



INGENIERÍA DEL SOFTWARE I

Práctica 5

Modelado de Diseño

Univ. Cantabria – Fac. de Ciencias

Patricia López



Introducción al Diseño

- **Modelamos la estructura software del sistema** (incluida la arquitectura) para que soporte todos los requisitos (incluidos requisitos no funcionales y otras restricciones).
- Se **optimiza** el modelo de análisis
 - Se define “como” va a ser implementado el modelo de análisis.
 - Se tienen en cuenta aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, SO, distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de almacenamiento de datos, etc ...
- Representa el punto de partida para la implementación:
 - Los programadores deben poder implementar el diseño sin que surjan dudas esenciales.
- Un diseño es una solución particular de las muchas posibles
 - Un mismo modelo de análisis es común a todos sus posibles diseños



Clases de Diseño

- Son clases completas, en las que sus estructuras de datos internas quedan perfectamente definidas, así como los diagramas de estados y actividades que especifican las operaciones y métodos de sus interfaces.
- Una clase de diseño debe incluir:
 - **Atributos**
 - Visibilidad
 - Tipo
 - Multiplicidad
 - **Métodos:**
 - Visibilidad
 - Parámetros de entrada y salida con tipo y multiplicidad
 - Flujo de ejecución (diagramas de actividad) o incluso pseudocódigo (campo Semantics de la especificación de la operación)
- En el modelo de diseño se debe realizar también la **identificación de clases activas:** aquellas que necesiten su propio flujo de control dentro del sistema.

Orden
-FechaSolicitud : Date
-fechaEntrega : Date
-numeroOrden : int
-impuestos : float
-precioTotal : float
#calcularImpuesto(pais, parameter) : float
#calcularTotal() : float
+getImpuesto() : float
+setNumeroOrden(nOrden : int)
+getNumeroOrden() : int



Diseño de la Arquitectura

- **Identificación de subsistemas y sus interfaces**

- Permiten **organizar** el modelo de diseño en piezas manejables.
- Se identifican a partir de los **paquetes de análisis**, pero puede ser necesario un refinamiento para tratar temas relativos al diseño, implementación y distribución del sistema:
 - Una **parte** de un anterior **paquete de análisis**, **compartida y utilizada** por **varios subsistemas**
 - Algunas **partes** de un anterior **paquete de análisis se realizan** por **productos SW reutilizados** (asignables a capas intermedias o subsistemas de SW del sistema)
 - Puede que la descomposición de subsistemas tenga que tratar los aspectos de distribución, descomponiéndolos en subsistemas más pequeños de forma que cada uno pueda asignarse a un nodo determinado.
- En algunos casos, **no todos los subsistemas se implementan** ya que algunos representan **productos reutilizados** y otros son recursos ya **existentes**. Su inclusión en el modelo de diseño permite analizar y evaluar las alternativas de reutilización.



Diagramas de Estado con VP

Crear Diagrama

The screenshot illustrates the steps to create a State Machine Diagram in a software application. The main window shows the 'File' menu with 'New Diagram' selected. A sub-menu is open, showing 'UML Diagrams' as the selected category. Below this, a list of diagram types is shown: 'Use Case Diagram', 'Class Diagram', 'Sequence Diagram', 'Communication Diagram', and 'State Machine Diagram'. The 'State Machine Diagram' option is highlighted. A 'New Diagram' dialog box is also visible, showing the 'Diagram type' section with 'UML Diagrams' selected. The 'State Machine Diagram' option is selected in this dialog. The 'New Diagrams' sub-menu is also visible, showing options for 'New Use Case Diagram', 'New Class Diagram', 'New Sequence Diagram', 'New Communication Diagram', and 'New State Machine Diagram'. The 'New State Machine Diagram' option is highlighted.



