

Diagrama de Actividades (DAct)

- Características del DAct
- Componentes del DAct
- Contracciones
- Repetición de actividades
- Bandas o carriles (*swimlanes*)
- Jerarquización

Características del DAct

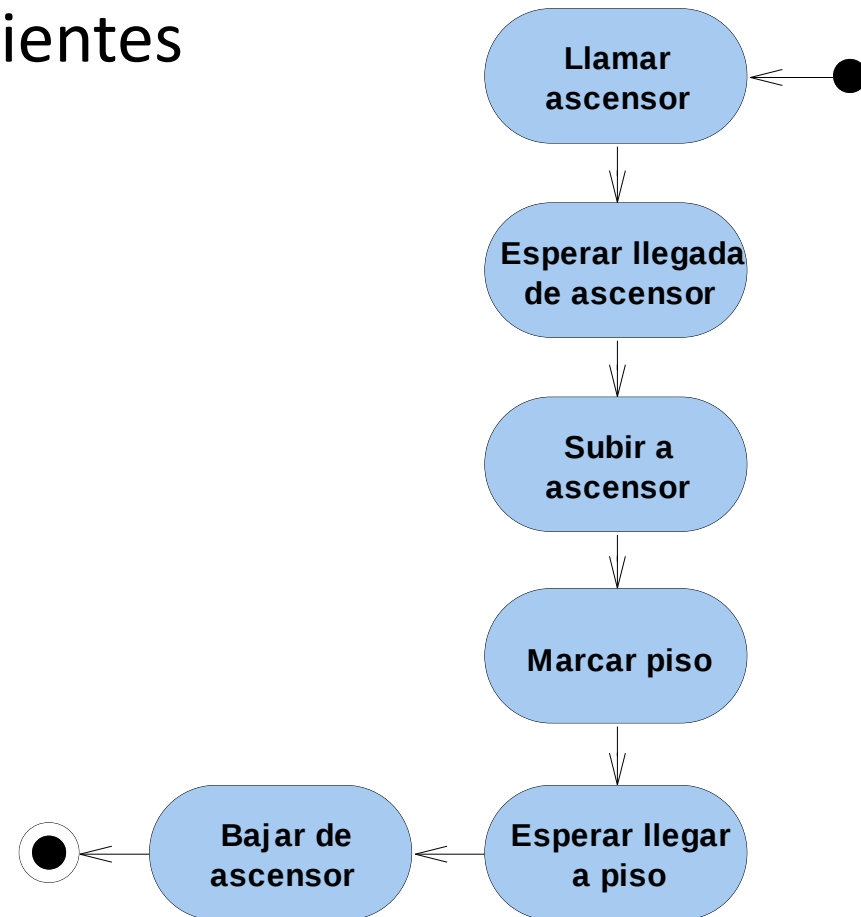
- Modelo híbrido propuesto a partir de:
 - diagrama de eventos de J. Odell
 - modelado de estados SDL (*Specification and Description Language*)
 - modelado de flujo de trabajo (*workflow*)
 - redes de Petri
- Es predominantemente dinámico (en términos de comportamiento), pero representa elementos funcionales (actividades).
- Describe las posibilidades procedimentales (organización temporal) del sistema sobre la base de actividades.

...Características del DAct

- Muestra una secuenciación de actividades con soporte para comportamiento condicional y concurrente.
- Responde a preguntas tales como:
 - ¿qué actividades se realizan en el sistema?
 - ¿cómo se organizan temporalmente las actividades?
 - ¿qué actividades pueden ejecutarse en paralelo?

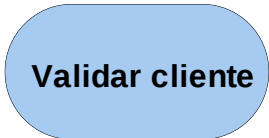
Componentes del DAct

- El DAct presenta los siguientes componentes:
 - actividad
 - transición
 - nodos división/unión
 - barras *fork/join*
- Ejemplo:



Actividad

- También se le denomina estado actividad.
- Es un paso o tarea simple dentro de un procedimiento.
- Por definición toma tiempo en concluirse.
- Representación:

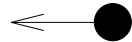


Validar cliente

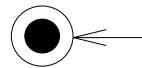
Transición

- También llamada transición de actividades.
- Representa el término de una actividad y el inicio de otra.
- Casos particulares:

- transición inicial



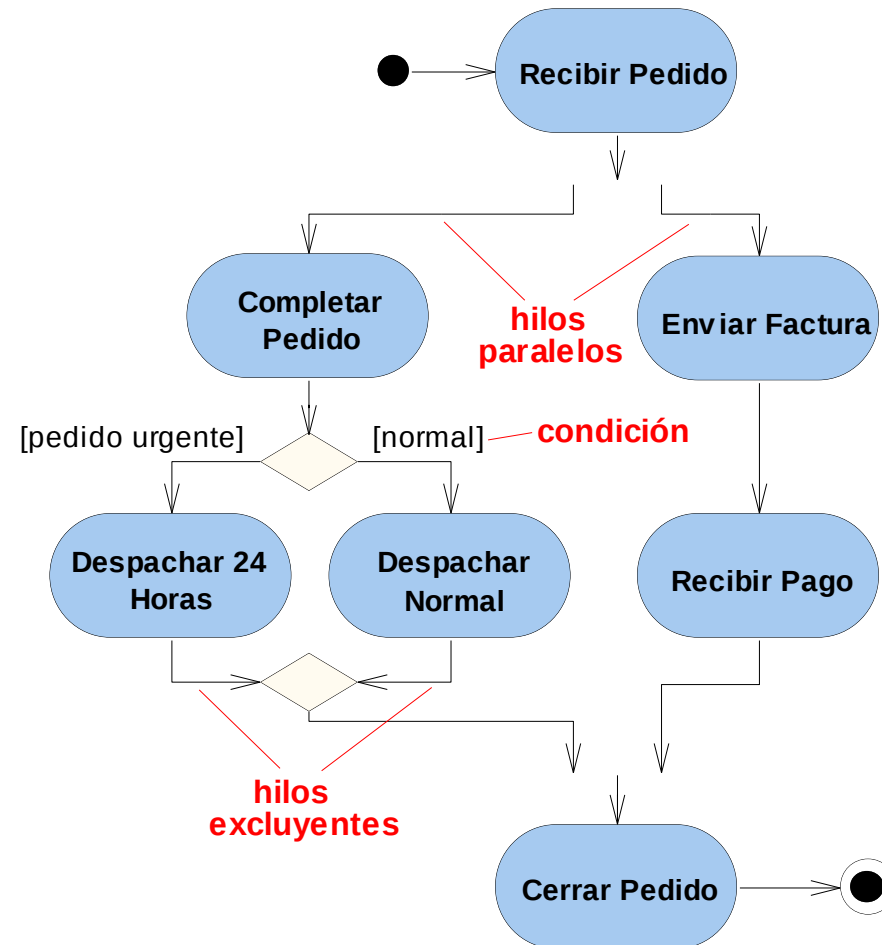
- transición final



Organización de actividades

- Actividades pueden organizarse:
 - en secuencia
 - condicionadas
 - en paralelo

- Ejemplo:

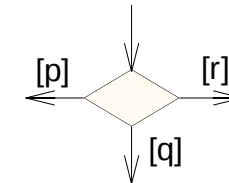


Nodos división/unión

- Permiten organizar actividades en forma condicionada:

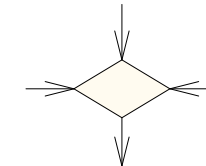
- **división** (*branch*): abre hilos excluyentes de actividad

- tiene una sola transición de entrada
- tiene 2 o más transiciones de salida con condiciones excluyentes entre si
- la única transición de salida, cuya condición es verdadera en el momento de la ejecución, es lanzada



- **unión** (*merge*): cierra los hilos excluyentes de actividad

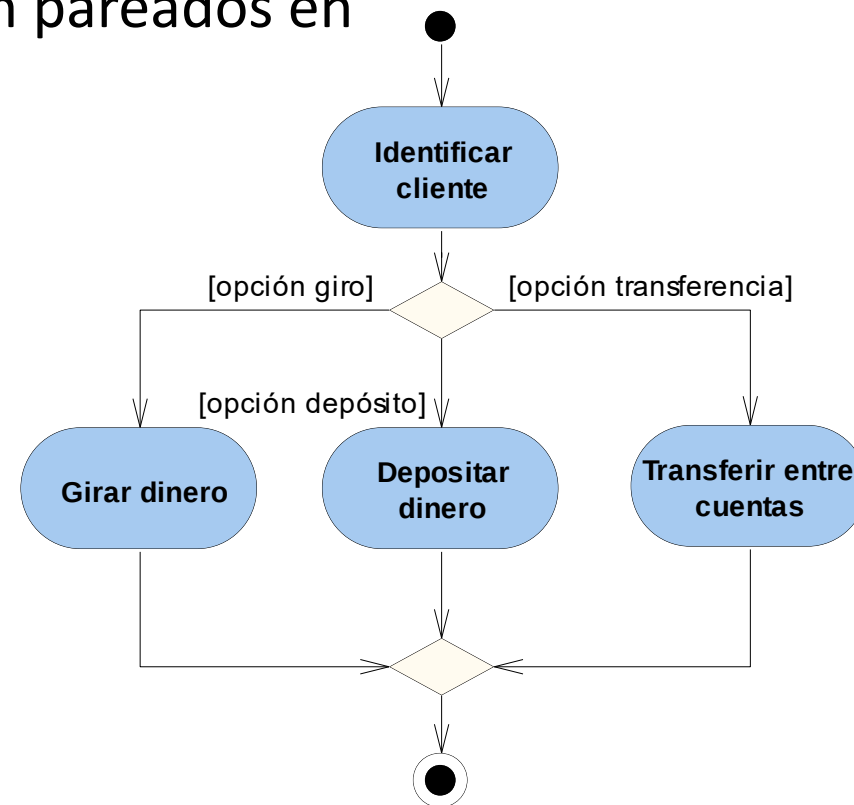
- tiene 2 o más transiciones de entrada y una sola de salida



...Nodos división/unión

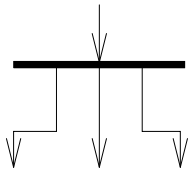
- **Uso disciplinado:** Se recomienda que los nodos división/unión vayan pareados en la mayoría de los casos.

- Ejemplo:



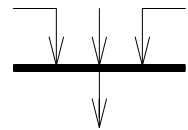
Barras *fork/join*

- Permiten organizar actividades en forma concurrente:



- **fork**: abre hilos paralelos de actividad

- cuando la única transición de entrada se realiza, las 2 o más transiciones de salidas son lanzadas en paralelo

