

## 2.- Diseño del comportamiento: Diagrama de actividades



M<sup>a</sup> Antonia Zapata



Los diagramas de actividades sirven para

representar el **comportamiento dinámico** de un sistema

haciendo hincapié

en la **secuencia de actividades** que se llevan a cabo y las **condiciones** que guardan o disparan esas actividades

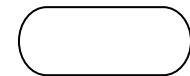


# Elementos básicos



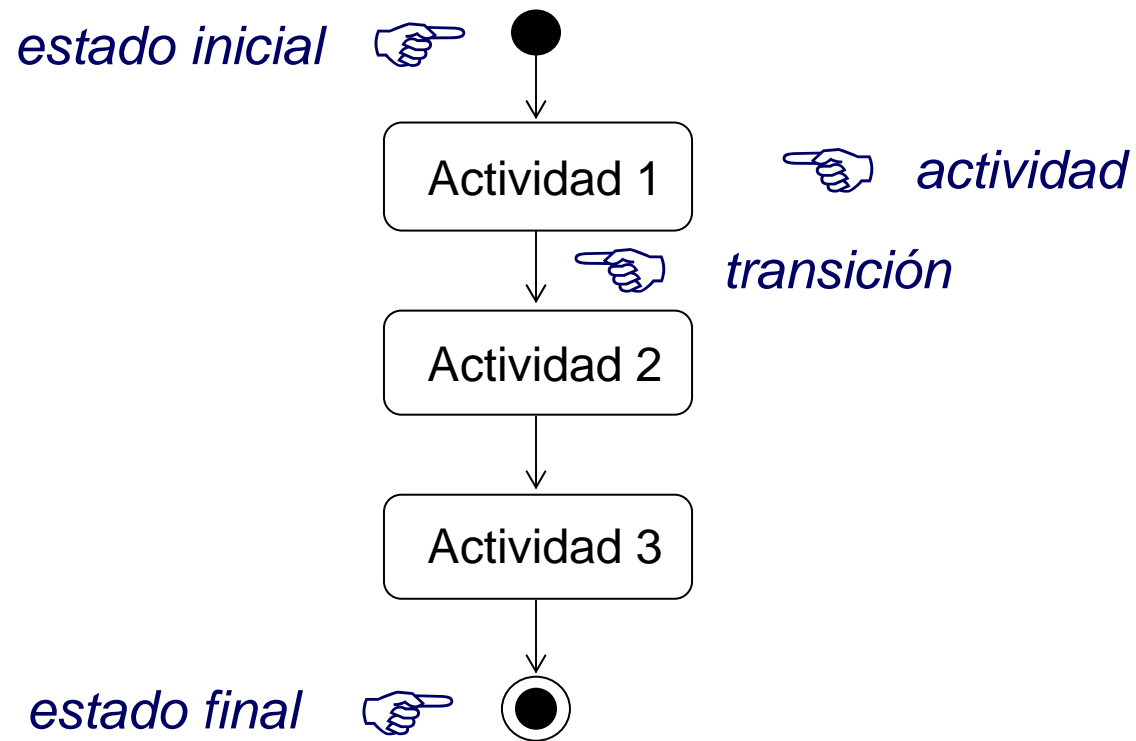
<b>Estado inicial:</b> Marca el punto de inicio del flujo de ejecución	
<b>Estado final:</b> Marca el punto final del flujo de ejecución	
<b>Actividad/Acción:</b> Representan la realización de un paso del flujo de ejecución	
<b>Flujo de control:</b> Determina qué actividad va a continuación de otra (se le puede asociar un nombre)	

En los libros aparecen ejemplos con la notación de la versión 1.5



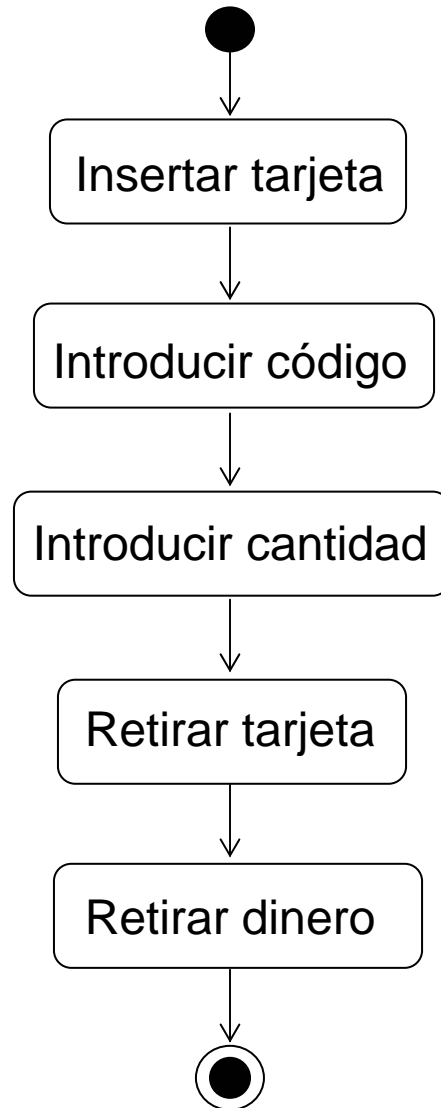


# Representación gráfica





# Ejemplo: cajero automático (versión 1)





# Restricciones



- Un estado inicial no puede ser destino de una transición
- Toda actividad tiene al menos un flujo de entrada y otro de salida
- Puede haber **cero o más estados finales** (por ejemplo, un proceso continuo no tendrá estado final)



# Recomendaciones



- Conviene colocar (no es obligado) el **estado inicial** en la parte **superior izquierda** del diagrama



Lo normal es que puedan aparecer **distintos flujos** dependiendo de que se den una serie de **circunstancias** u otras.

**Por ejemplo:**

¿qué pasa si la tarjeta no es válida?

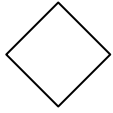
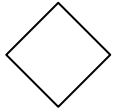
¿qué pasa si el código introducido no es el de la tarjeta?

¿qué pasa si la cantidad solicitada supera la cantidad disponible en la tarjeta?

¿qué pasa si con los tipos de billetes que tiene el cajero en ese momento no puede entregar la cantidad solicitada?

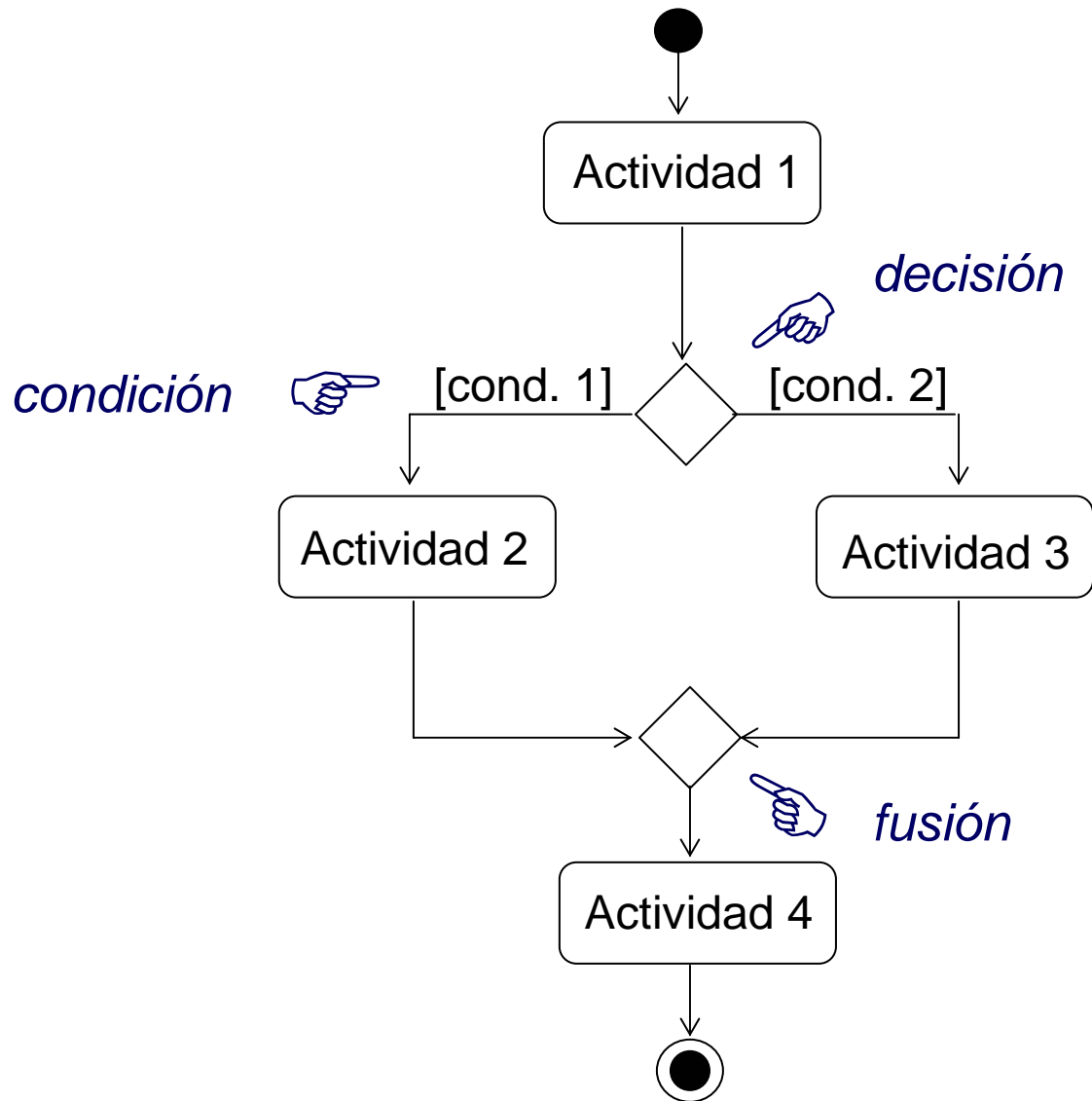




<p><b>Decisión:</b></p> <p>Marca la existencia de <b>flujos alternativos</b></p>	
<p><b>Condición/guarda:</b></p> <p>Se escribe encima de un flujo de control e indica la <b>condición que se debe cumplir</b> para que el flujo continúe a través de él</p>	[cond.]
<p><b>Fusión (Merge):</b></p> <p>Sirve para <b>juntar dos o más flujos</b> alternativos de ejecución que se han producido por una decisión</p>	