Diagramas de Estado con VP: Elementos

Estado

Crear estado

Estados Inicial y Final

Especificación de un Estado

Open Specification

Open Specification... Introduzca

Especificación de la actividad asociada

Name: State

Entry: Edit... Remove

Exit: Edit... Remove

Do activity: Edit... Remove

Reset OK Cancel Apply Help

Estados Inicial y Final

Activity Specification (Entry)

Name: Activity

Language:

Precondition:

Postcondition:

Body:

Documentation:

Reset OK Cancel Apply Help

Acciones de Estado de entrada/salida

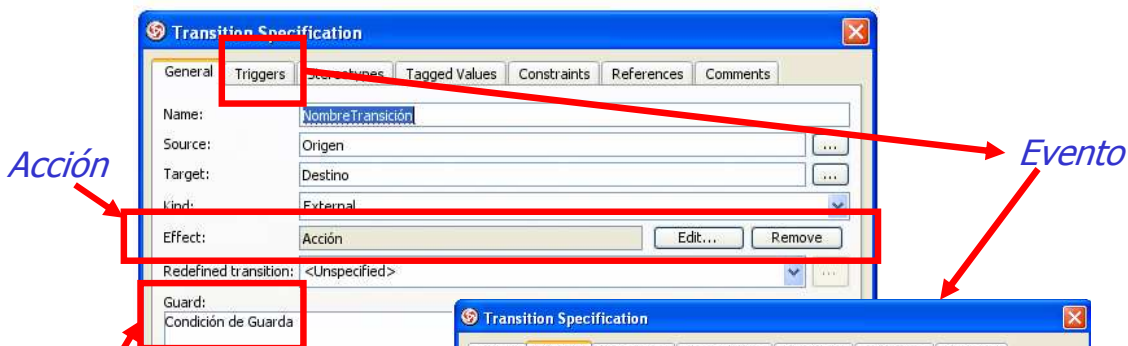
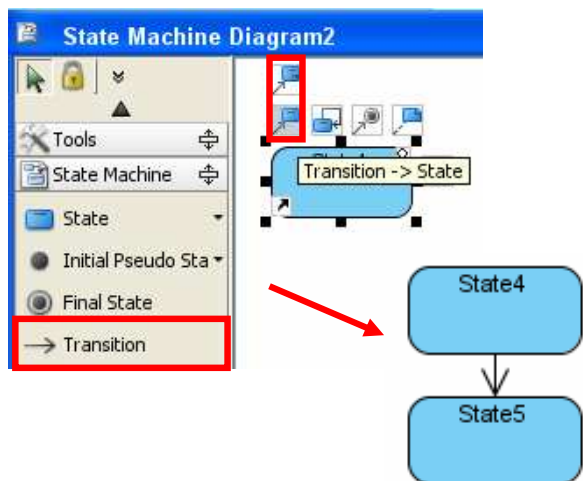
Acciones que se ejecutan independientemente de los eventos que lleguen a los estados

RegistrarseEnElSistema
entry / HacerAlgoAlEntrar
do / HacerAlgoDuranteLaActividad
exit / HacerAlgoAlSalir