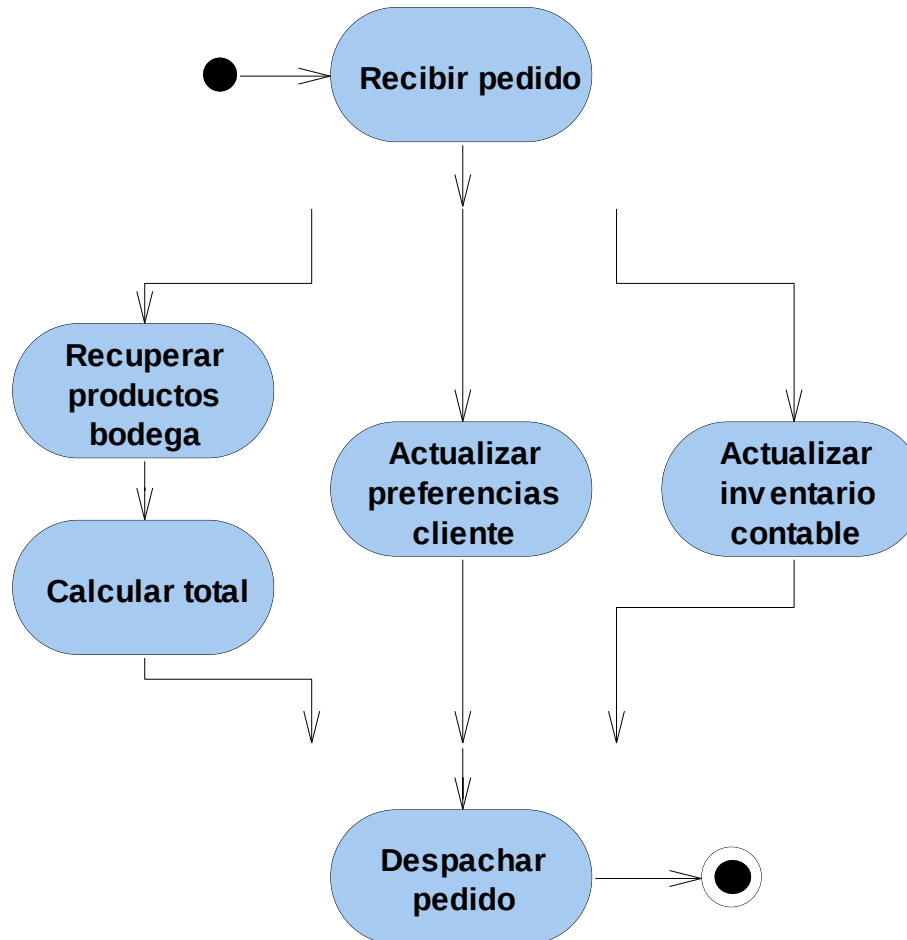


- **join**: cierra los hilos paralelos de actividad

- cuando las 2 o más transiciones de entrada se hayan realizado, la única transición de salida es lanzada

...Barras *fork/join*

- Ejemplo:

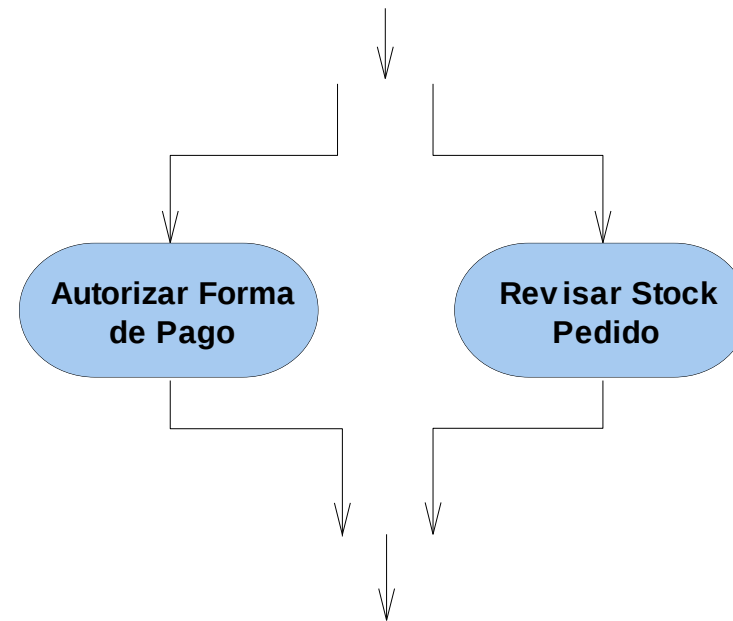


...Barras *fork/join*

- **Uso disciplinado:**

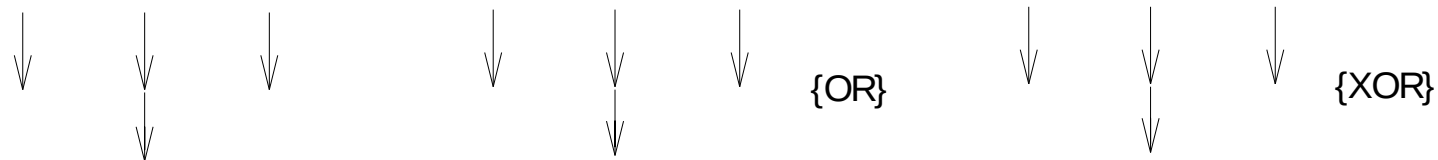
- Barras *fork/join* deben ir pareadas. Ejemplo:
- Para cada par de barras *fork/join*:

Nº de transiciones de salida de la barra *fork* = Nº de transiciones de entrada de la barra *join*



...Barras *fork/join*

- La barra *join* espera a que **todas** las transiciones de entrada se hayan realizado $\Rightarrow$ sincronización **AND** por omisión
- Otras formas de sincronización:
 - **OR**: al menos una de las transiciones se realizó
 - **XOR**: exactamente una de las transiciones se realizó
- Representación:

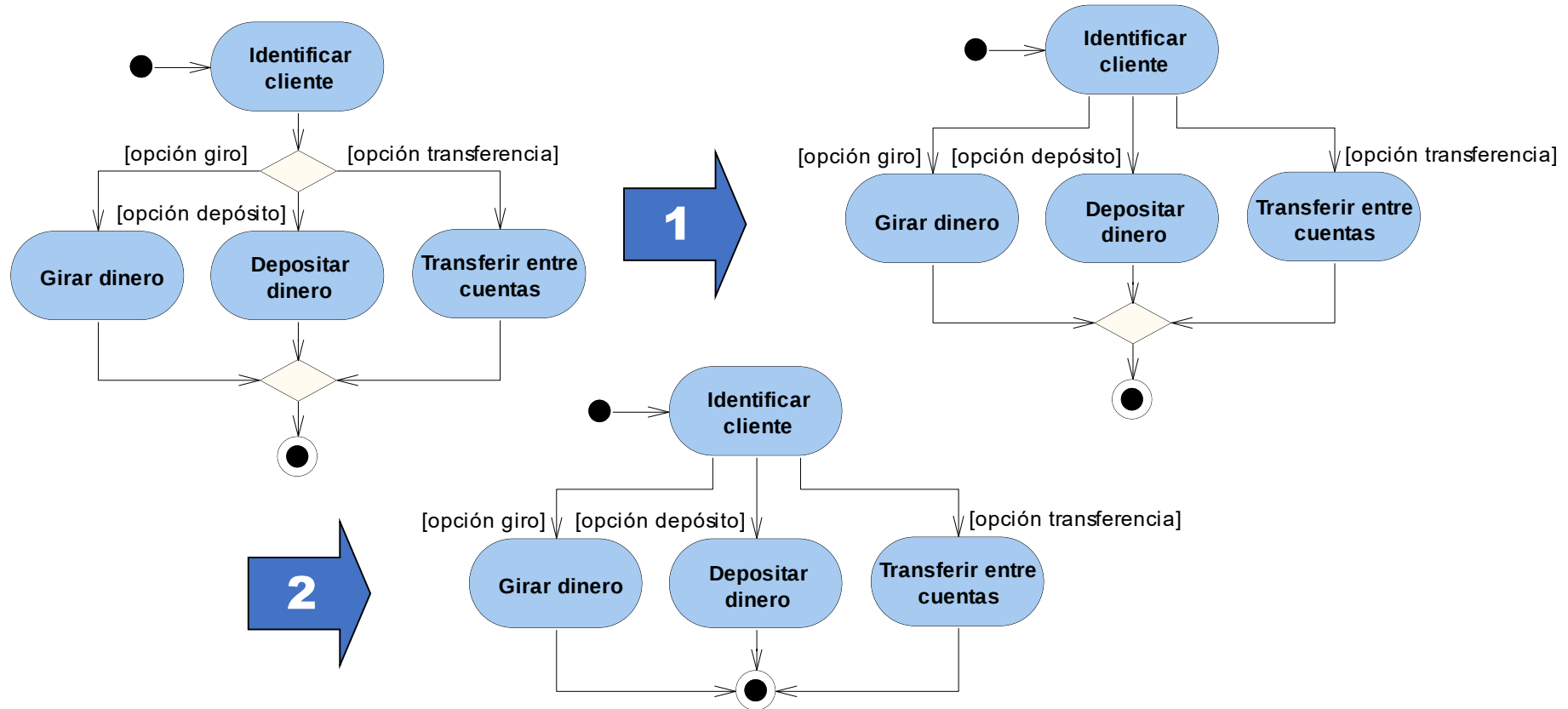


Contracciones

- Algunos componentes consecutivos del DAct pueden contraerse o fundirse para efectos de simplificación.
- Casos posibles:
 - En DAct simples uno o ambos nodos de división/unión puedan fundirse con actividades que los anteceden/suceden.
 - Transiciones inicial/final pueden fundirse con nodos división/unión.