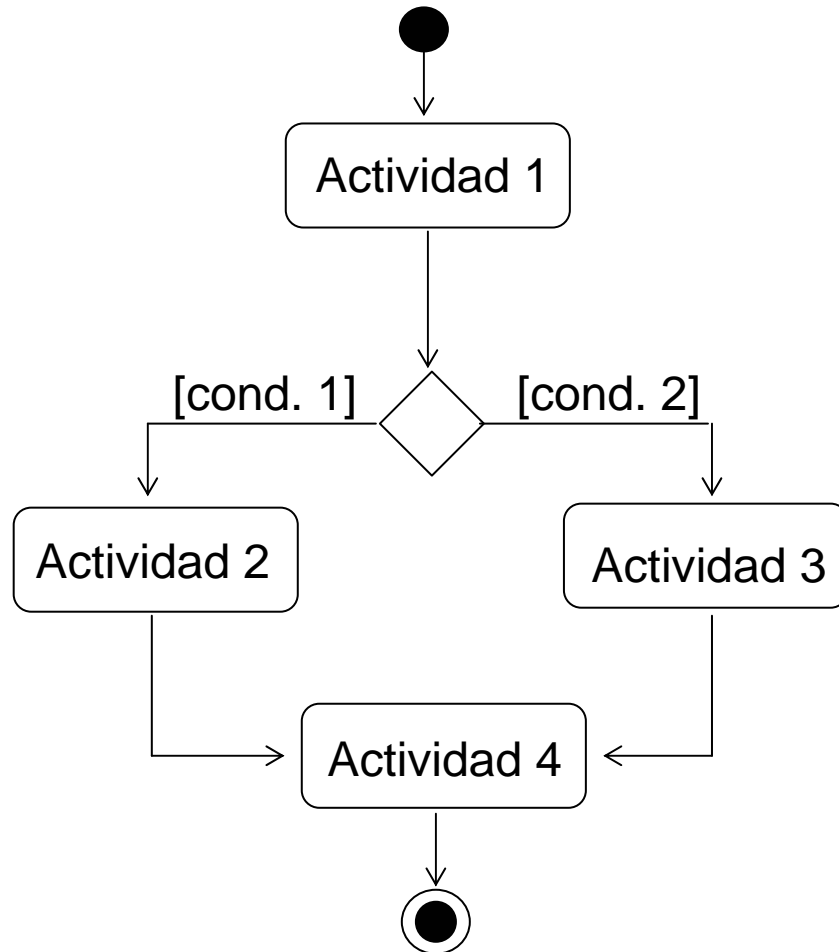
# Representación gráfica



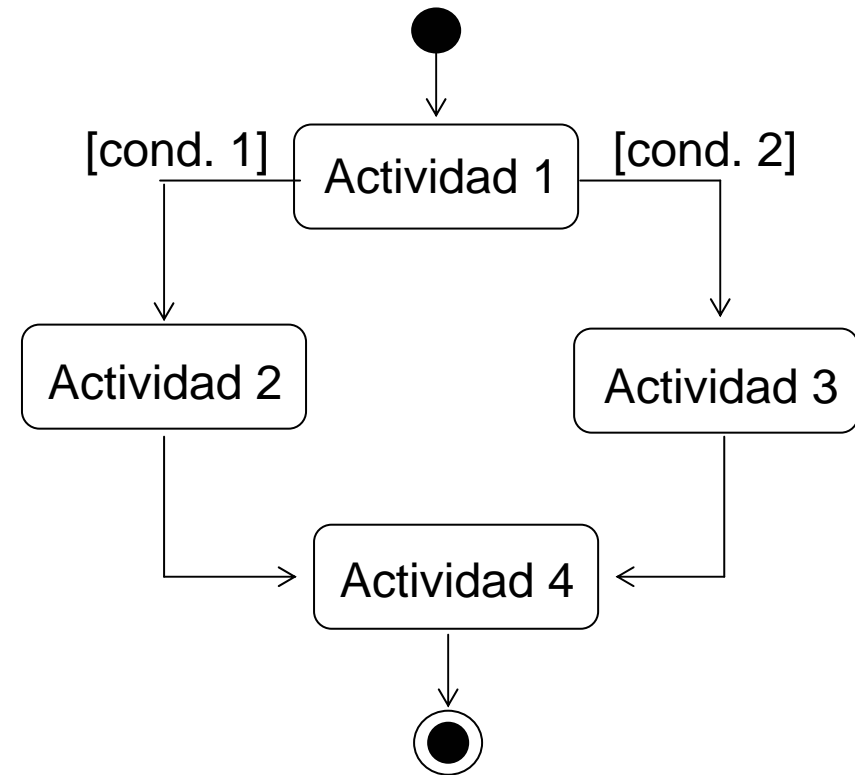
*Flujos alternativos:  
se lleva a cabo la  
actividad 2 o la 3.*



# Otras posible representaciones



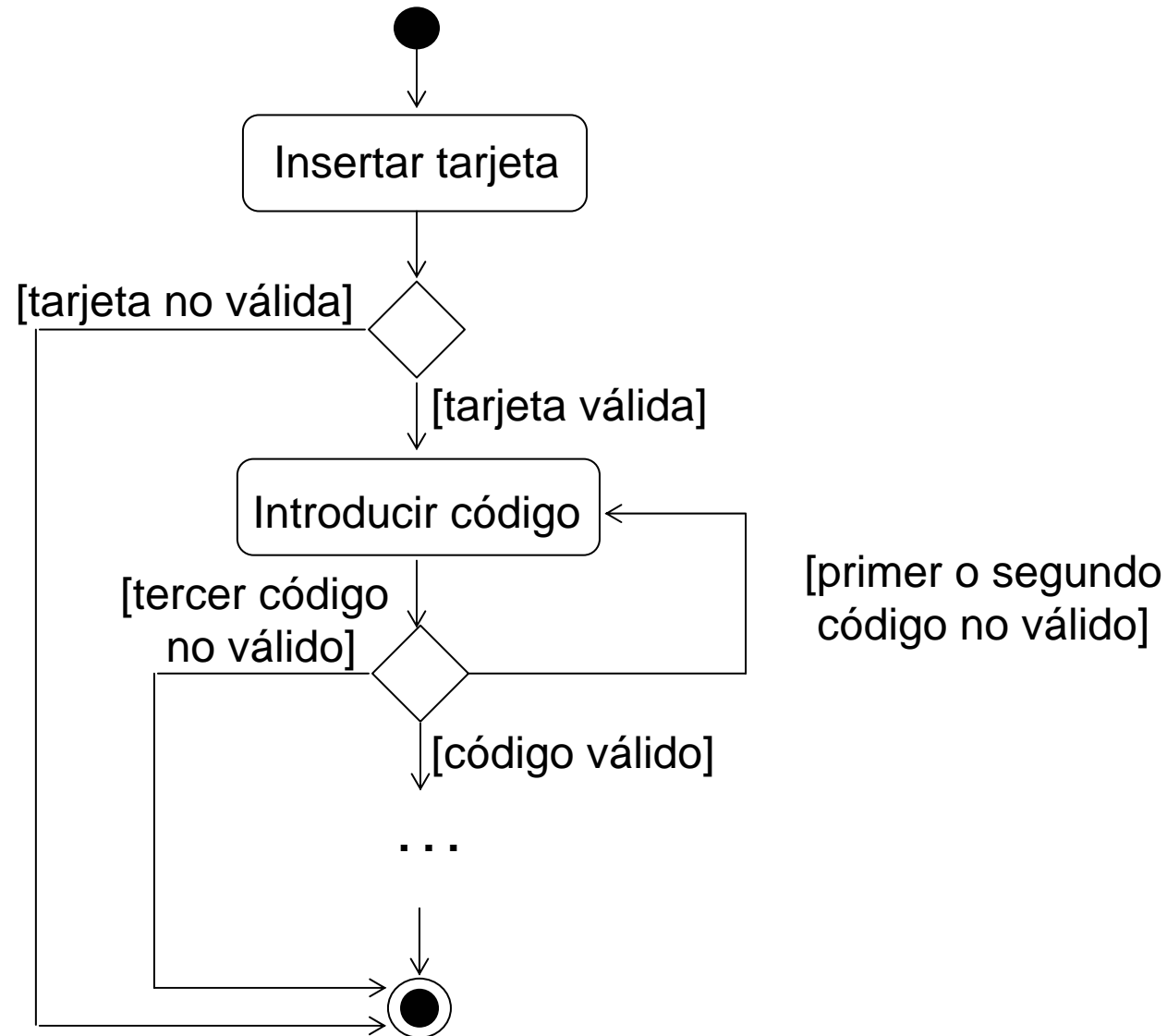
Sin representar la fusión  
(la más usual)



Sin representar la decisión  
ni la fusión



# Ejemplos: cajero automático (versión 2)





- Una decisión tiene un flujo de entrada y dos o más de salida
- Todo flujo de salida de una decisión debe estar etiquetado con una condición
- Las condiciones de todos los flujos de salida de una decisión deben ser disjuntas y completas
- Se puede utilizar la condición **else** para representar el flujo que se sigue en caso de que ninguna de las otras condiciones sea cierta
- Una fusión tiene dos o más flujos de entrada y un flujo de salida



Un diagrama de actividades también nos permite representar flujos que ocurren de forma concurrente (en paralelo).