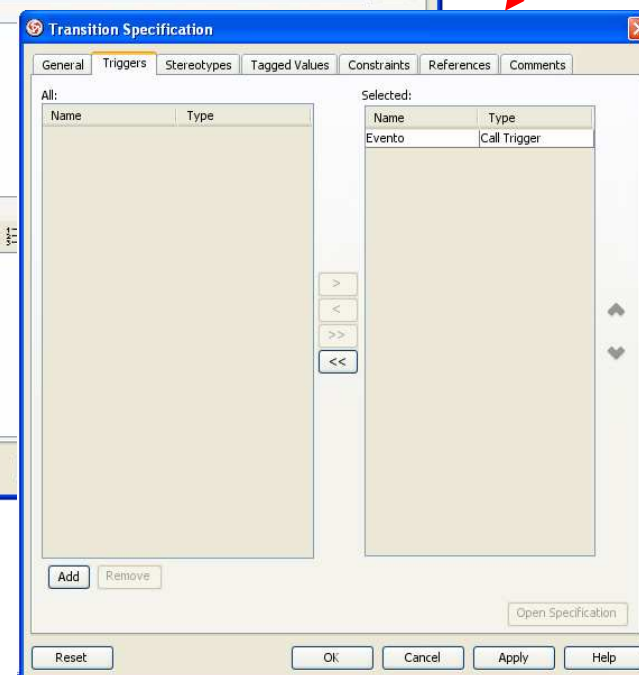


Diagramas de Estado con VP: Elementos

Transiciones



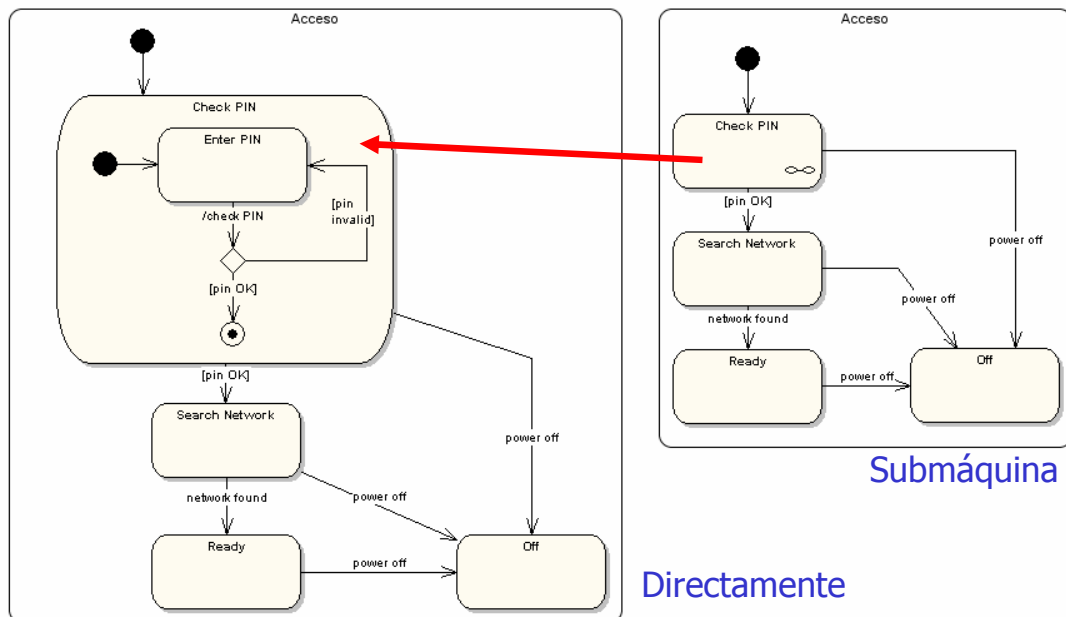
Transiciones Recursivas





Diagramas de Estado con VP: Elementos

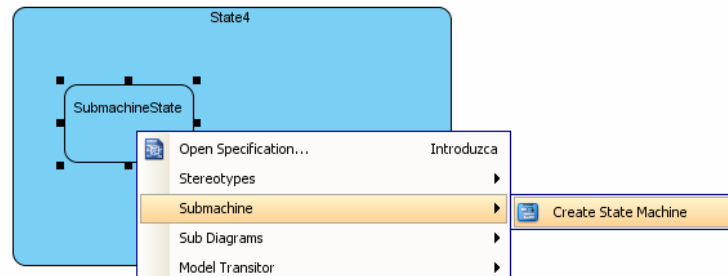
Estados Compuestos: Subestados



Un diagrama de máquina de estado puede incluir diagramas de submáquinas

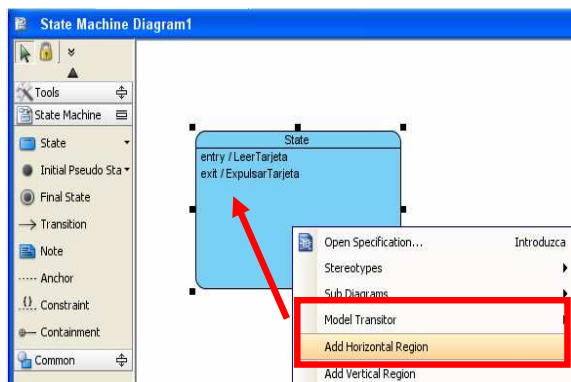
- Mediante estados de Submáquina: los detalles se muestran en un diagrama separado
- Directamente

