

# CAPÍTULO 9. DIAGRAMAS DE ACTIVIDAD

## 1. Introducción

Los diagramas de actividad son uno de los diagramas UML que muestran el comportamiento dinámico del sistema. Esencialmente, consisten en un diagrama de flujo en el que se muestran los pasos que deben ejecutarse para cumplir un proceso de cómputo, pudiendo también incluir aspectos de sincronización.

Los diagramas de actividad con un caso especial de las máquinas de estados. Si con éstas representábamos el comportamiento de un solo elemento, con los diagramas de actividad representamos procesos en los que (1) interviene más de un clasificador y (2) consideramos necesario representar cómo colaboran esos clasificadores entre sí.

La siguiente figura muestra la sintaxis abstracta de los diagramas de actividad. Como se observa, muchos de los elementos se incluían ya en la Figura 108 (página 115), en la que se describían las máquinas de estados. El contorno de los nuevos elementos se marca en color rojo y, para los que lo vean en blanco y negro, también en trazo más grueso.

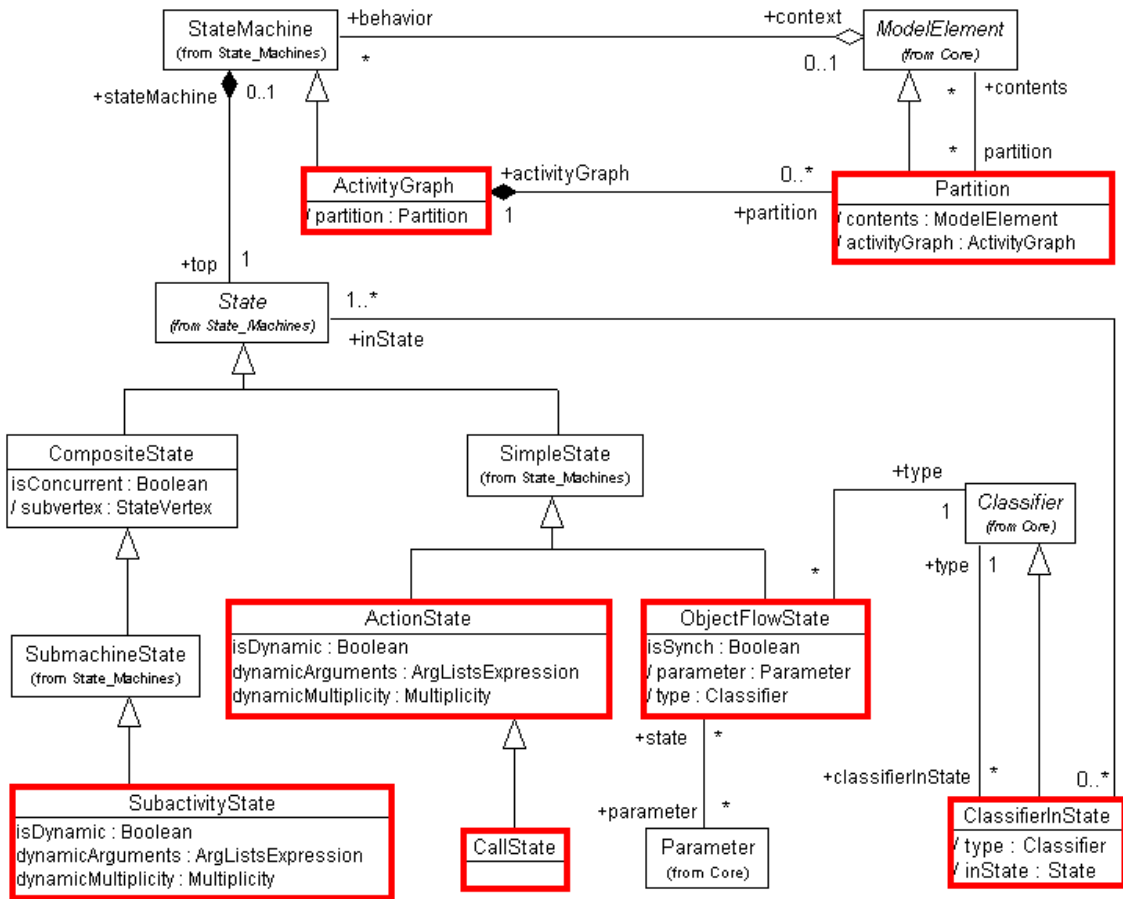


Figura 121. Sintaxis abstracta de los diagramas de actividad

## 2. Elementos de los diagramas de actividad

A continuación se describen los elementos nuevos que aparecen en la Figura 121. Complétense estas descripciones con las de la tabla Tabla 2 (página 119 y anteriores), en el capítulo anterior.