También permite indicar actividades que se pueden hacer en cualquier orden (si lo hicieran elementos distintos lo podrían hacer a la vez)

Por ejemplo:

A la vez que se expulsa una tarjeta no válida se le muestra un mensaje al usuario

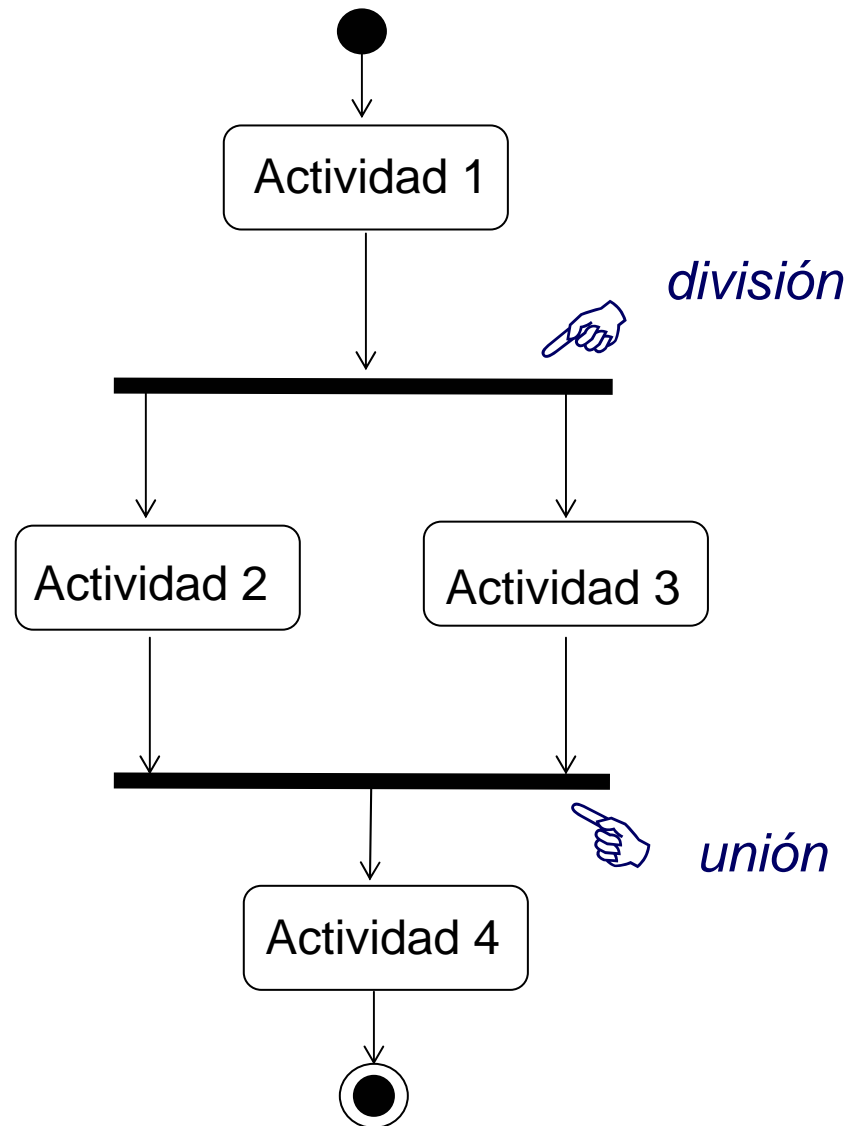
Supongamos que el código y la cantidad se pueden introducir en cualquier orden.



# Flujos concurrentes



<b>División:</b> Marca el inicio de flujos de actividades en paralelo	
<b>Unión:</b> Marca el fin de flujos de actividades en paralelo	

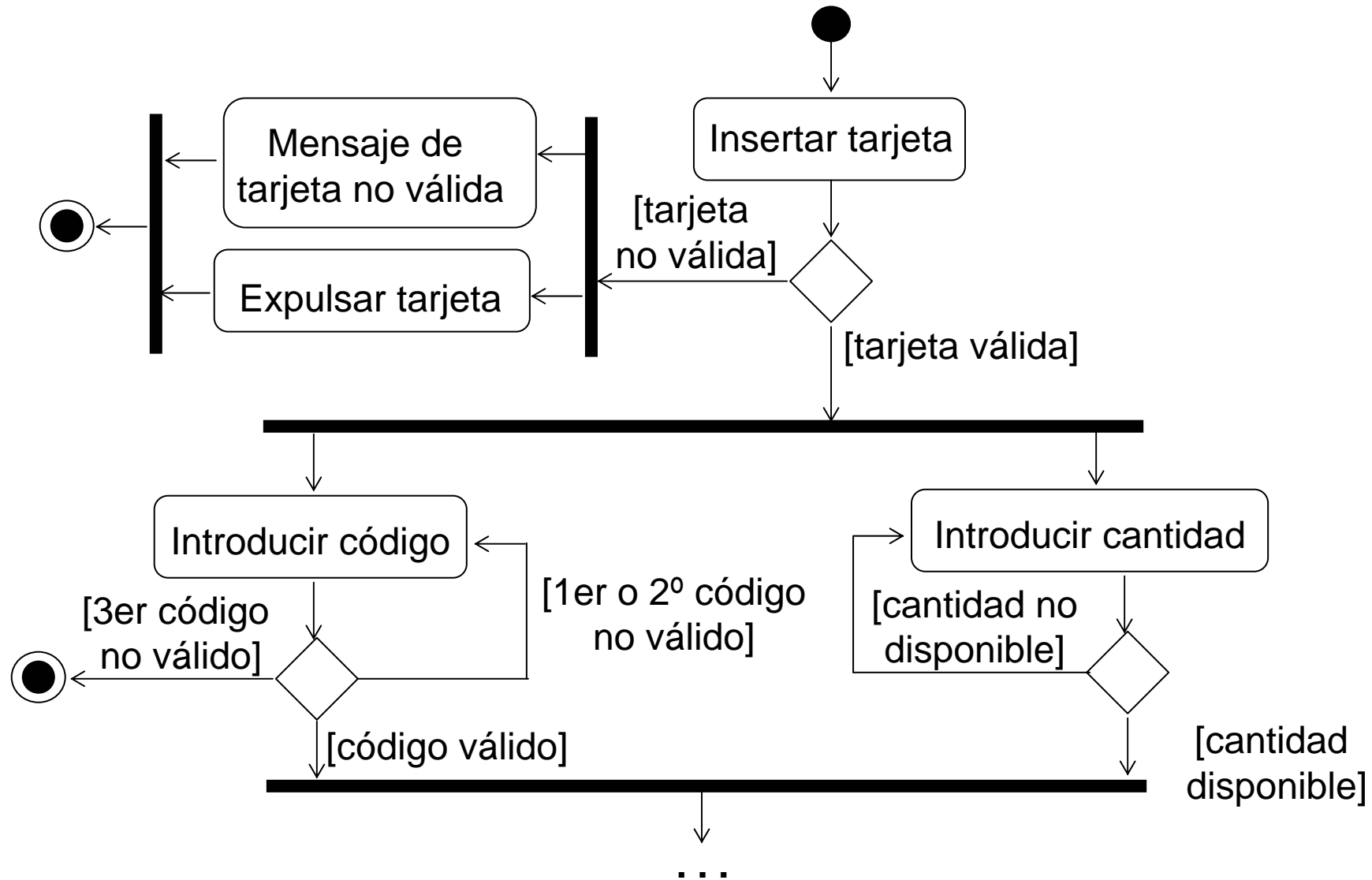


*Flujos concurrentes: se lleva a cabo la actividad 2 y la 3.*





# Ejemplos: cajero automático (versión 3)





- Una división tiene un flujo de entrada y dos o más flujos de salida
- Una unión tiene dos o más flujos de entrada y un flujo de salida
- El flujo de salida de una unión se dispara cuando se han finalizado todos los flujos de entrada en la unión (todos ellos discurren en paralelo)



Para que los diagramas no queden excesivamente complejos se pueden modularizar haciendo uso de **subactividades**.

Por ejemplo:

Los procesos de:

- validación del código
- introducción de una cantidad
- tarjeta no válida

se puede representar aparte facilitando la comprensión del diagrama

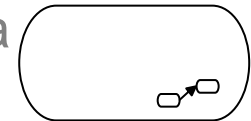


## Subactividad:

La actividad se describe más en detalle en un diagrama de actividades aparte



Nota: en la versión 1.5 en los diagramas de actividades se utilizaba otro símbolo



## Acción/Actividad

Una **acción** representa un paso del flujo de ejecución que se considera atómico, mientras que una **actividad** representa un comportamiento compuesto de elementos individuales que son acciones.