Mediante Submáquina creando diagrama

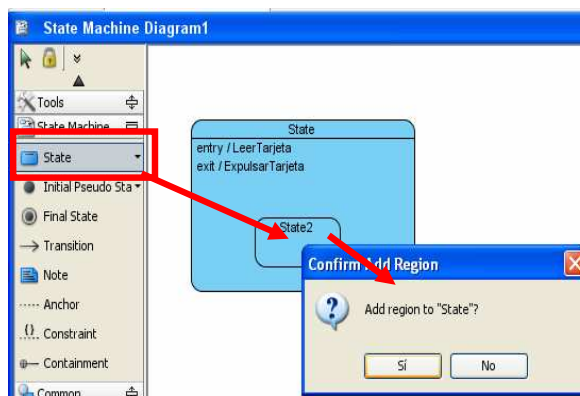


Directamente

Añadimos región y luego elementos



Añadimos un estado y una región



Subestado Concurrente

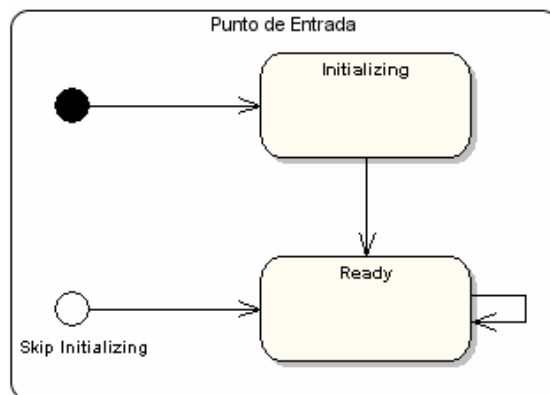
Un estado se puede dividir en regiones conteniendo subestados que existen y se ejecutan concurrentemente.



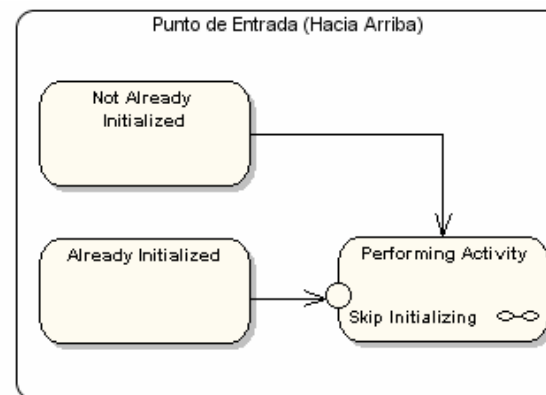
Diagramas de Estado con VP: Elementos

Puntos de Entrada

Algunas veces no deseará ingresar una submáquina en un Estado Inicial normal. Por ejemplo, en la siguiente submáquina sería normal comenzar en el estado inicial, pero si por alguna razón no fuera necesario realizar la inicialización, sería posible comenzar en el estado Ready realizando una transición al punto de entrada nombrado

