

# Ingeniería del Software I

## Introducción a los Diagramas de Actividad

2do. Cuatrimestre 2005

<b>INTRODUCCIÓN .....</b>	<b>2</b>
<b>SEMÁNTICA .....</b>	<b>2</b>
<b>NOTACIÓN .....</b>	<b>3</b>
ESTADO ACCIÓN .....	3
<i>Transiciones Simples.....</i>	<i>3</i>
<i>Estados Acción Compuestos.....</i>	<i>3</i>
<i>Estados Acción Iniciales y Finales.....</i>	<i>4</i>
<i>Decisiones .....</i>	<i>4</i>
<i>Andariveles.....</i>	<i>4</i>
<i>Transiciones Concurrentes (Fork y Join).....</i>	<i>6</i>
<b>EJEMPLO .....</b>	<b>7</b>

## Introducción

El objetivo de este breve apunte es describir los Diagramas de Actividad, propuestos por el lenguaje estándar de modelado de sistemas de software UML (Unified Modelling Language). Casi todas las definiciones de este apunte fueron tomadas de la guía semántica de UML, versión 1.1. Sin embargo, se hicieron algunas modificaciones menores para limitar estos diagramas al alcance con el que se los quiere utilizar en la materia, de tal forma que no garantizamos “compatibilidad” entre los diagramas que resultan de seguir este apunte y las definiciones formales del UML.

Algunos aspectos de los Diagramas de Actividades no fueron incluidos en este apunte ni serán usados en la materia, pero pueden ser consultados por los alumnos en la documentación de UML. Para más información, se pueden utilizar los punteros de la página de Web de la materia.

## Semántica

Estos diagramas muestran básicamente actividades, representando la realización de operaciones y transiciones entre ellas, representando el flujo de control necesario entre ellas, para completar la operación.

Un diagrama de actividad puede estar asociado a la implementación de un caso de uso, con el propósito de enfocarse en los flujos manejados por el procesamiento interno (en contraposición con eventos externos).

Se debe usar diagrama de actividad en situaciones donde todos o la mayoría de los eventos representan la finalización de acciones generadas internamente (esto es, flujo de control procedural). Este tipo de diagrama no es adecuado en situaciones donde ocurren eventos asincrónicos.

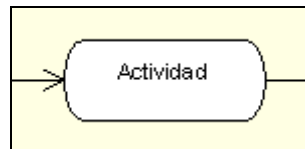
Teniendo en cuenta que los casos de uso se centran en la interacción entre el actor y el sistema y no en el procesamiento interno del sistema durante el caso de uso, aparece la necesidad de utilizar este diagrama para evitar que la documentación de las actividades que realiza el sistema no esté limitada al texto informal de los casos de uso. De esta forma, un caso de uso puede estar acompañado por cero, uno o más diagramas de actividad.

Si resulta necesario, se pueden construir diagramas de actividad jerárquicos, donde una actividad de un diagrama sea descompuesta en actividades menores en un diagrama de nivel inferior.

## Notación

### **Estado Acción**

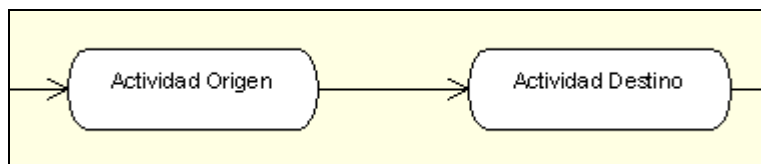
Un *estado de acción*, o *acción* simplemente es una representación de un estado con una acción interna y al menos una transición saliente que es el evento implícito de finalización de la acción interna. Las acciones no deben tener transiciones internas o transiciones salientes basadas en eventos explícitos: se deben usar otras acciones para esta situación. El uso normal de una acción es modelar un paso o un conjunto de pasos en la ejecución de un algoritmo (un procedimiento).



Los estados acción se representan con un rectángulo de ángulos redondeados.

### **Transiciones Simples**

Las transiciones simples representan el paso de una actividad a otra. Las transiciones siempre se disparan de forma inmediata con la finalización de la actividad origen, tienen duración 0 (se las considera instantáneas) y no tienen efectos laterales. En algunas situaciones podremos encontrar transiciones condicionales, que se dispararán una vez finalizada la actividad, pero siempre y cuando se cumpla una determinada condición. Cuando se utilizan transiciones condicionales, las condiciones deben ser excluyentes y se debe garantizar que siempre exista una transición viable (en la materia no veremos transiciones condicionales, para dar el mismo sentido usaremos decisiones).