Elemento	Descripción	Asociaciones y otros atributos
Diagrama de actividad ( <i>ActivityGraph</i> )	Tipo especial de máquina de estados que describe un proceso computacional en función del flujo de control y del flujo de objetos entre las actividades que lo forman el proceso descrito.	
Partición ( <i>Partition</i> )	Mecanismo que permite dividir los estados del diagrama en grupos. Los estados de cada partición son los conocidos a través del atributo <i>contents</i> .	
Estado de acción ( <i>ActionState</i> )	Estado que representa un paquete atómico de una o más acciones (o se ejecutan todas las acciones en el paquete o no se ejecuta ninguna). Tiene una o más transiciones de salida “implícitas”: es decir, que se sale del estado de manera automática cuando el paquete de acciones termina de ejecutarse. Si hay más de una transi-	<i>isDynamic</i> : valor booleano que, si vale <i>true</i> , indica que las acciones del estado se pueden ejecutar de forma paralela. Se usa junto a los dos siguientes atributos. <i>dynamicArguments</i> : lista de objetos que determina en tiempo de ejecución el número de ejecuciones paralelas de las acciones del estado. Cada objeto sirve como parámetro para una de las ejecuciones concu-

Elemento	Descripción	Asociaciones y otros atributos
	ción de salida, debe etiquetarse cada una con una guarda. Los estados de acción no pueden tener actividades ( <i>doActivity</i> ) ni acciones de salida ( <i>exit</i> ).	rrrentes. <i>dynamicMultiplicity</i> : número máximo de ejecuciones paralelas de esta actividad.
Objeto de flujo ( <i>ObjectFlowState</i> )	Representa una instancia que es entrada o salida de una acción.	<i>isSynch</i> : valor booleano que, si vale <i>true</i> , indica que el objeto se usa como estado de sincronización. <i>type</i> : tipo de la instancia.
Clasificador con estado ( <i>ClassifierInState</i> )	Representa una instancia que se encuentra en un cierto estado. Se usa cuando en la situación que describe el diagrama se manipula un objeto cambiándolo varias veces de estado.	<i>type</i> : tipo de la instancia <i>inState</i> : estado en el que se encuentra la instancia <i>parameter</i> : parámetros que el objeto suministra como salida o toma como entrada
Estado de llamada ( <i>CallState</i> )	Es un estado de acción ( <i>ActionState</i> ) que llama a una única operación. Es una forma de separar las acciones de un <i>ActionState</i> en varios pequeños estados: esto a veces es interesante para remarcar, por ejemplo, las acciones que producen determinados resultados.	
Estado de subactividad ( <i>SubactivityState</i> )	Estado que representa una secuencia no atómica de ejecución.	<i>isDynamic</i> : valor booleano que, si vale <i>true</i> , indica que las acciones del estado se pueden ejecutar de forma paralela. Se usa junto a los dos siguientes atributos. <i>dynamicArguments</i> : lista de objetos que determina en tiempo de ejecución el número de ejecuciones paralelas de las acciones del estado. Cada objeto sirve como parámetro para una de las ejecuciones concurrentes. <i>dynamicMultiplicity</i> : número máximo de ejecuciones paralelas de esta actividad.

Tabla 3. Descripción de los elementos de los diagramas de actividad

### 3. Ejemplos y notación

A continuación se muestra la descripción de la situación en que se realiza una transferencia con una tarjeta desde un cajero automático: intervienen tres instancias (la tarjeta, su cuenta asociada y la cuenta destino). Los diagramas de actividad permiten la creación de *swimlanes* (literalmente: “calles de piscina”). Cada calle se utiliza para representar los estados por los que pasa cada uno de los objetos que colabora activamente en el proceso que se está describiendo.

