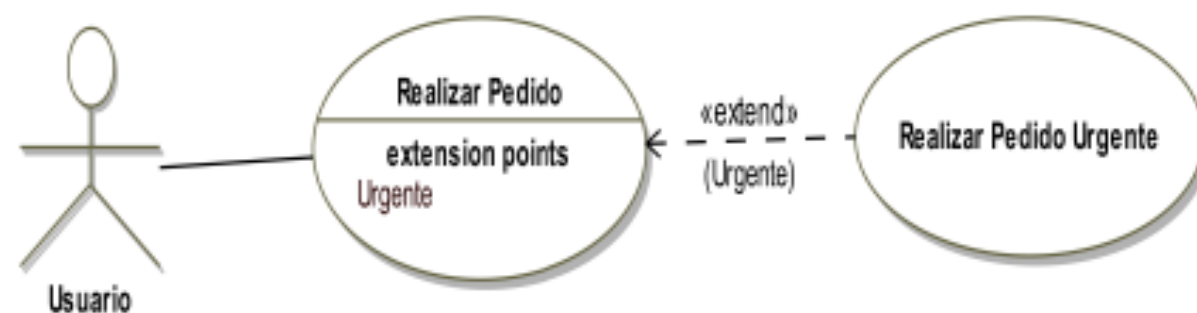
- ✓ **Relación de Inclusión:** Esta relación es entre dos casos de uso. Se usa para evitar describir el mismo flujo de eventos repetidas veces.
- ✓ En el siguiente ejemplo de la biblioteca, si un usuario tiene 3 libros prestados no puede coger otro libro. Por tanto cuando el empleado de la biblioteca vaya a registrar el préstamo, el sistema tendrá que comprobar los préstamos que tiene ese usuario.



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: RELACIONES

- ✓ **Relación de Extensión:** Esta relación también es entre dos casos de uso. Se utiliza cuando un caso de uso extiende el comportamiento de otro. Sirven para separar el comportamiento obligatorio del opcional, o para modelar ciertos subflujos de eventos que se ejecutan sólo bajo ciertas condiciones. Se usa para evitar describir el mismo flujo de eventos repetidas veces.
- ✓ Ejemplo de un tienda online. Los usuarios pueden realizar pedidos. Cuando están realizando el pedido hay una opción que es “urgente”. Si seleccionan esa opción el pedido se envía lo más rápido posible aunque el coste será mayor.



*Pedido Urgente extiende de Realizar Pedido.*

# DISEÑO ORIENTADO A OBJETOS


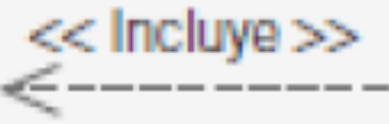
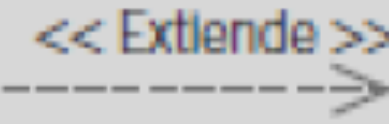

## DIAGRAMAS DE CASOS DE USO: RELACIONES

- ✓ **Relación de Generalización:** Esta relación es igual que la de las clases. Tenemos un caso de uso abstracto cuyo comportamiento lo proporcionarán sus hijos.
- ✓ Ejemplo de validación de usuarios. Hay sistemas en los que es necesario validarse para poder usarlos. Podemos validarnos con la típica forma de usuario y contraseña o con formas más futuristas como la comprobación de retina o de huellas dactilares.