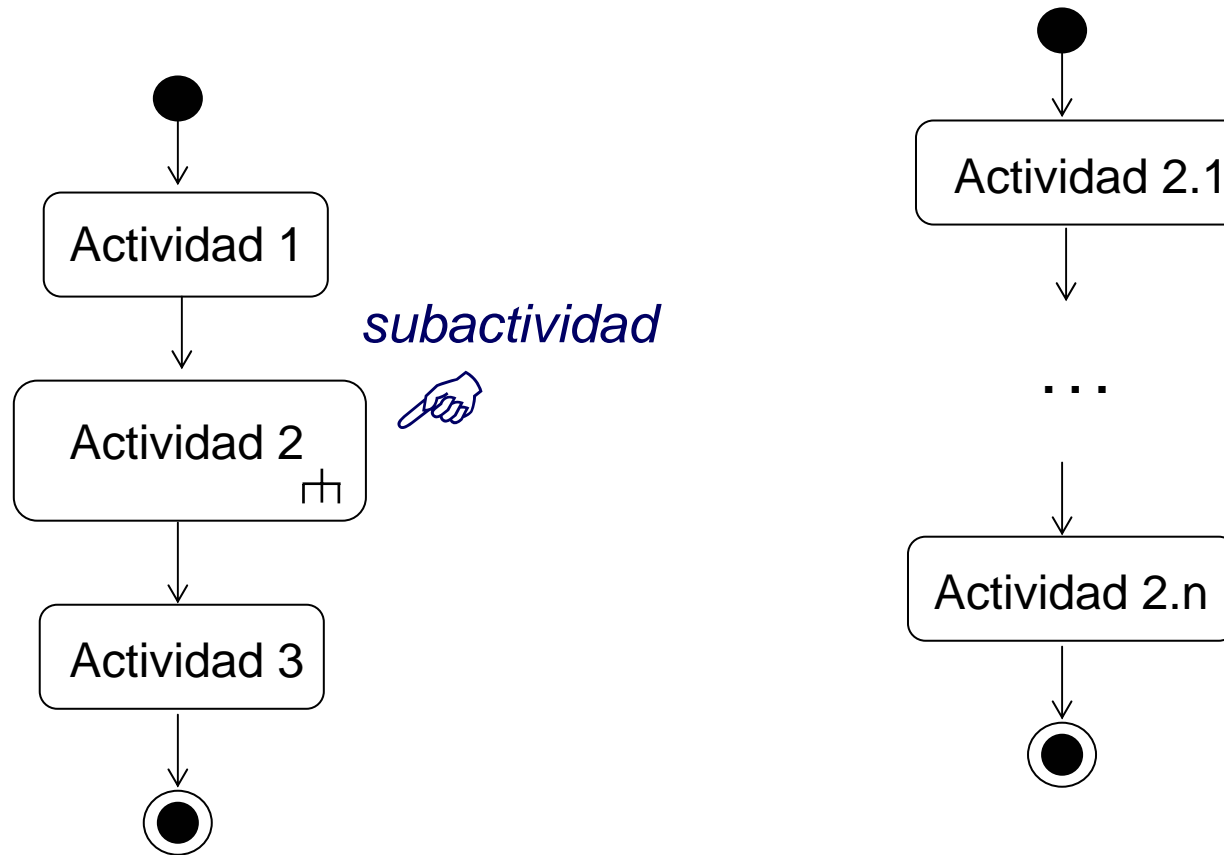
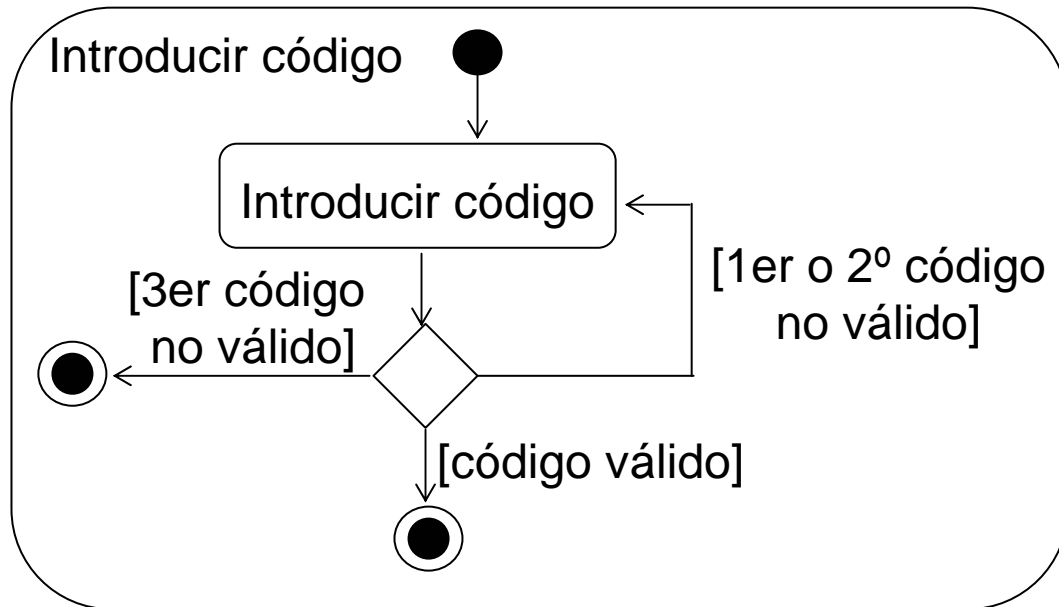
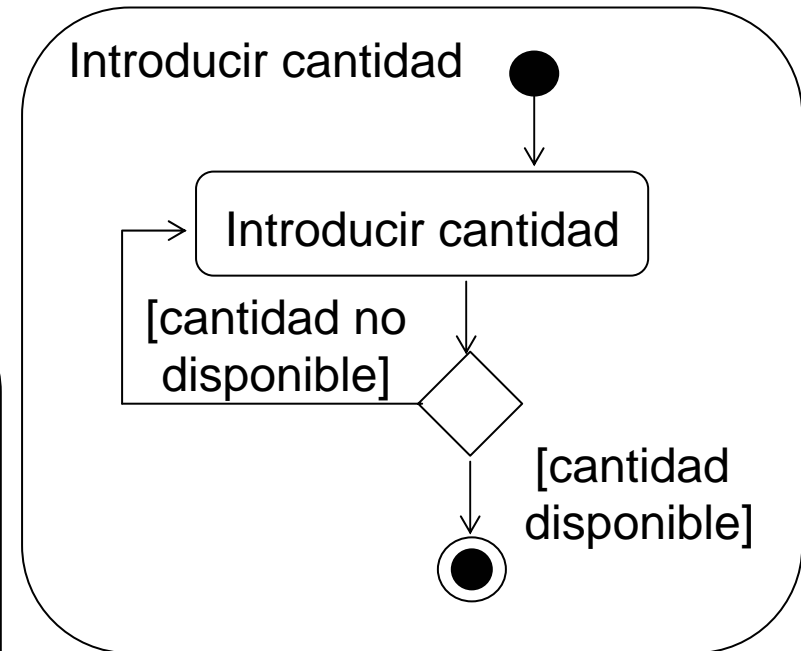
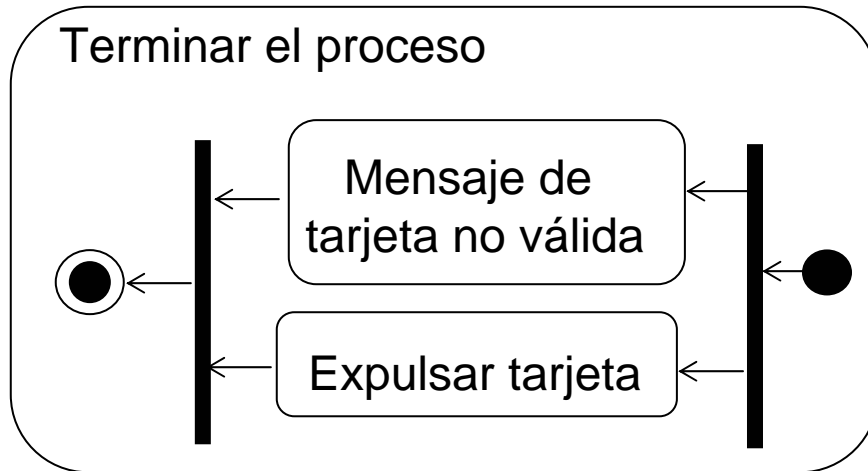