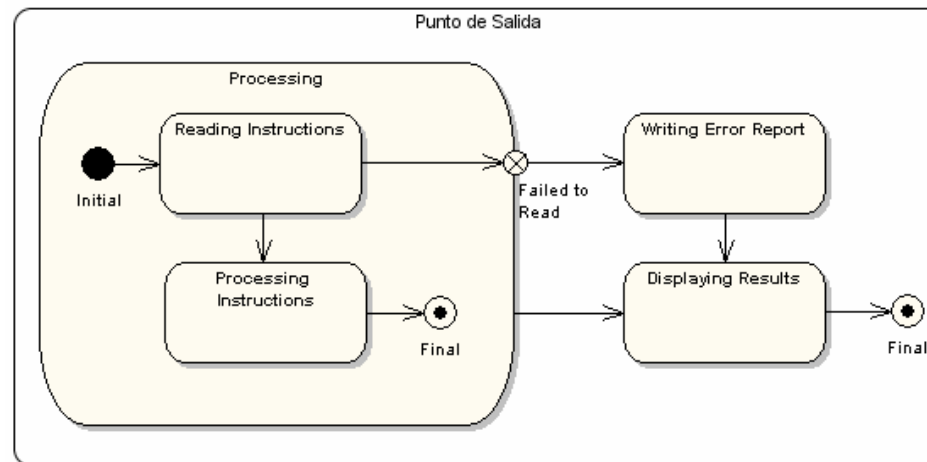


Puntos de Entrada y Salida



Puntos de Salida

Es posible nombrar Puntos de Salida distintos del normal. El siguiente diagrama provee un ejemplo donde el estado ejecutado después del estado de procesos principal depende de que ruta se use para realizar la transición del estado



- State Machine
- Submachine State
- Initial Pseudo State
- Shallow History
- Deep History
- Choice
- Junction
- Fork
- Join
- Entry Point
- Exit Point
- Terminate