# DISEÑO ORIENTADO A OBJETOS

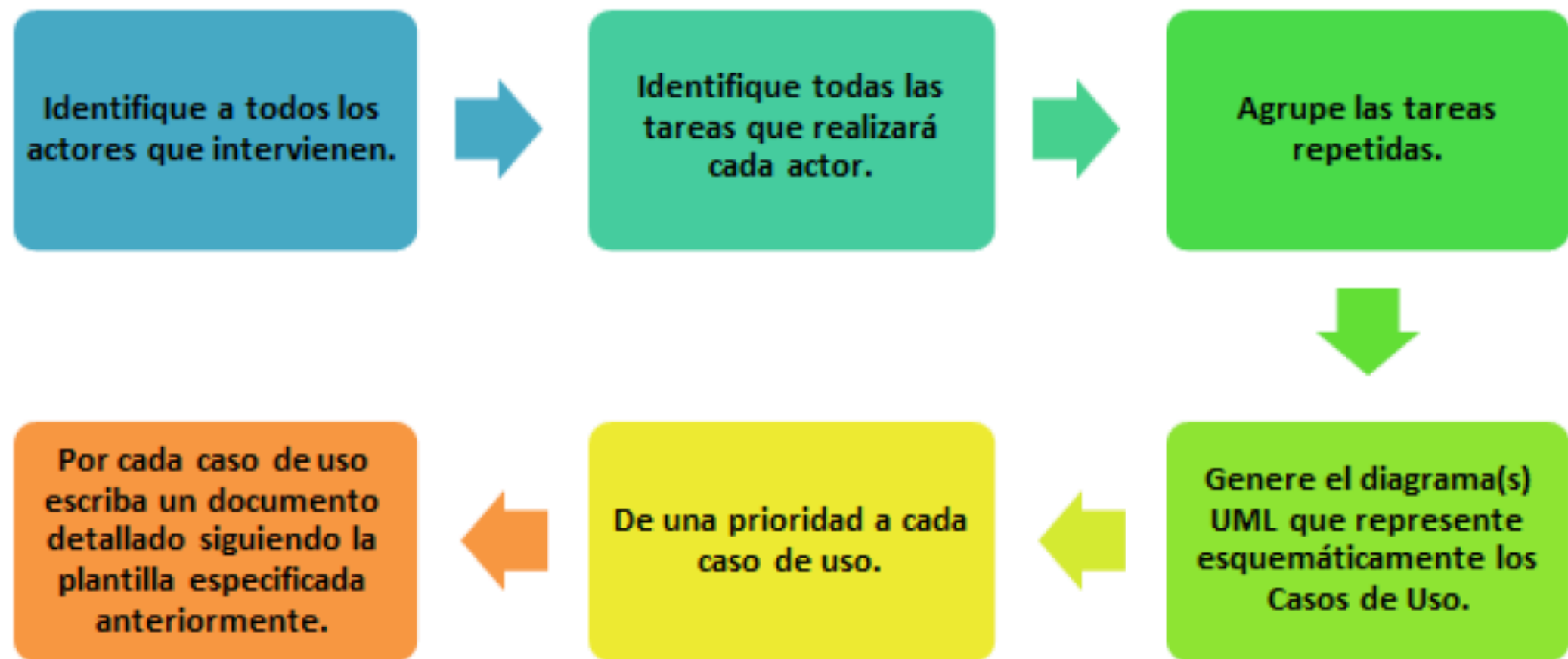
## DIAGRAMAS DE CASOS DE USO: RELACIONES

Relación	Símbolo	Significado
Comunica		Para conectar un actor con un caso de uso se utiliza una línea sin puntas de flecha.
Incluye		Un caso de uso contiene un comportamiento común para más de un caso de uso. La flecha apunta al caso de uso común.
Extiende		Un caso de uso distinto maneja las excepciones del caso de uso básico. La flecha apunta del caso de uso extendido al básico.
Generaliza		Una "cosa" de UML es más general que otra "cosa". La flecha apunta a la "cosa" general.

# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: DOCUMENTACIÓN

- ✓ Los **Diagramas de Casos de Uso** se documentan mediante:
  - **Diagrama en UML**
  - **Documento** detallado.
- ✓ Pasos identificables para escribir los casos de uso.



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: DOCUMENTACIÓN

- ✓ Se documenta, de manera detallada, cada caso de uso.
- ✓ Es una especificación completa del caso que incluye:
  - **Descripción** del caso de uso, centrándose en lo que debe hacerse, no en la manera de hacerlo. Deben evitarse expresiones imprecisas. Se busca sencillez y claridad.
  - **Precondiciones** para que se dé el caso de uso,
  - **Postcondiciones** o que es lo que pasará cuando se ejecute el caso de uso,
  - **El flujo de eventos o secuencias**. Puede utilizarse un lenguaje estructurado para representar la secuencia, el flujo de eventos, repeticiones y situaciones opcionales (pseudocódigo).
- ✓ Para especificar los casos de uso se suelen usar **plantillas**.