Diagrama de actividades describiendo la Actividad 2

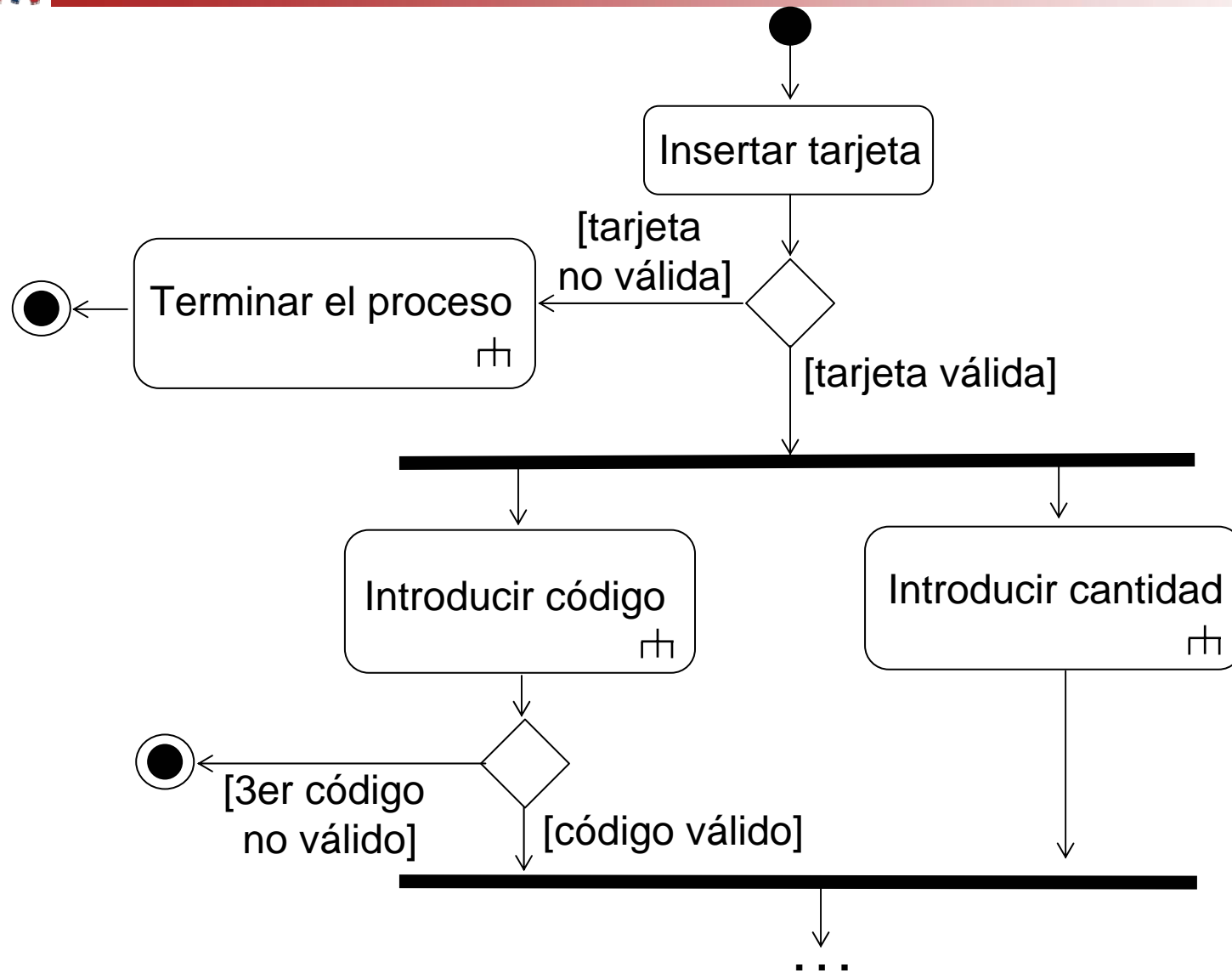


# Ejemplos: cajero automático (versión 4)





# Ejemplos: cajero automático (versión 4)





- Un diagrama de actividades **demasiado grande** nos debe hacer pensar que igual conviene **incluir alguna subactividad** para simplificarlo





Ya conocemos:

## Estado final:

Marca el punto final de **todos los flujos** de ejecución

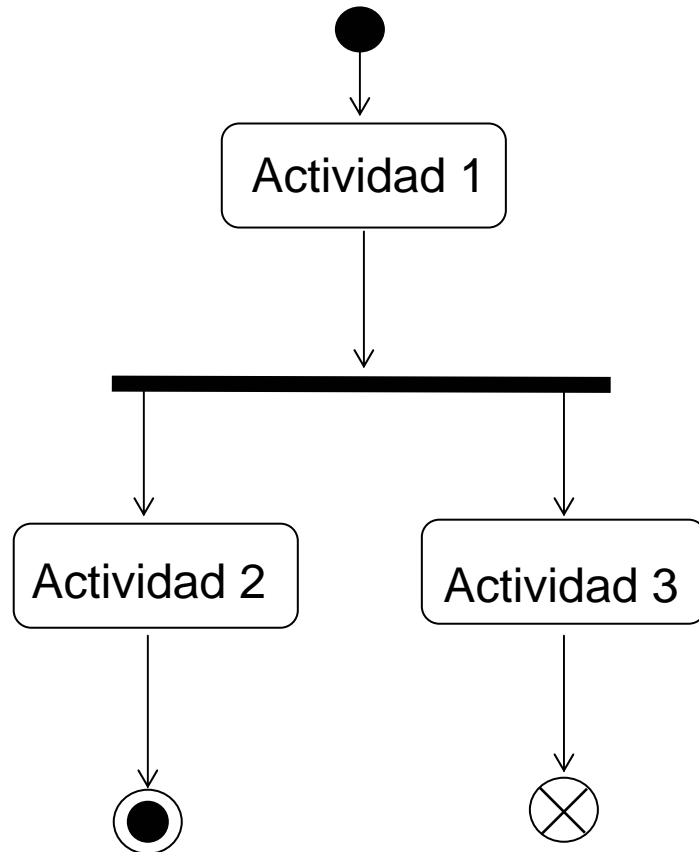


UML 2.0 incorpora la noción de:

## Final de flujo:

Marca el punto final de un flujo, dejando en ejecución el resto de flujos





- Si la Actividad 2 es la primera que acaba entonces la Actividad 3 se ve interrumpida
- Si la Actividad 3 es la primera que acaba entonces la Actividad 2 continúa ejecutándose hasta que se acaba



Se pueden hacer **particiones** en un diagrama de actividades para identificar las acciones que tienen alguna **característica en común**. Por ejemplo que se llevan a cabo por un mismo **actor**.

Por ejemplo:

Indicar que es el usuario el que introduce el código y la cantidad

Indicar que es el sistema el que expulsa la tarjeta y muestra el mensaje de error.

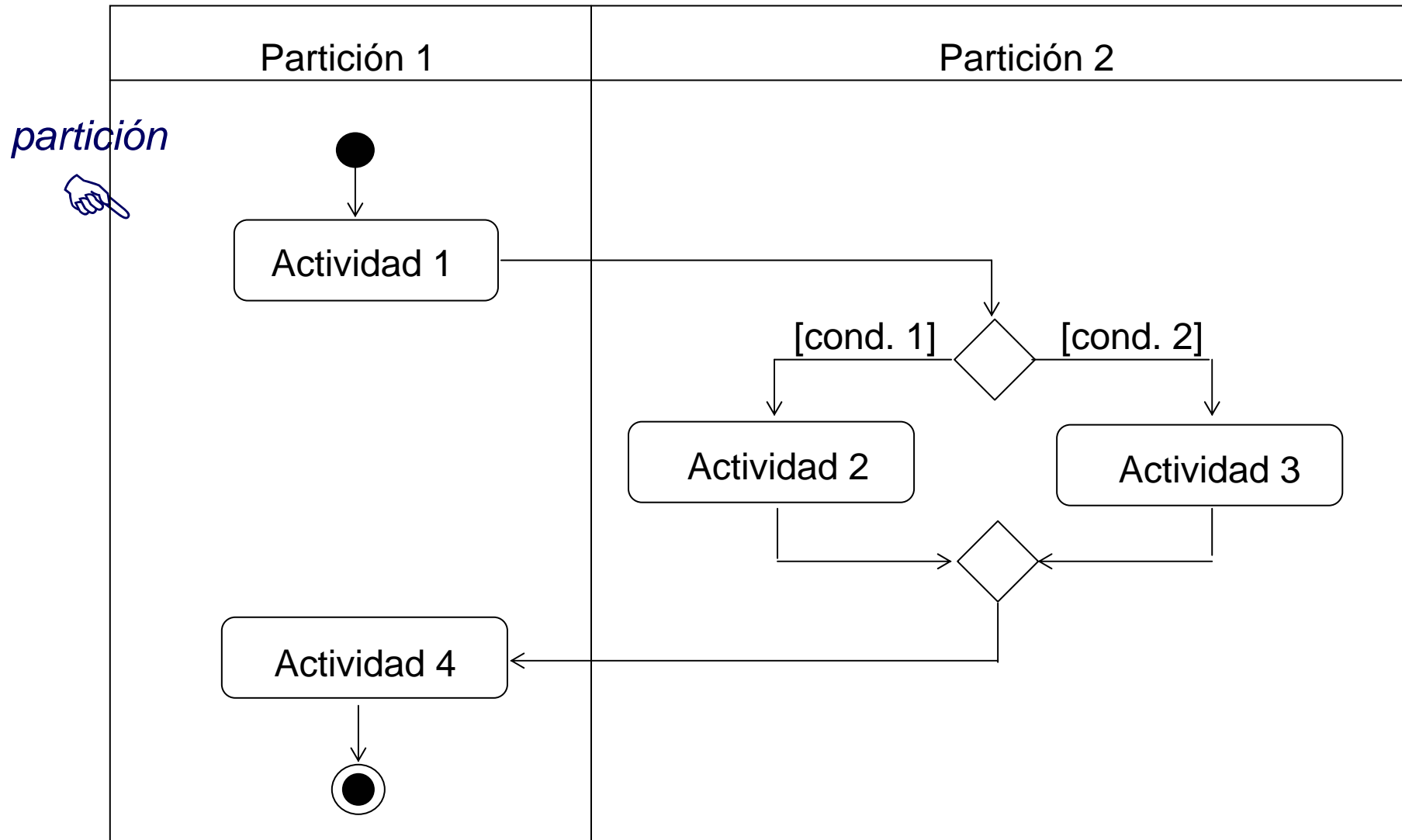


## Particiones:

El diagrama se divide en partes, agrupando las actividades que tienen algo en común