La Figura 122 muestra el diagrama de actividad correspondiente dibujado con Rational Rose. Esta herramienta no permite la adición de objetos al diagrama y permite construcciones ilegales, como la adición de acciones de salida a los estados de acción.

El mismo diagrama se muestra en la Figura 123, pero en esta ocasión con la herramienta Poseidon CE. En este caso, sólo existen estados de actividad y no de acción, y no es posible añadir *swimlanes*, aunque sí objetos.

La situación es bastante parecida con otras herramientas de análisis y diseño orientado a objetos (Fujaba, JDeveloper, Enterprise Architect...): todas dejan construir diagramas de actividad, pero ninguno de forma completa y fiel a la sintaxis abstracta.

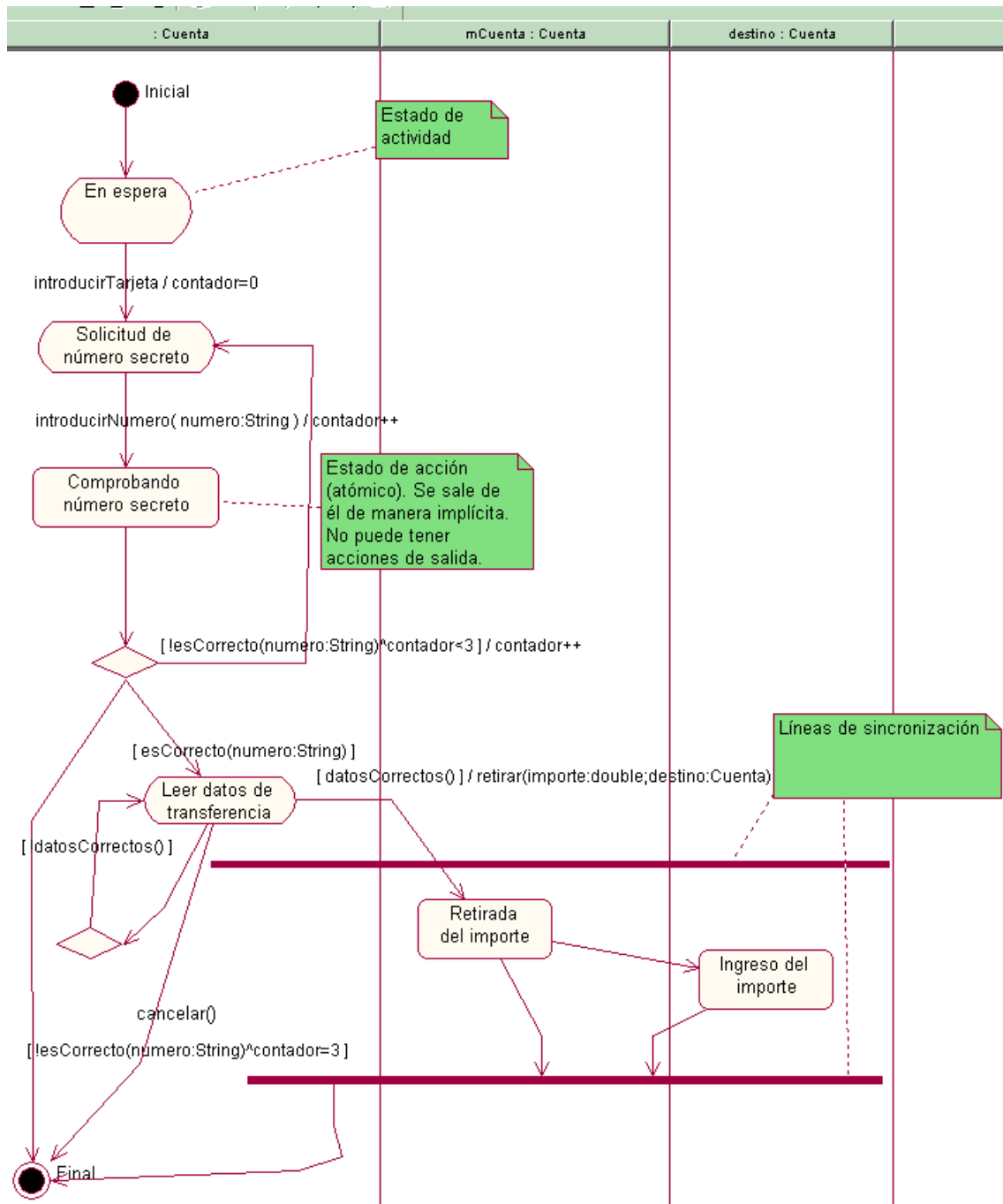


Figura 122. Un diagrama de actividad hecho con Rational Rose

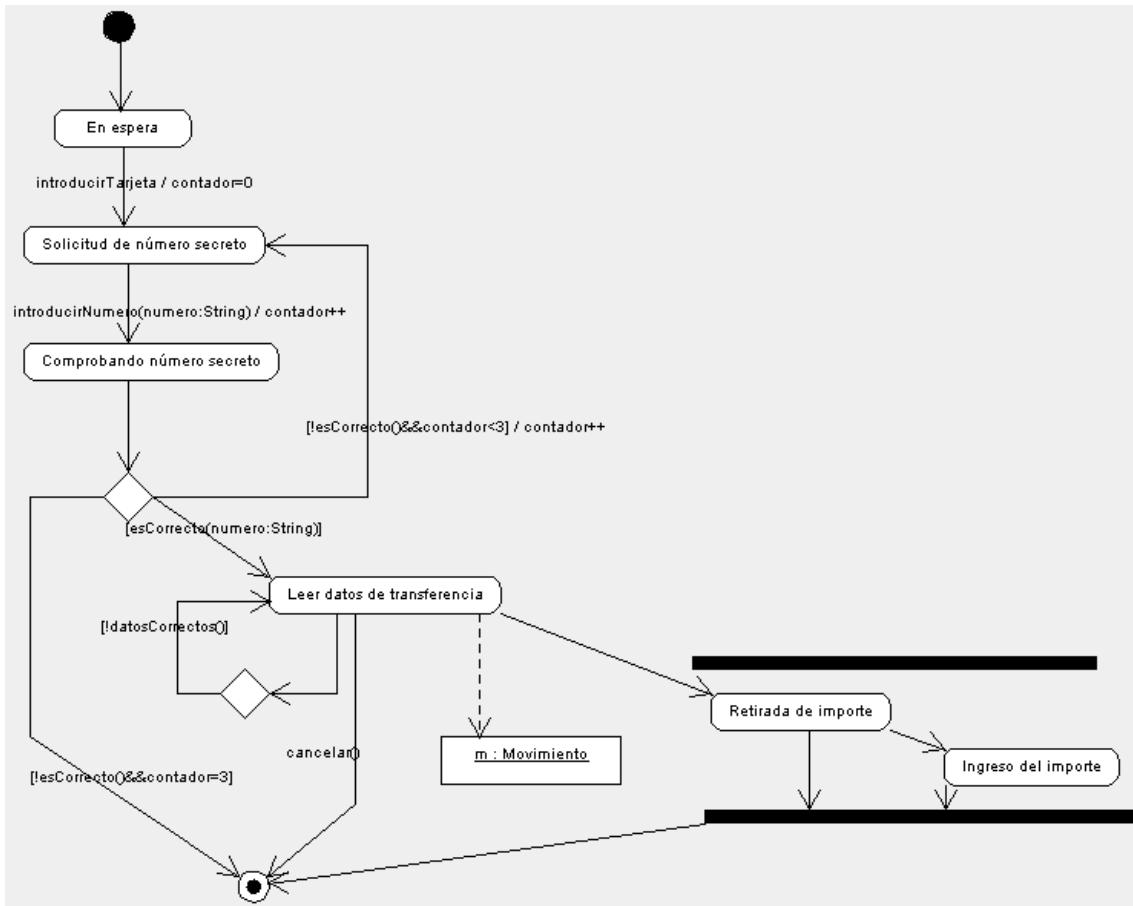


Figura 123. Un diagrama de actividad hecho con Poseidon CE