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: PLANTILLAS

Describir la **secuencia** y las **excepciones** como **escenarios** (similar a los caminos básicos de los grafos): situaciones concretas que deben recorrer total o parcialmente el caso de uso. Se deben considerar en lo posible todos los escenarios de modo que se pueda validar el caso de uso.

<Identificador>	<nombre descriptivo>	
Descripción<	El sistema deberá permitir a [lista actores] en [instante en el que se puede realizar el caso de uso] [funcionalidad que define el caso de uso]<según se describe en el siguiente caso de uso:<	
Actores	Actores que intervengan en el caso de uso. Tanto los principales (inician el caso de uso), como los secundarios (interactúan con el caso de uso durante la ejecución del mismo).	
Secuencia Normal	Paso <	Acción
	1<	{<acción a realizar>, realizar el caso de uso [caso de uso]}<
	2	<Situación que produce una alternativa><
	<	2a Si [Situación que produce una alternativa] el sistema deberá {<acción a realizar>, realizar el caso de uso [caso de uso]}
		2b Si [Situación que produce una alternativa] el sistema deberá {<acción a realizar>, realizar el caso de uso [caso de uso]}
	...	....
	...<	Include (<OtroCasoDeUso>)
	n	Paso n que tiene un punto de extensión
Excepciones<	Punto de extensión: <nombrePuntoExtensión>	
	Paso <	Acción
	p	En el caso de que [situación que provoca la excepción] el sistema deberá {<acción a realizar>, realizar el caso de uso [caso de uso]}
	...	...
Precondiciones	<q	...
	Condiciones que pudiera haber previas	
Postcondiciones	Situación del sistema después del caso de uso	