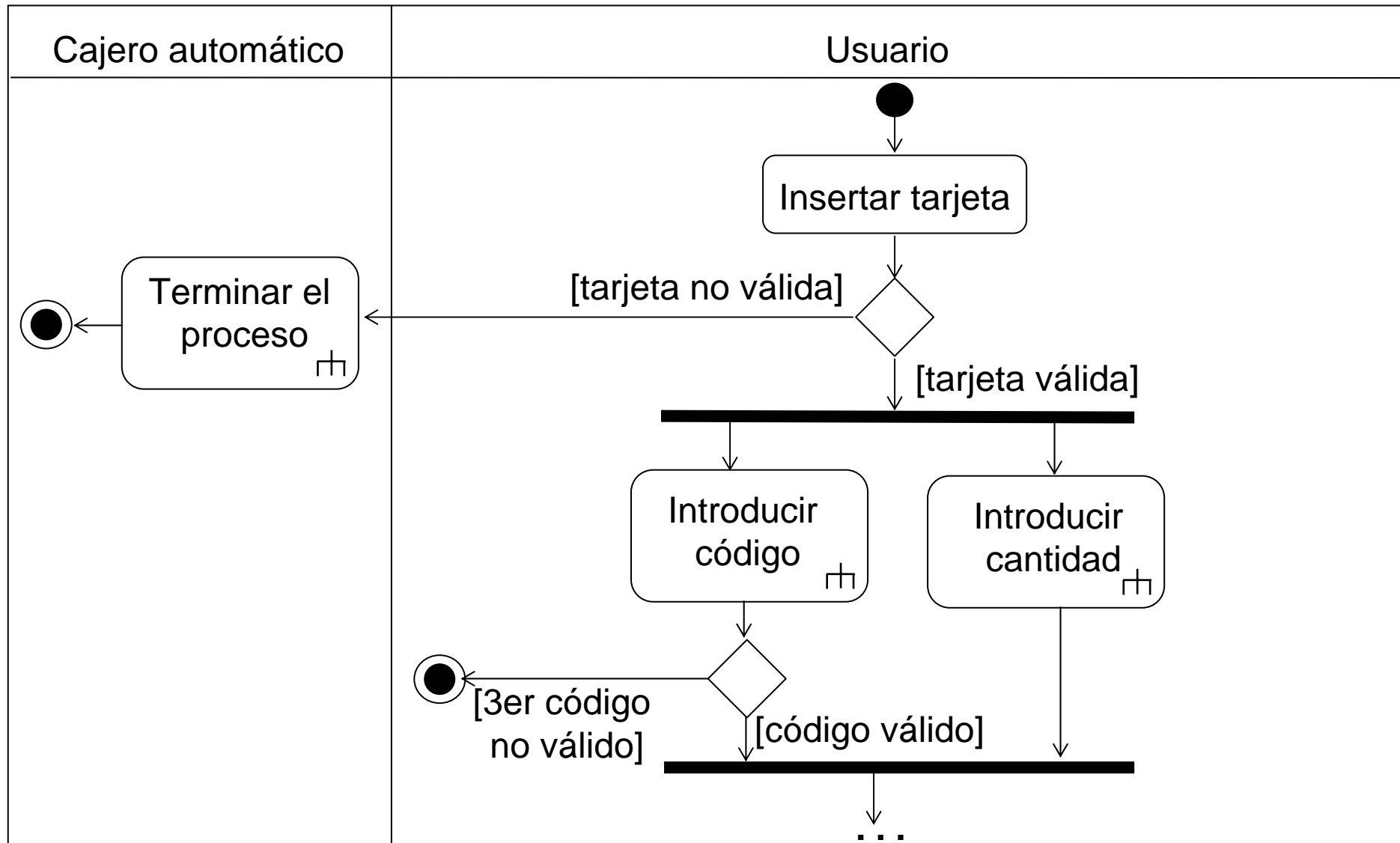



# Representación gráfica





# Ejemplos: cajero automático (versión 5)





# Reglas

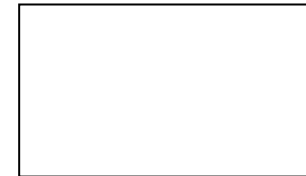


- Cada **actividad** debe estar en una partición
- No aconsejan representar diagramas con más de **cinco** particiones por simplicidad



## Objetos:

Objetos que se ven involucrados en las actividades

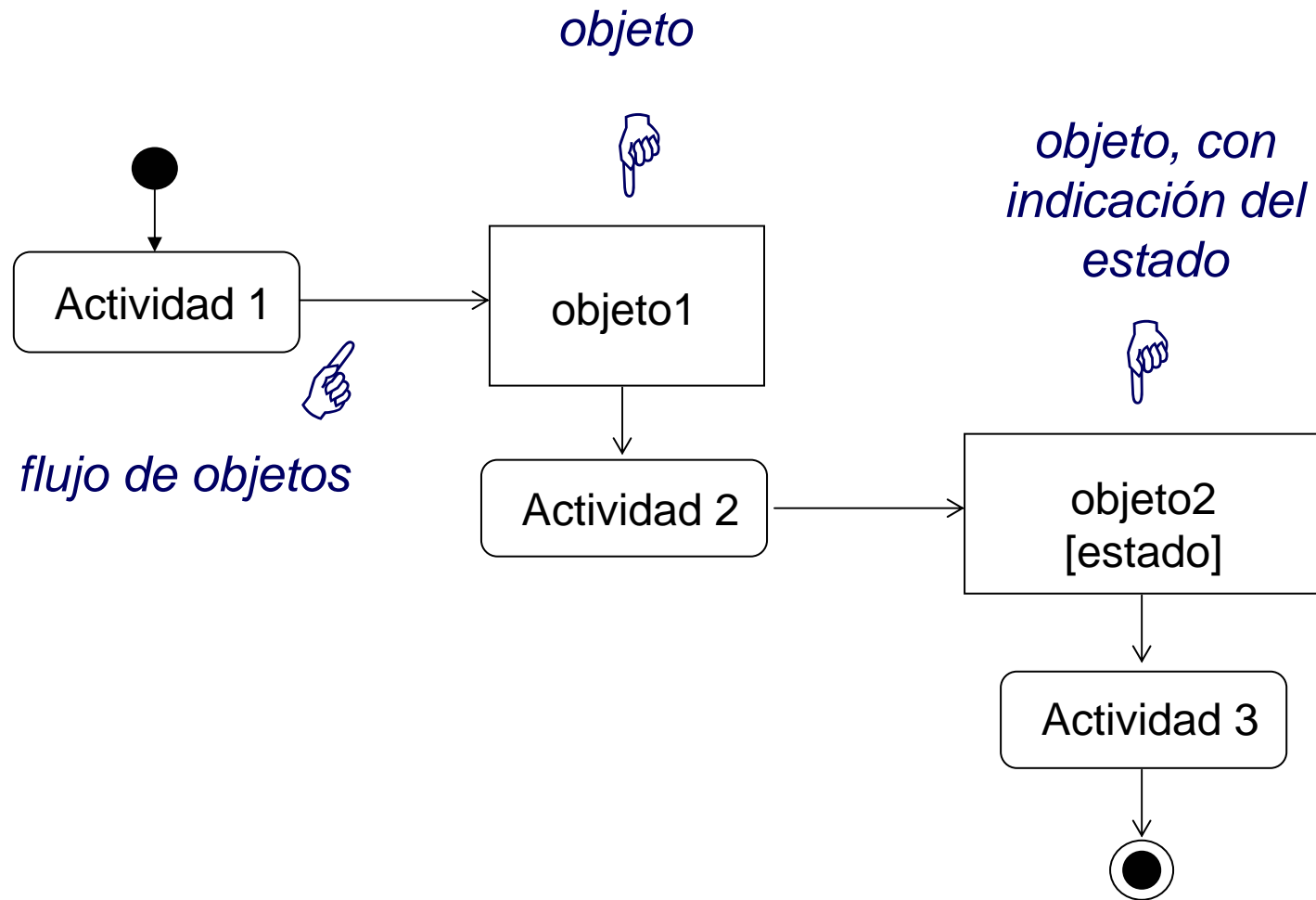


## Flujo de objetos:

Conectan los objetos con las actividades que los producen o los consumen.  
Un flujo de objetos de una actividad a otra implica un flujo de control.