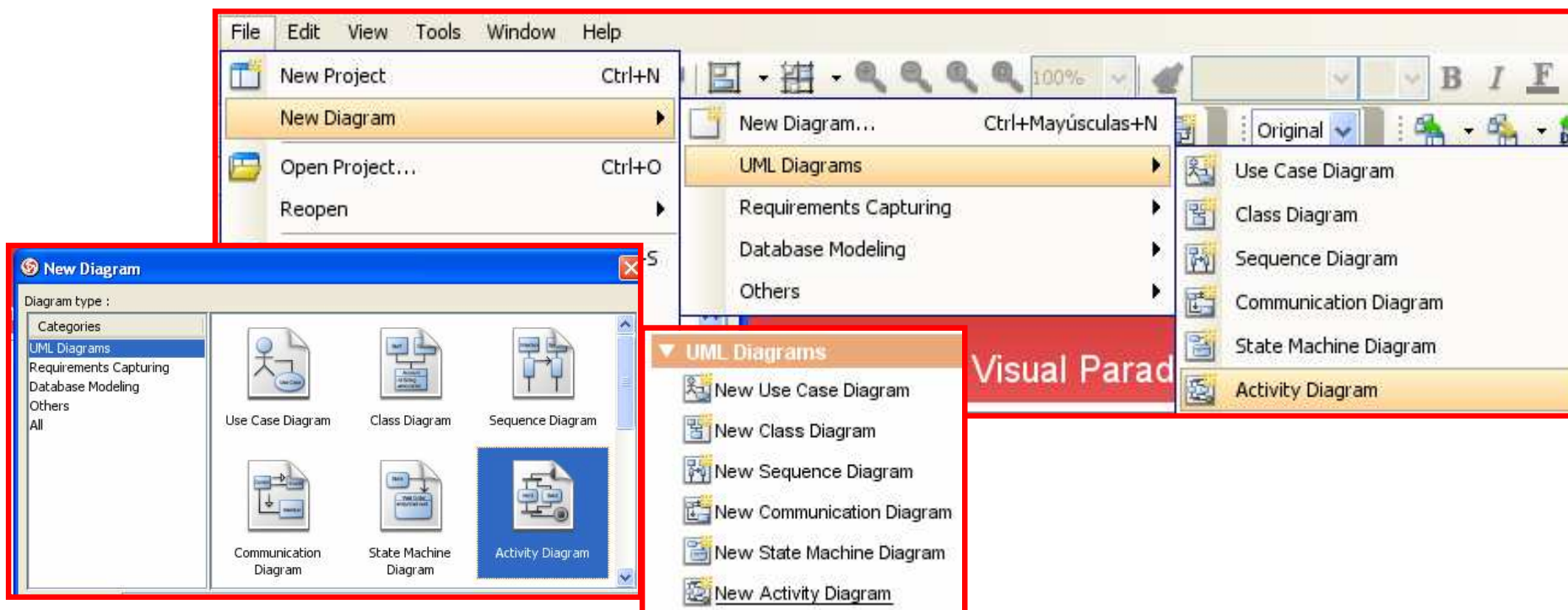
Tipos de Estados

Puntos de Entrada y Salida



Diagramas de Actividad con VP

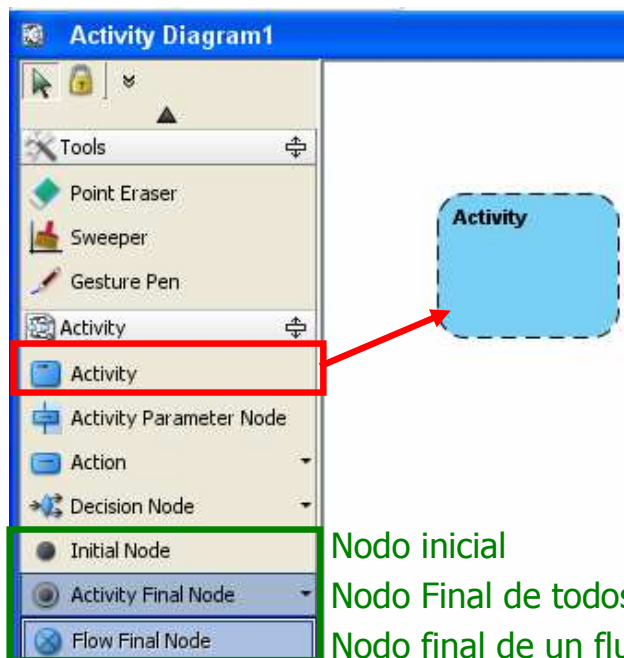
Crear Diagrama



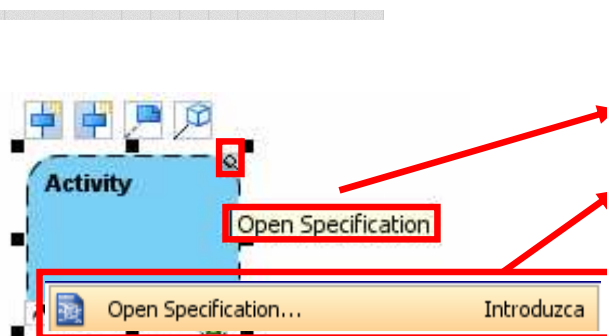