# DISEÑO ORIENTADO A OBJETOS

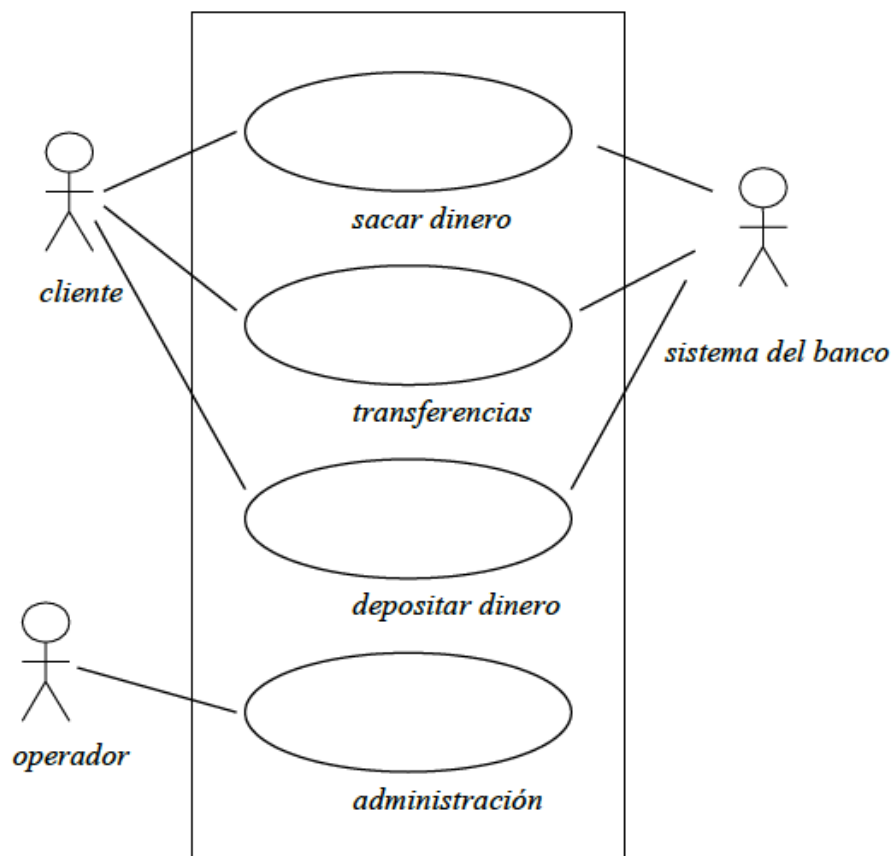
## DIAGRAMAS DE CASOS DE USO: EJEMPLO

Código y Nombre →	Id + Nombre	CU-05 Generar Factura Cliente
Si se han ordenado los casos de uso por prioridad →	Autores	Víctor Gómez
	Prioridad	Alta
	Criticidad	Alta
	Fuente	Abogado del Bufete de Abogados S.C.P.
	Responsable	Víctor Gómez
Actores que participan →	Actor Principal	<u>System</u>
	Actores Secundarios	El usuario y todos los integrantes de <u>Kalendar</u>
Descripción sencilla y clara. Centrarse en lo que hay que hacer →	Descripción	Cada vez que un usuario realice un compra (en nuestro caso una compra es un pago para promocionar una actividad) se le enviará la factura de su compra.
De dónde viene →	Objetivo	Generar Factura al Cliente
	Evento de Activación	Una vez que el usuario realiza una compra
	Precondición	El usuario deberá haber efectuado el pago
	Garantías Si Éxito	El usuario recibirá en su área de notificaciones y en su correo la factura con los datos de su compra.
Utilizar un lenguaje estructurado o pseudocódigo →	Garantías Mínimas	El usuario recibirá la factura en su área de notificaciones
	Escenario Principal	<ol style="list-style-type: none"><li>1. El sistema recoge la información de la compra</li><li>2. El sistema elabora un informe con esa información</li><li>3. El sistema envía un mensaje al usuario adjuntando la factura</li></ol>
	Extensiones	<ol style="list-style-type: none"><li>1. El sistema no puede obtener los datos de la compra.<ol style="list-style-type: none"><li>1a. El caso de uso termina.</li></ol></li><li>2. El sistema no puede generar la factura.<ol style="list-style-type: none"><li>2a. El sistema envía los datos como un mensaje plano.</li><li>2b. El caso de uso continúa con normalidad.</li></ol></li><li>3. El sistema no puede enviar el mensaje.<ol style="list-style-type: none"><li>3a. El caso de uso termina</li><li>3b. El caso de uso se repetirá al día siguiente.</li></ol></li></ol>

# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: EJEMPLO

### Casos de uso - Ejemplos

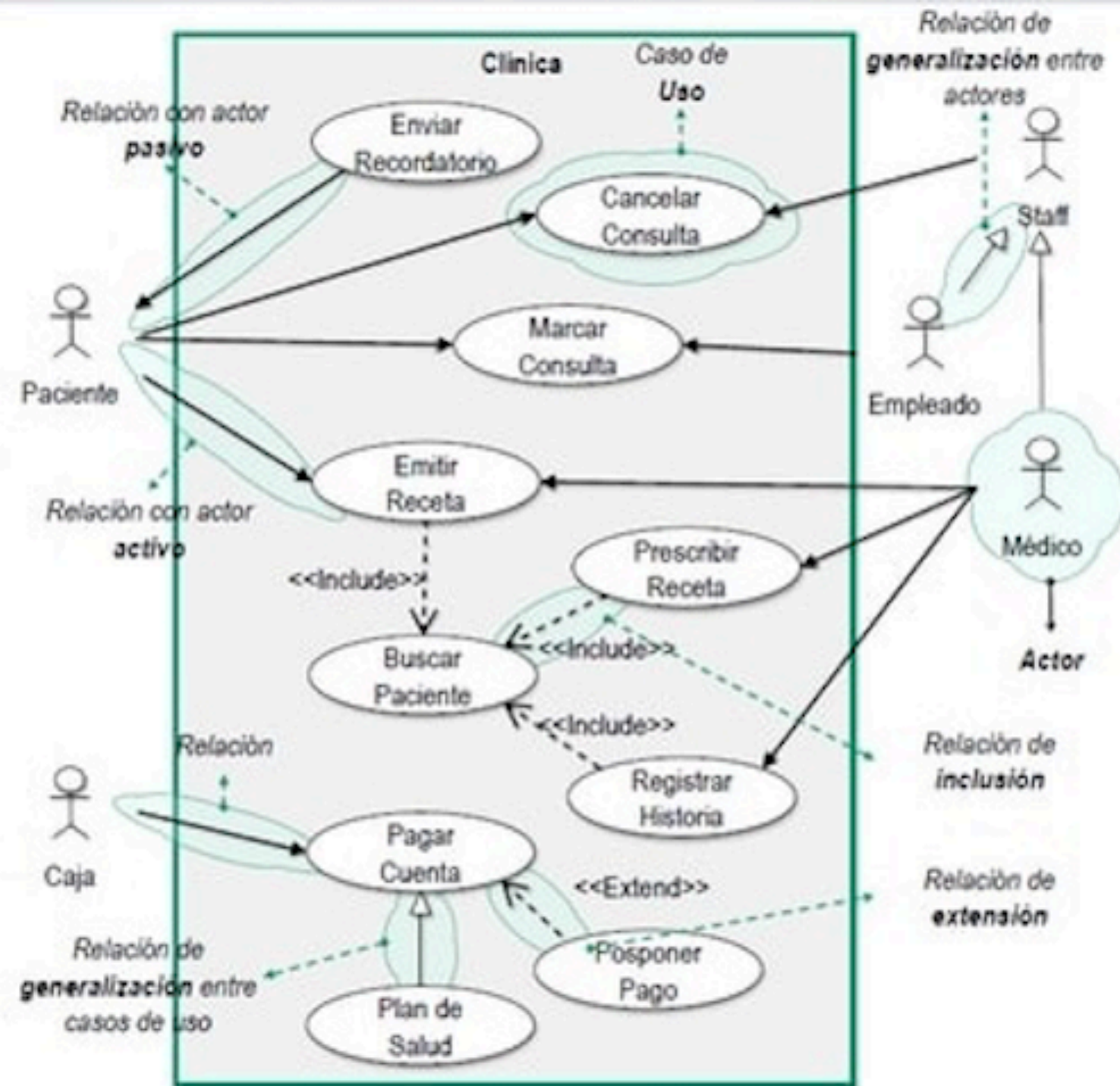


<b>CU-003</b>	<b>Sacar dinero</b>
<b>Descripción</b>	El sistema deberá permitir al cliente del banco, en cualquier momento, sacar dinero según se describe en el siguiente caso de uso:
<b>Secuencia Normal</b>	<div>1+ El usuario inserta la tarjeta en el cajero</div> <div>2 + El cajero lee el código de la banda magnética de la tarjeta y verifica si es aceptable y pide el código del usuario</div> <div>3+ El usuario introduce el código</div> <div>4 + Si el código es correcto, el cajero pide al usuario que seleccione el tipo de transacción deseada</div> <div>5+ El usuario selecciona la función sacar dinero,</div> <div>6 + El cajero le pide al usuario que teclee la cantidad deseada</div> <div>7 + El usuario teclea la cantidad que quiere sacar,</div> <div>8 + El cajero envía la petición al sistema del banco</div> <div>9 a Si conecta el sistema deberá comprobar si hay dinero en la cuenta</div> <div>9 b Si no conecta el sistema deberá comprobar si el dinero es menos que el límite</div> <div>10 En cualquiera de los dos casos el sistema: + expulsa la tarjeta + imprime el recibo + entrega el dinero</div>
<b>Excepciones</b>	<div>2' La tarjeta no es aceptada + Se expulsa emitiendo un sonido</div> <div>4' Código incorrecto (1,2) + Se emite un mensaje dando al usuario la oportunidad de volver a introducir el código (paso 3)</div> <div>4" Código incorrecto (3) + Se emite un mensaje y se retiene la tarjeta</div> <div>9' No autorizado para sacar dinero + El sistema de banco no autoriza a sacar dinero. Se emite un mensaje de información y se expulsa la tarjeta</div> <div>9 a ', 9 b' No hay dinero suficiente + El cajero no dispone de la cantidad pedida. Emite un mensaje y vuelve al paso 7</div> <div>1..10' Cancelar + En cualquier momento el usuario puede cancelar la transacción, con lo que se expulsa la tarjeta</div>



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: EJEMPLO



# DISEÑO ORIENTADO A OBJETOS

## DIAGRAMAS DE CASOS DE USO: EJEMPLO