*Las transiciones simples se representan con una flecha que sale desde la actividad origen y termina en la actividad destino.*

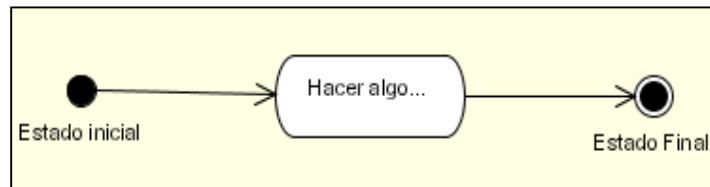
### **Estados Acción Compuestos**

Si resulta necesario, se pueden construir diagramas de actividad jerárquicos, donde una actividad de un diagrama sea descompuesta en subactividades, representándose esto en un diagrama de nivel inferior. A esta actividad se la llama estado o acción compuesta.

Los estados acción compuestos se representan igual que los estados acción, pudiendo tener una indicación gráfica en su zona inferior derecha para notar que se trata de un estado compuesto.

### Estados Acción Iniciales y Finales

Dentro de un diagrama de actividades tenemos pseudoacciones iniciales y acciones finales. El inicio de las acciones de un diagrama de actividad se da a partir de una pseudoacción inicial. Una transición a una acción final representa la finalización del diagrama de actividad.

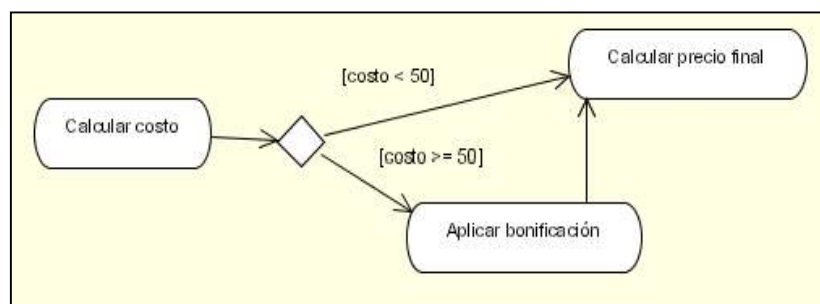


Los pseudoestados iniciales se indican con un círculo negro (sólido) del cual sale una transición al estado acción que se ejecuta en primer lugar.

Los estados finales se indican con un círculo negro (sólido) dentro de otro círculo, al cual le llegan transiciones desde las actividades que son las últimas en ejecutarse.

### Decisiones

Un diagrama de actividad expresa una decisión cuando una condición es usada para indicar diferentes transiciones posibles que dependen de un valor booleano. Cuando se utilizan decisiones, las condiciones deben ser excluyentes y se debe garantizar que siempre exista una transición viable. Se suele utilizar la condición *else* para indicar que una transición se dispara cuando ninguna otra condición resultó verdadera.