Diagramas de Actividad con VP: Elementos

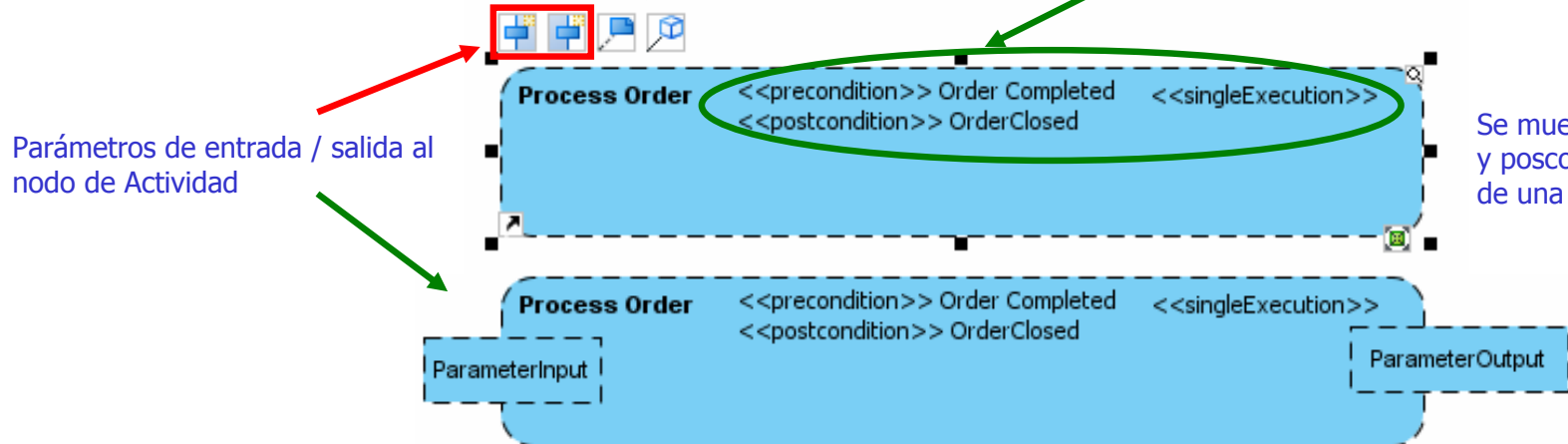
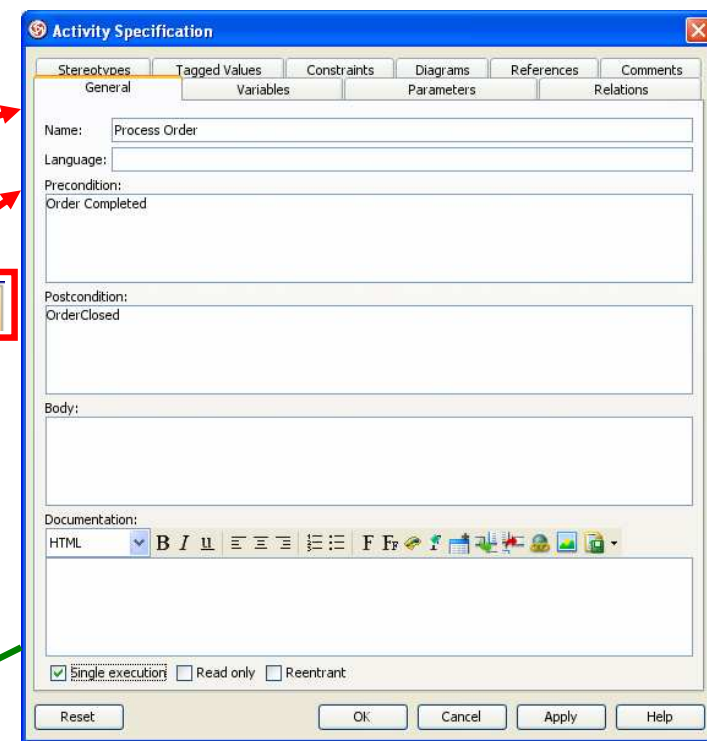
Actividades



Nodo inicial
 Nodo Final de todos los flujos de la actividad
 Nodo final de un flujo de la actividad