...Contracciones

- ...Casos posibles: ejemplos

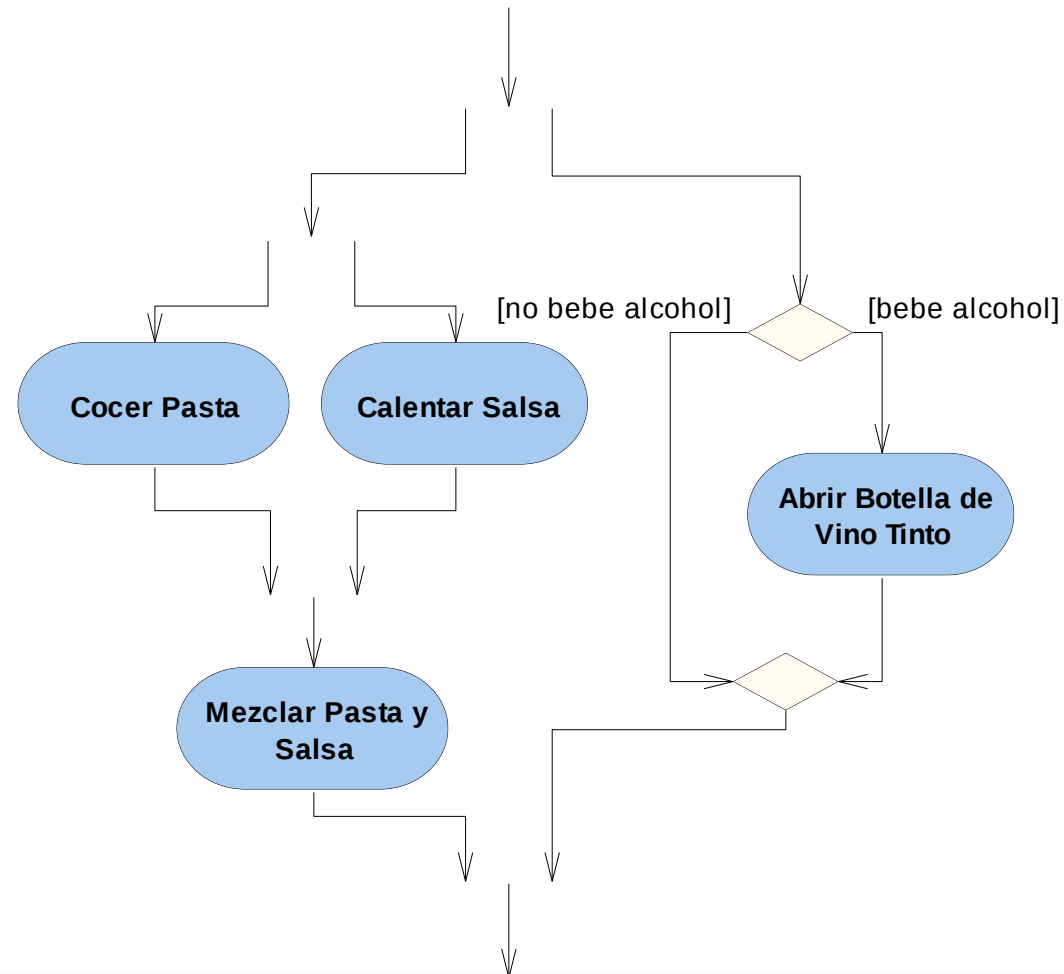


...Contracciones

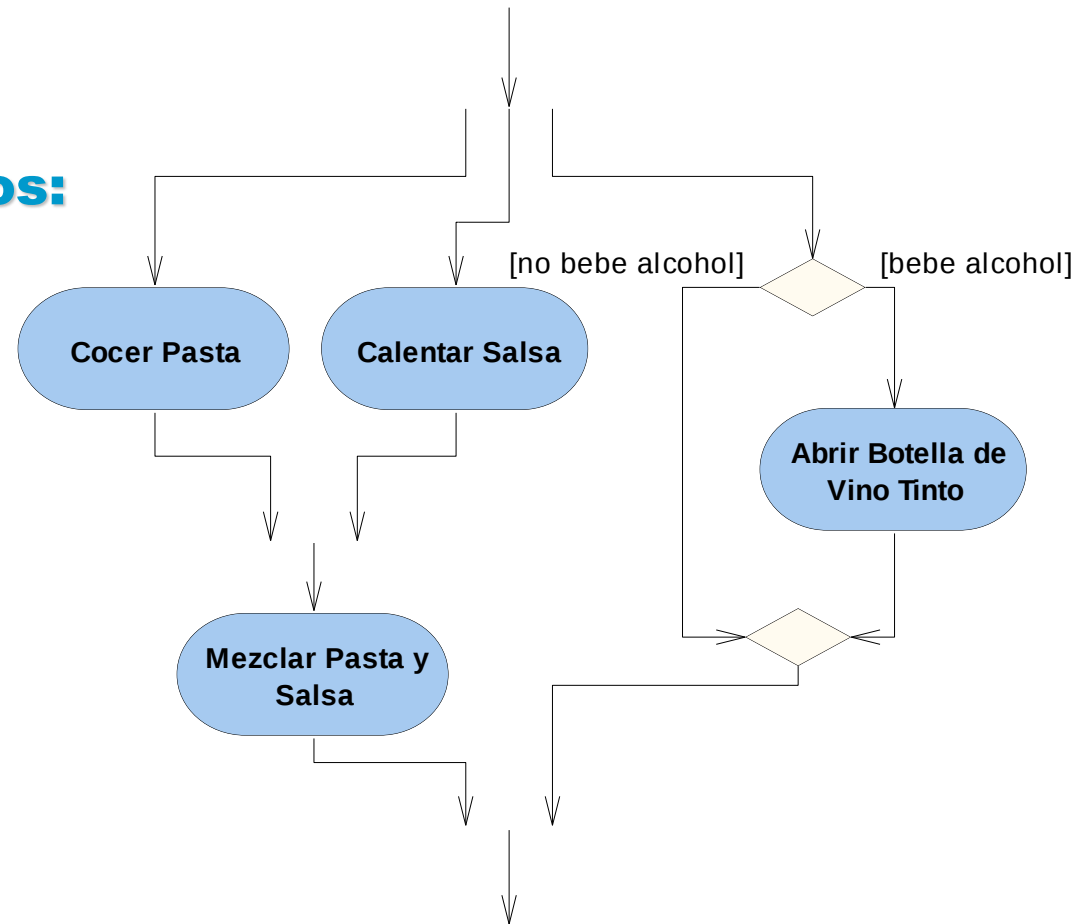
- ...Casos posibles:
 - Barras *fork* o barras *join* consecutivas pueden fundirse. Esto sigue respetando el uso disciplinado a pesar de que:
 - haya un número impar de barras en el DAct
 - sea diferente el nº de transiciones de salida y entrada entre barras *fork* y *join*
 - Barras *fork* puede fundirse con nodos de división, cuando existen hilos excluyentes sin actividades, creando hilos condicionados $\Rightarrow$ reinterpretación de las barras *fork/join*

Ejemplo de contracciones

inicial:

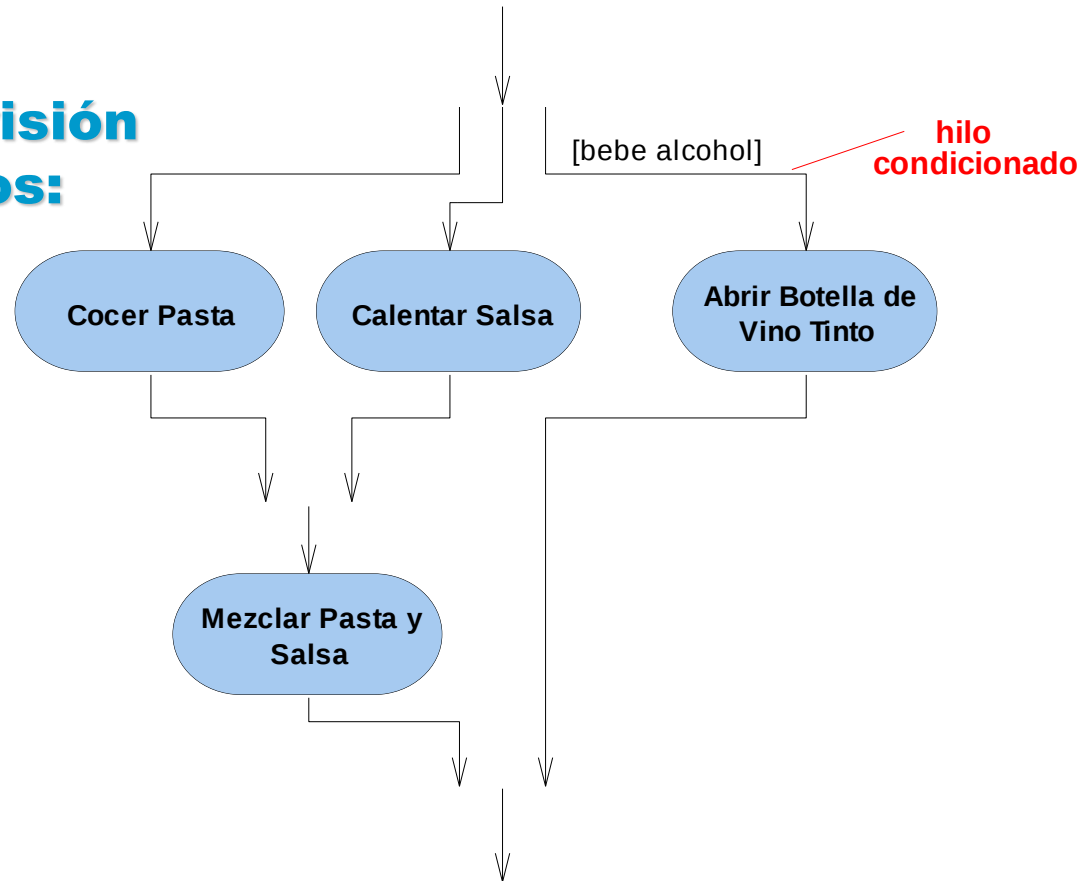


forks
fusionados:



...Ejemplo de contracciones

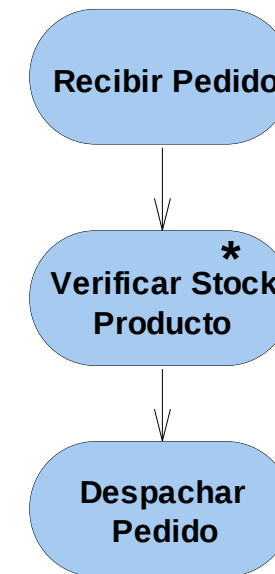
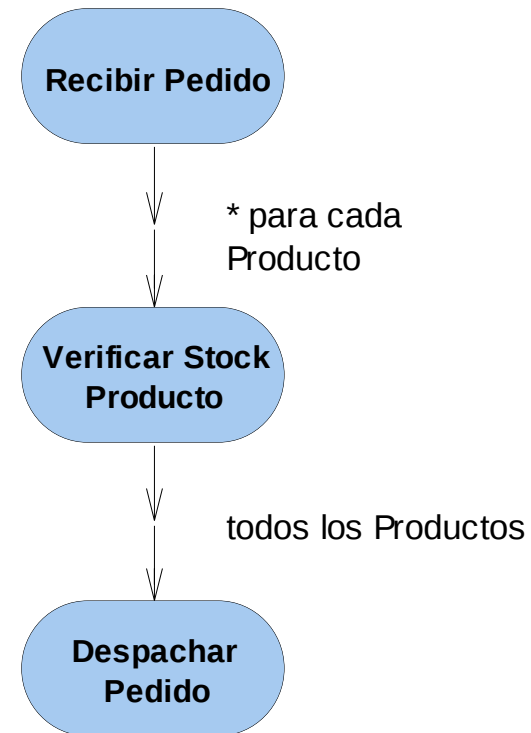
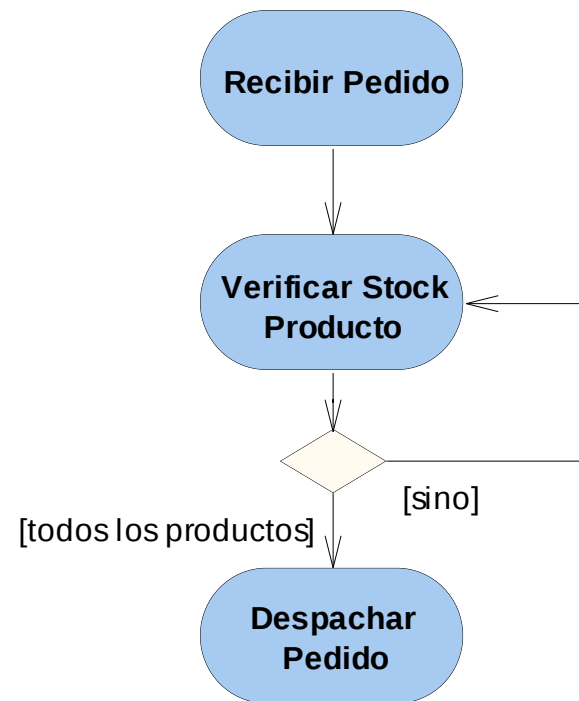
***fork* y división
fusionados:**