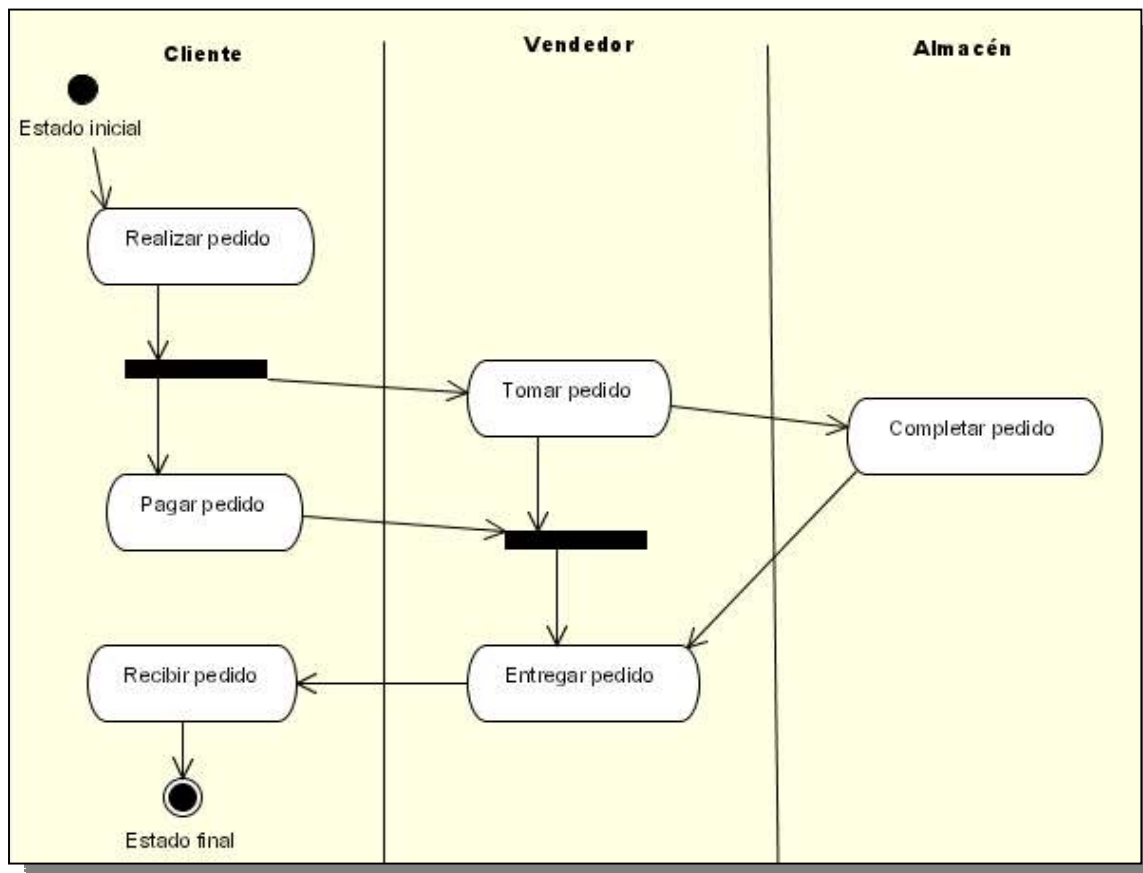


Las decisiones se indican con un rombo al cual le llegan transiciones y del cual salen dos o más transiciones rotuladas según las condiciones para realizar esa transición.

### Andariveles

Las acciones pueden ser organizadas en *andariveles*. Los andariveles se usan para organizar las responsabilidades de las actividades. Usualmente corresponden a unidades organizacionales dentro de un modelo de negocio (por ejemplo áreas de una empresa). No debemos olvidar que cuando estamos modelando los casos de uso, las actividades que realiza el sistema que estamos empezando a idear pueden ser llevadas a cabo tanto por máquinas como por personas que pertenezcan a distintas áreas de la organización. La utilidad de los andariveles aparece en estos casos, cuando quiero mostrar que la secuencia de pasos que el usuario está expresando como parte del procesamiento del sistema es realizada por personas de distintas áreas o distintos tipos de máquinas.

Al hacer esto puede parecer que uno se está adelantando al diseño, ya que cuando escribimos los casos de uso estamos modelando el nuevo sistema, y no un sistema que tal vez ya existe y que queremos reemplazar. En realidad es cierto que nos estamos adelantando un poco al diseño. Sin embargo, esto es en la gran mayoría de los casos algo inevitable, e incluso positivo, ya que es muy difícil hablar siempre “en abstracto” con los usuarios. En algún momento surge la necesidad de pensar cómo, a grandes rasgos, se podrá implementar ese sistema. Lo que el analista no debe hacer es condicionar innecesariamente las decisiones de diseño que puedan ser postergadas, como por ejemplo hablar de lenguajes de programación, modelos de máquinas, sistemas operativos, motores de bases de datos, etc.

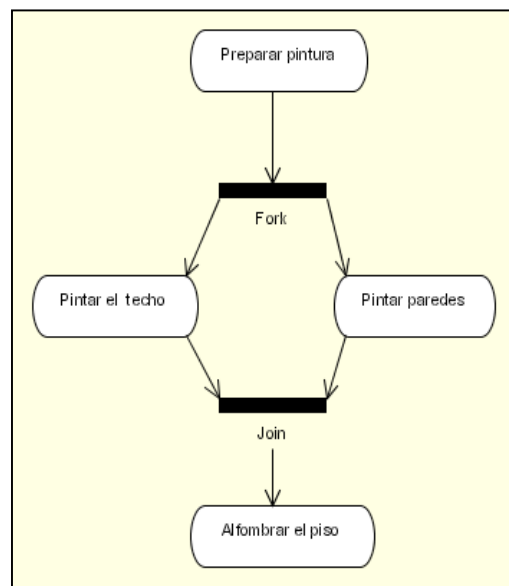


Los andariveles se indican con líneas verticales que delimitan zonas (como “columnas”) en las cuales se colocan los elementos que son responsabilidad de quien se indique en ese andarivel.

### Transiciones Concurrentes (Fork y Join)

Una transición concurrente puede tener muchas acciones origen y muchas acciones destino. Representa una sincronización y/o bifurcación de control en ejecuciones concurrentes sin subacciones concurrentes. Una transición concurrente se habilita cuando todas las acciones origen han finalizado. Cuando ocurre una transición concurrente, todas las acciones destino son disparadas en forma concurrente (salvo que tengamos condiciones de guarda, aunque esto no lo veremos en la materia).

Este tipo de transición es una de las principales diferencias entre los diagramas de actividad y los flujogramas tradicionales.



Los fork se indican con una barra horizontal corta desde la cual salen las diferentes transiciones (siempre más de una).

Los join se indican con una barra horizontal corta (idéntica a la del fork) hacia la cual llegan las diferentes transiciones (siempre más de una).

## Ejemplo

“... de esta manera las órdenes ingresadas por los vendedores, son autorizadas por el jefe de ventas. Una vez autorizada, el depósito es el responsable de preparar la entrega. Cuando la orden se encuentre paga, el cliente se encontrará en condiciones de recibir su pedido...”