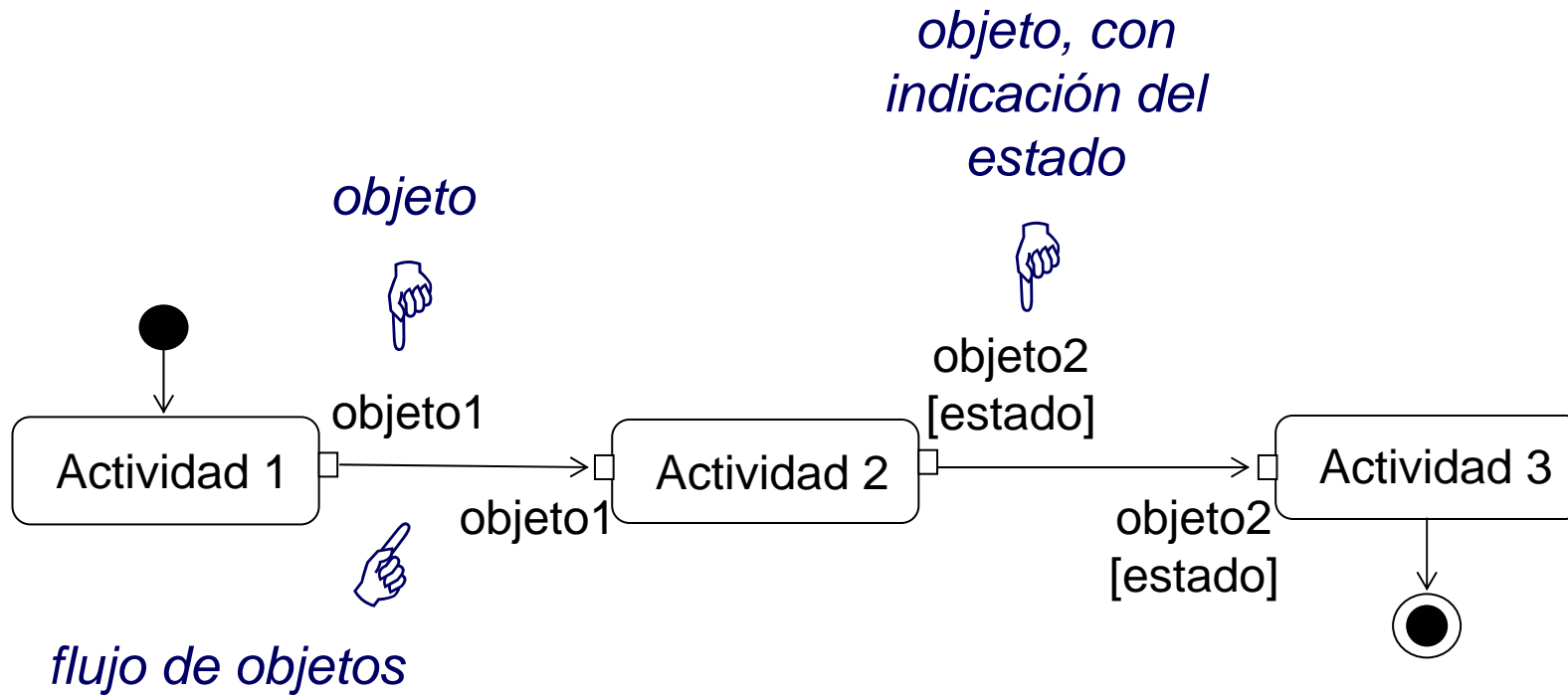






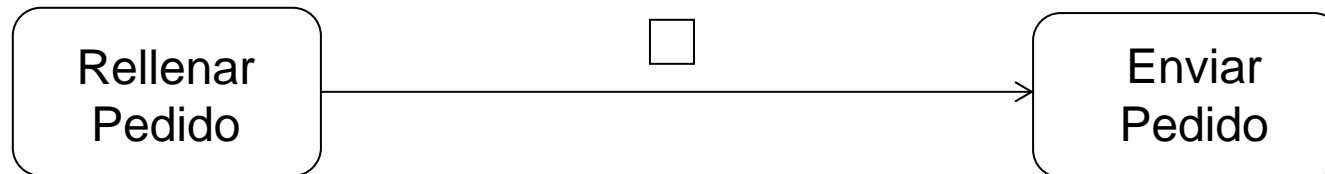
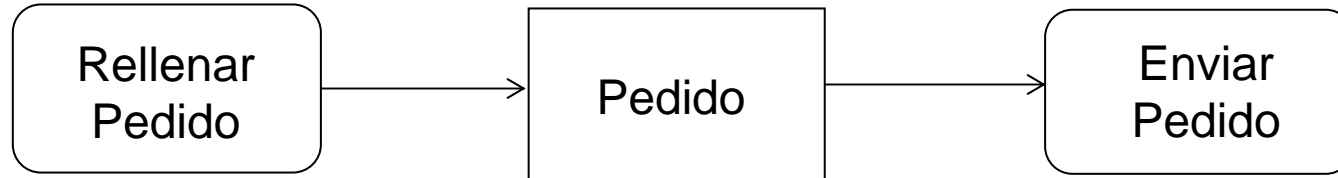


# Representación gráfica (alternativa)





# Ejemplo





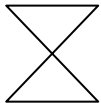
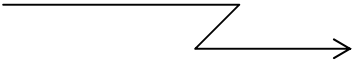



# Reglas



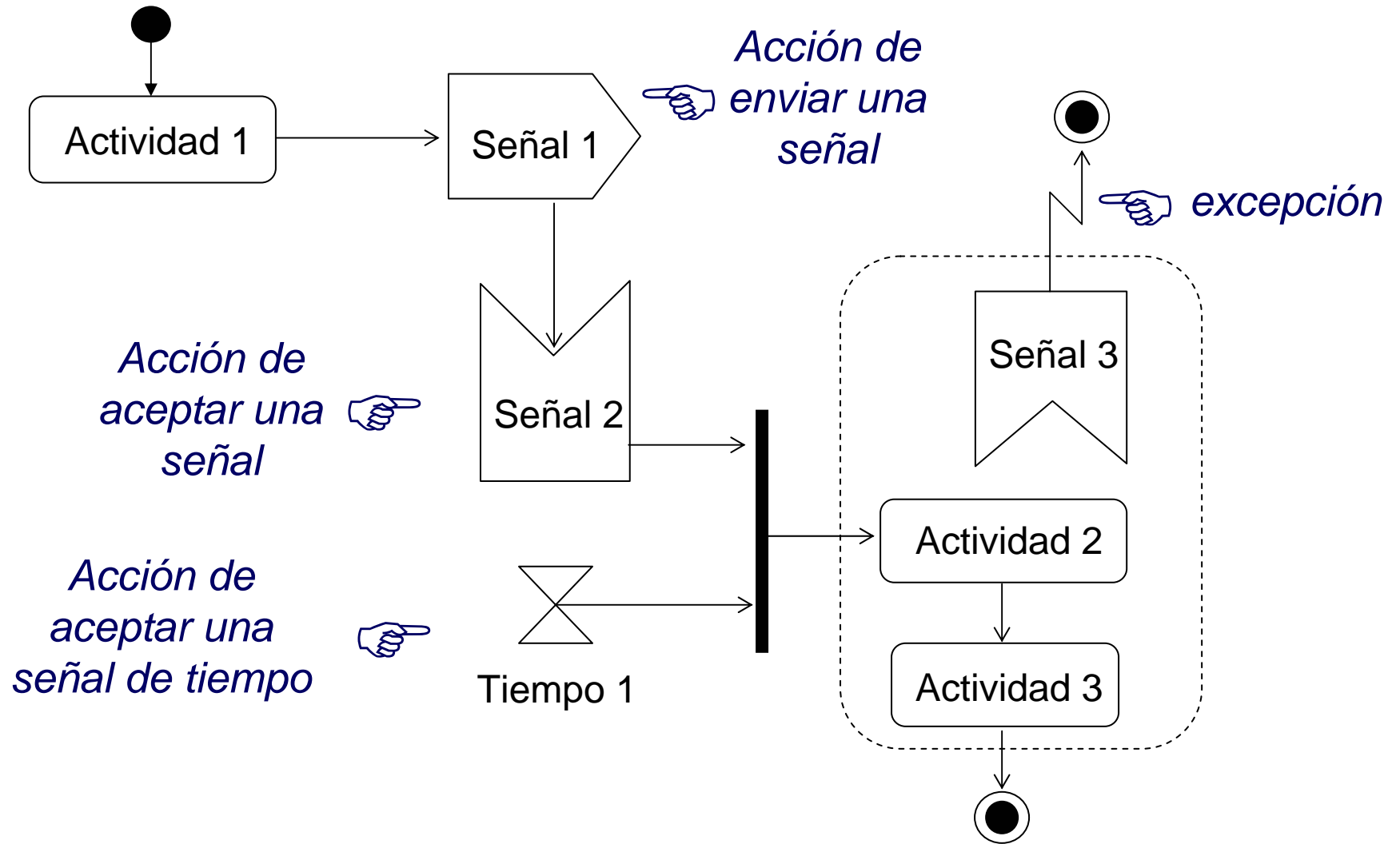
- Un flujo de objetos no puede iniciarse en un estado inicial
- Los objetos (por no tratarse de actividades) pueden aparecer encima de las líneas que sirven para dividir las calles de una partición



<p><b>Enviar (SendEventAction):</b> Representa la acción de enviar una señal</p>	
<p><b>Aceptar (AcceptEventAction):</b> Representa la acción de aceptar una señal.</p>	
<p><b>Aceptar evento temporal:</b> Tipo particular de acción 'aceptar' en la que la señal es una señal de tiempo.</p>	
<p><b>Excepción:</b> Representa la ocurrencia de una excepción.</p>	
<p><b>Región que se puede interrumpir:</b> Representa un grupo de actividades que se pueden interrumpir.</p>	

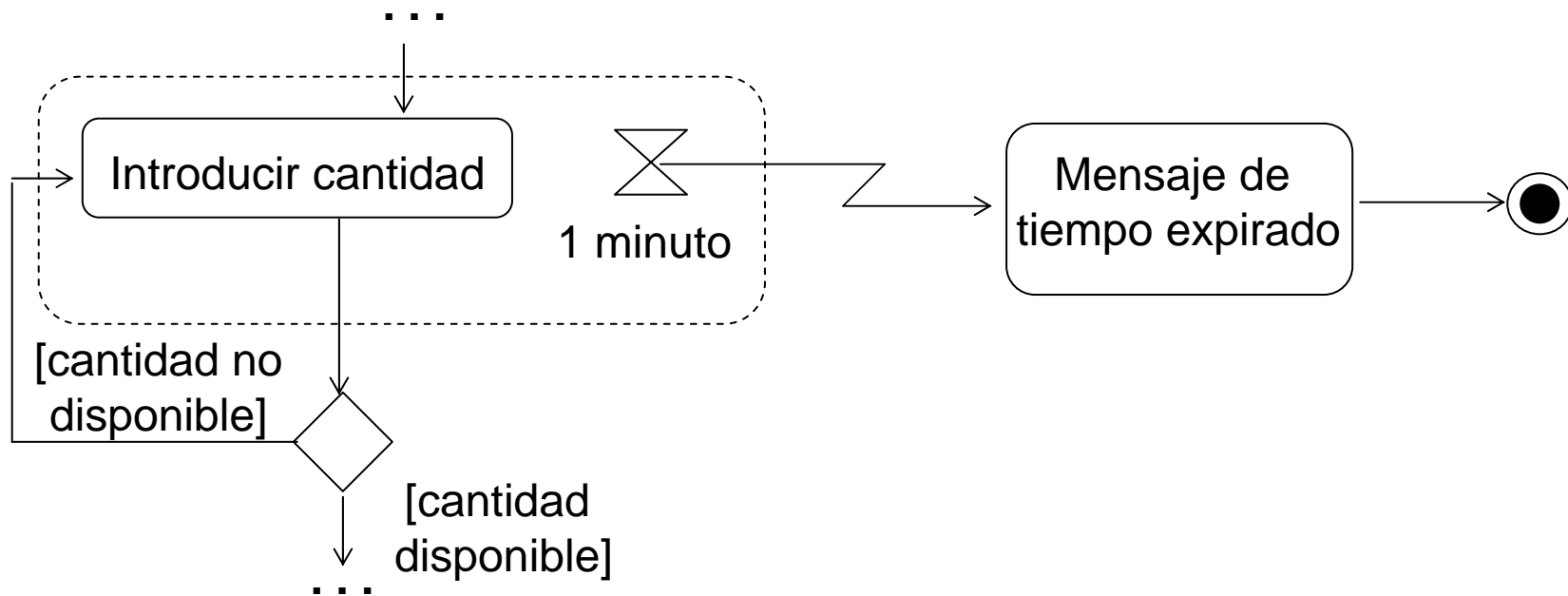
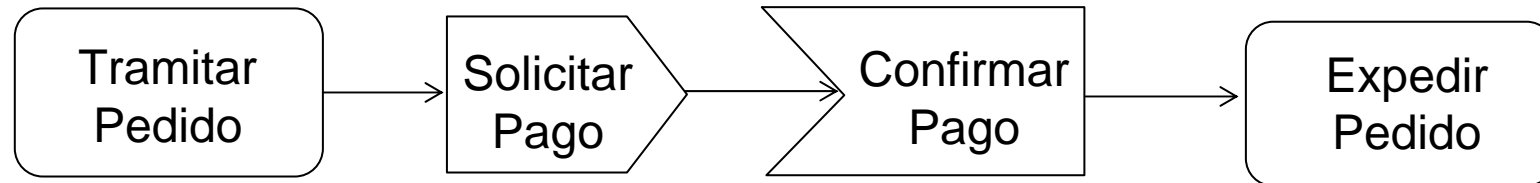