Repetición de actividades

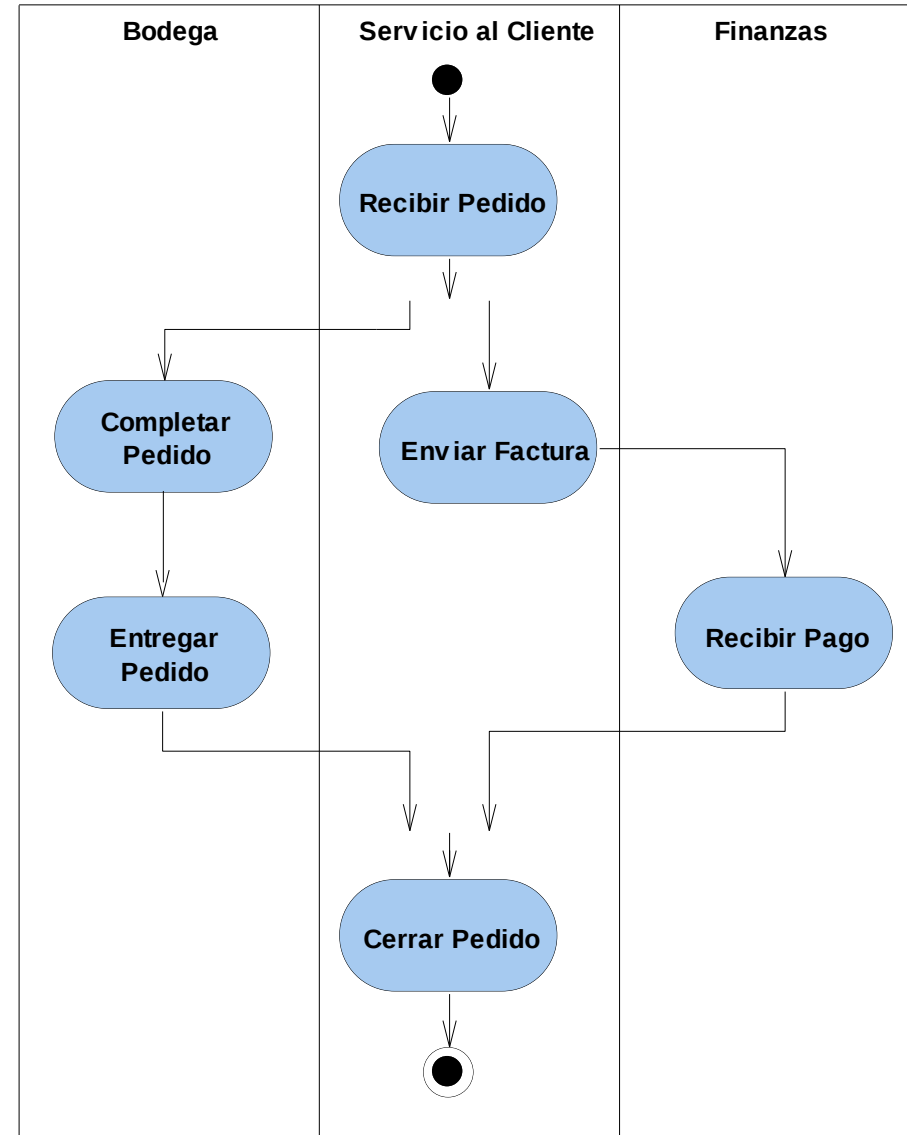
- Formas para repetir una actividad:
 - **nodo división**: representa una repetición secuencial
 - **barras *fork/join***: representa explícitamente una repetición concurrente
 - **concurrency dinámica**: representa implícitamente una repetición concurrente

Ejemplos de repetición



Bandas o carriles (*Swimlanes*)

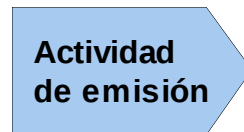
- Indican los responsables de la realización de las actividades, tales como:
 - cargos
 - sistemas informáticos
 - unidades organizacionales
 - organizaciones
- Ejemplo:



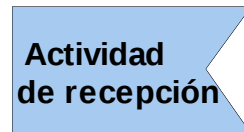
Otras notaciones

- Actividades de emisión y recepción de datos (o señales) pueden anotarse de forma distinta:

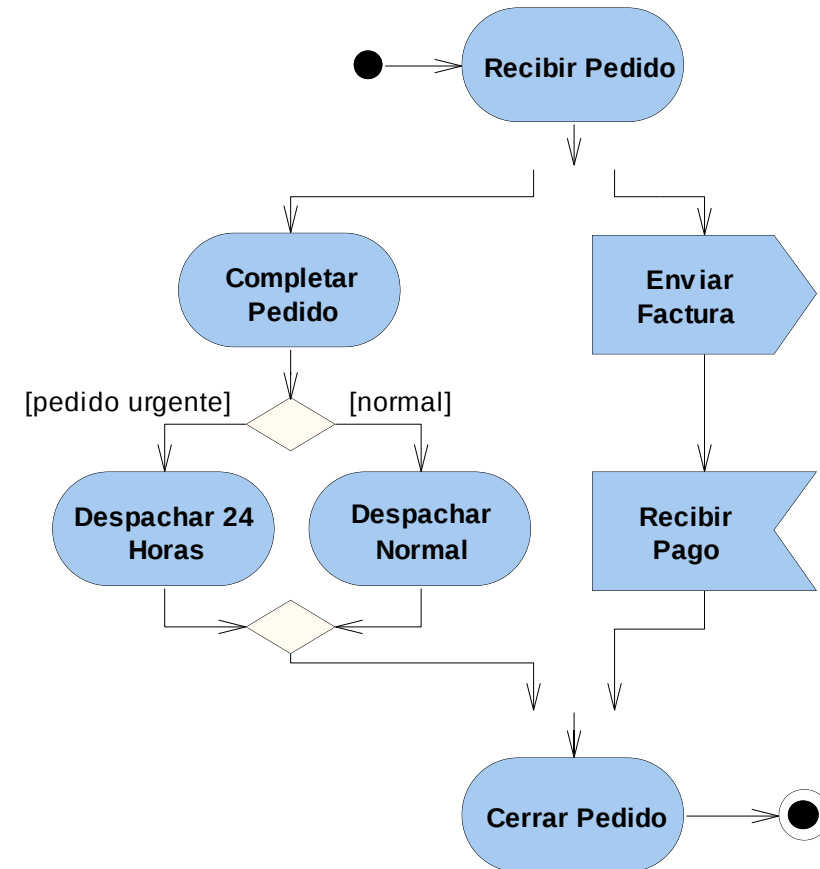
- emisor



- receptor



- Ejemplo:



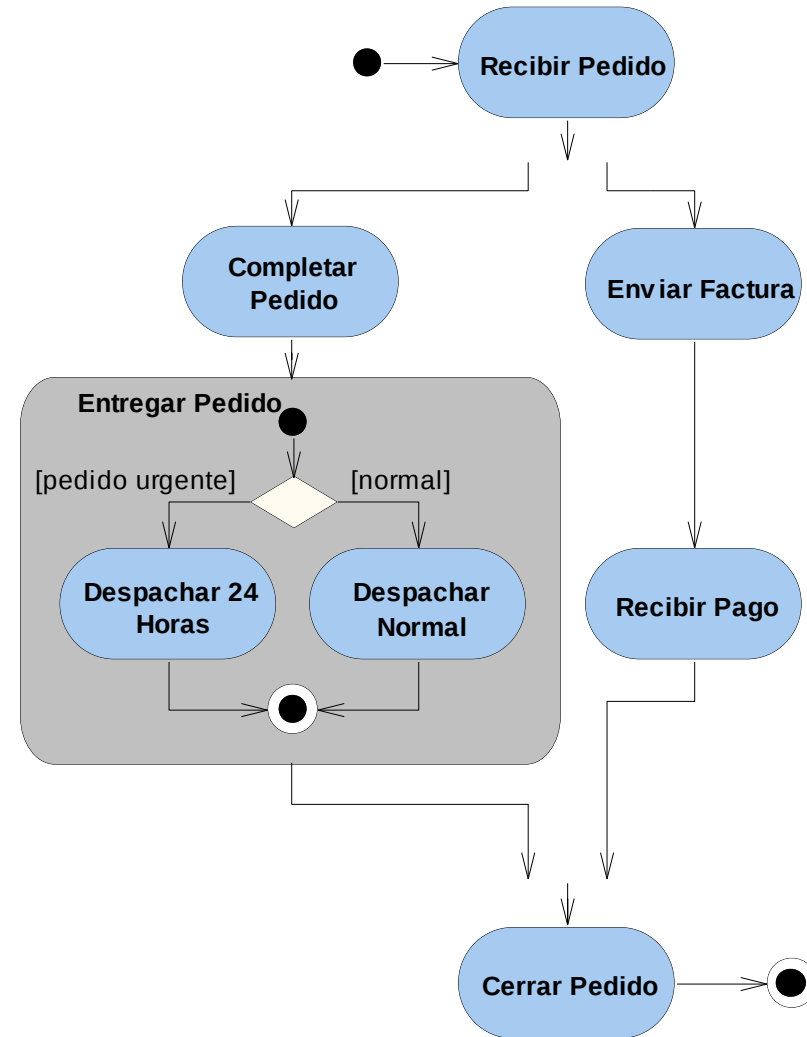
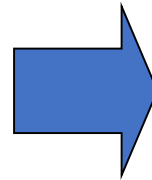
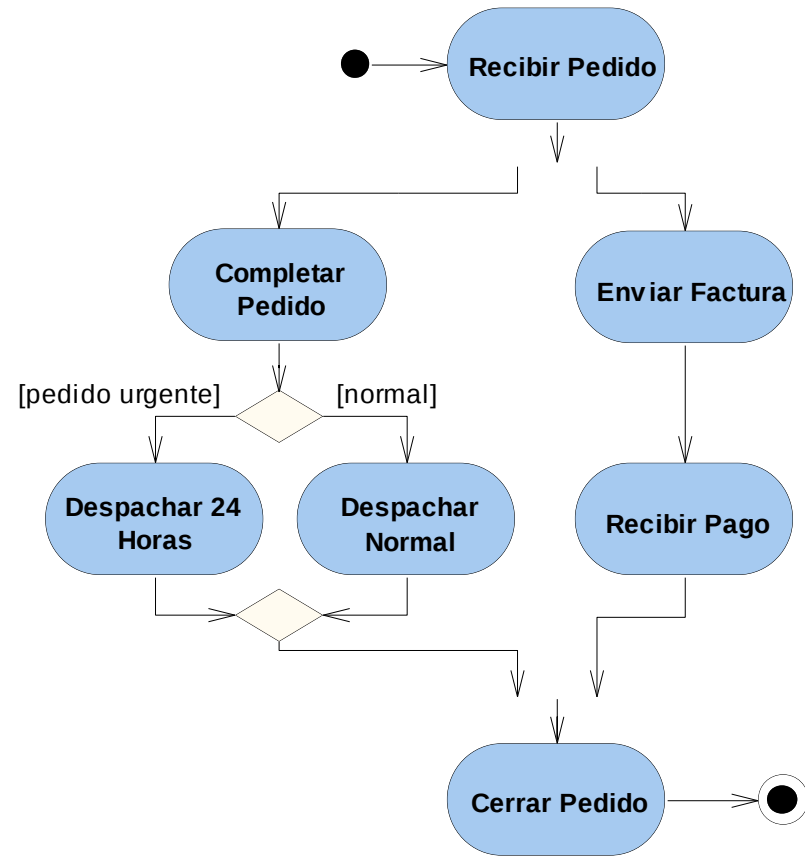
Uso del DAct

- Privilegia el comportamiento concurrente $\Rightarrow$ útil para *workflow* y procesos de negocio.
- Para construir DAct se recomienda en general:
 1. Identificar las actividades
 2. Organizar temporalmente las actividades
 3. Jerarquizar el DAct si necesario

Jerarquización

- Se usa preferentemente descomposición de actividades:
 - agrupar una porción de un DAct y representarla como un DAct separado (DAct hijo)
 - reemplazar la porción por un actividad compuesta en el DAct original (DAct padre)
- Descomposición de actividades permite:
 - desacoplar los DAct padre e hijo
 - reutilizar actividades compuestas

...Jerarquización



...Jerarquización