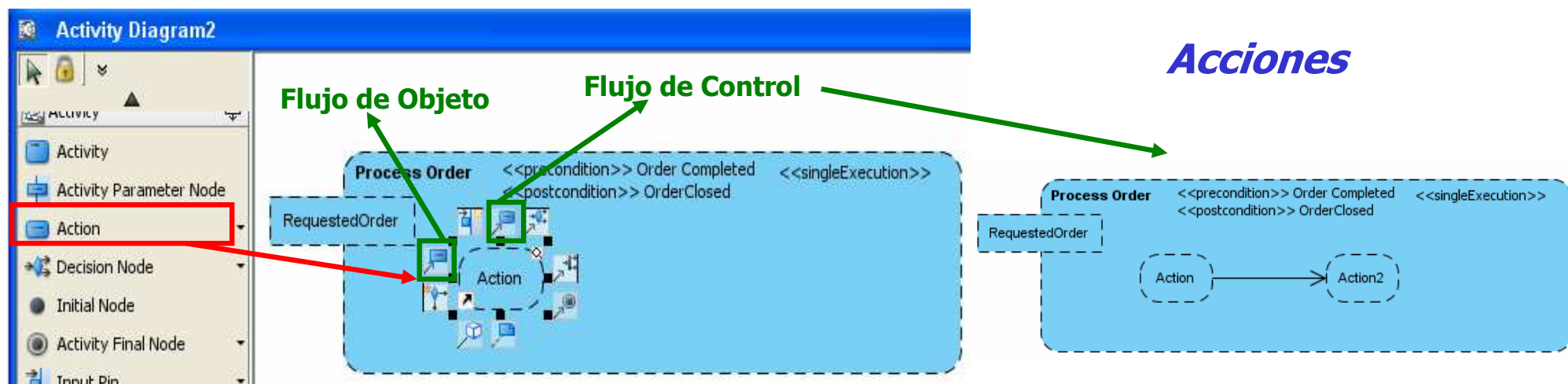


Especificación de Actividades

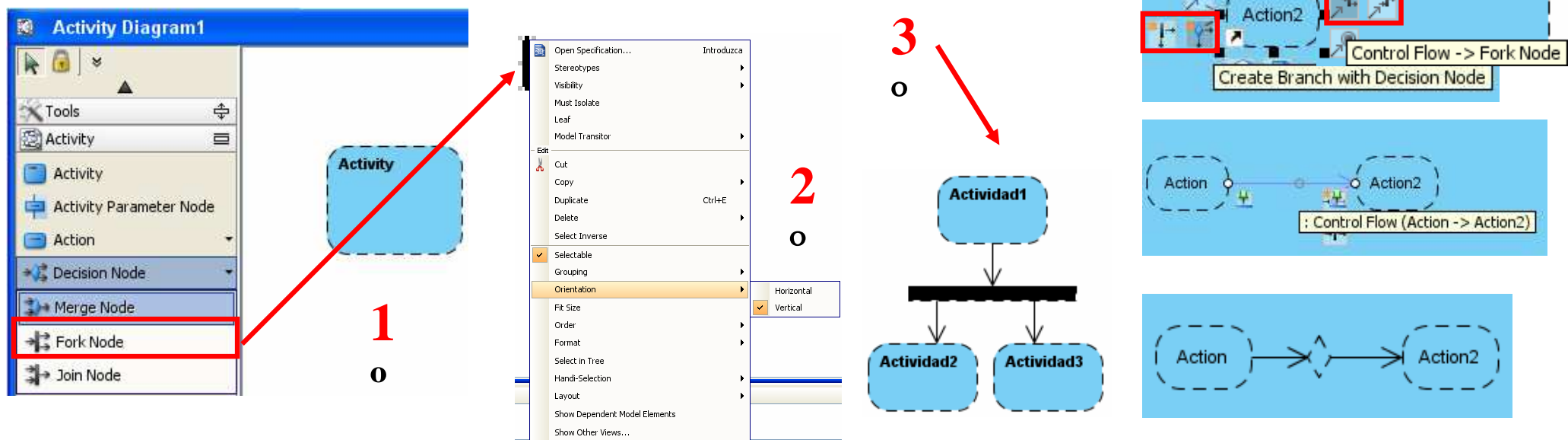




Diagramas de Actividad con VP: Elementos

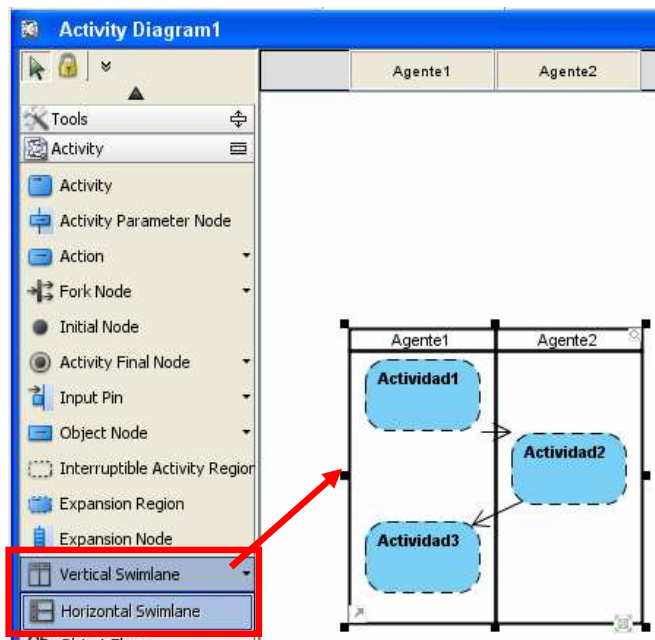


Nodos de Unión/Bifurcación





Diagramas de Actividad con VP: Elementos



Swimlanes

*Flujo de Control
y de Objetos*