# Representación gráfica

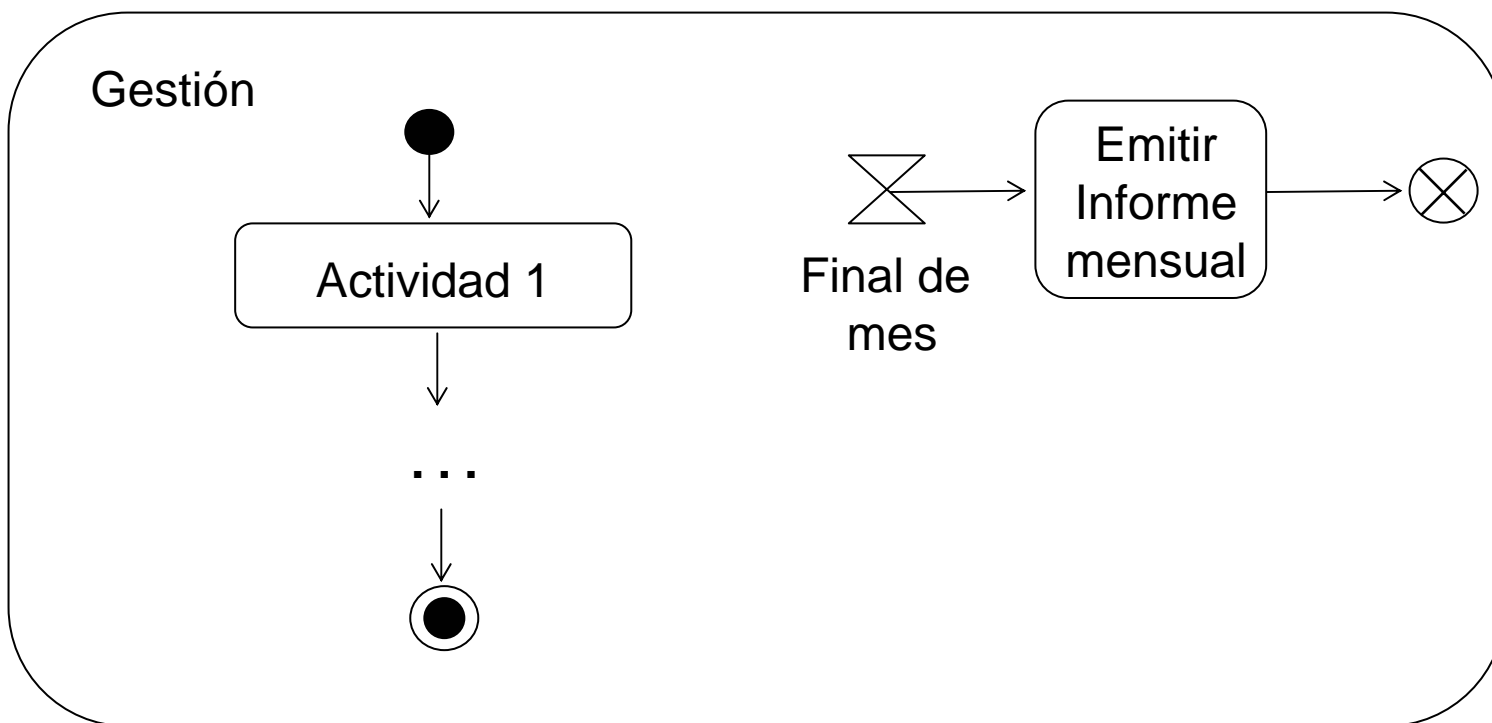




# Ejemplos



- Cuando un nodo 'aceptar' no tiene ningún flujo de entrada su comportamiento es especial. No acaba cuando acepta un evento sino que se queda a la espera de nuevos eventos mientras la actividad que lo contiene permanece activa.







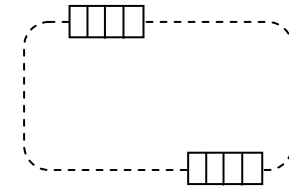
## Nodo de expansión:

Flujo de una colección a través de la frontera de una región de expansión



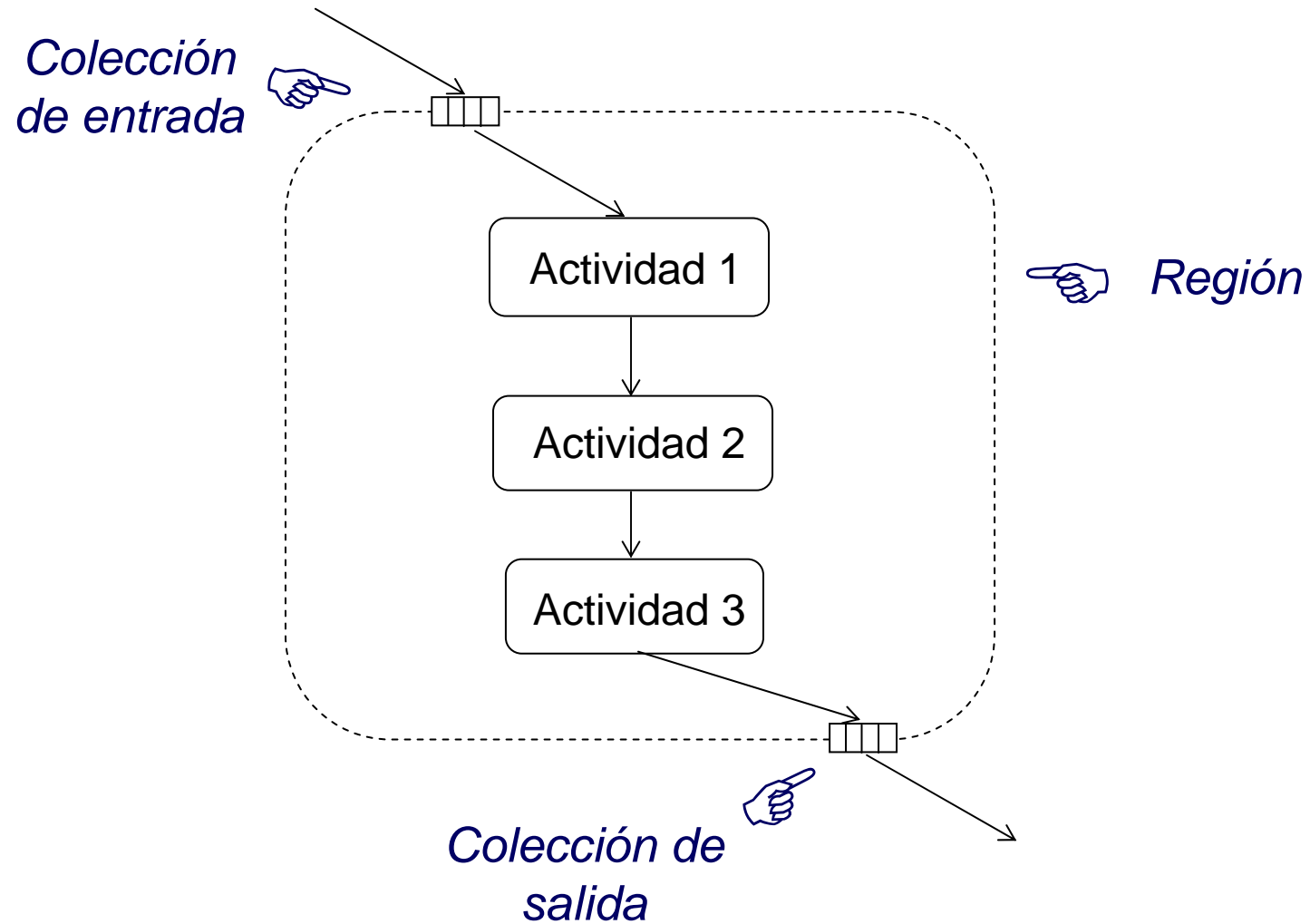
## Región de expansión:

Permite representar la ejecución de un bloque de actividades para cada elemento de una colección de entrada





# Representación gráfica





- Todos los vectores (de entrada y salida) deben tener el mismo **tamaño**
- Existe al menos un nodo de expansión de entrada y cero o más nodos de expansión de salida
- Si un nodo de expansión tiene nombre entonces corresponde al nombre de un elemento individual
- La **ejecución** para cada uno de los elementos puede ser:
  - **en paralelo**: las ejecuciones son independientes
  - **iterativa**: secuencial, una detrás de otra
  - **como corriente**: una vez empezada la ejecución sigue recibiendo elementos de entrada