

Introducción a UML 2.0 y Herramientas de Modelado

Código del Curso: CY450

Versión 5.0

Introducción a UML 2.0 y Herramientas de Modelado

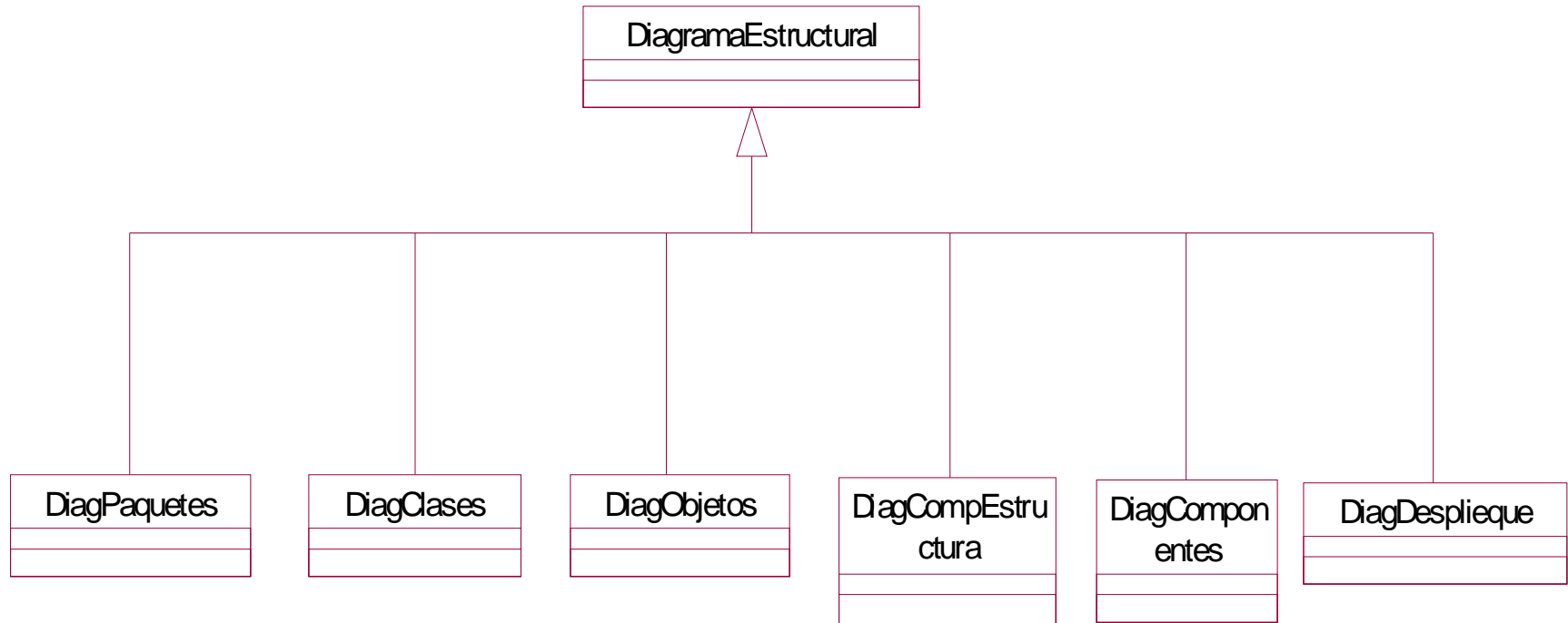
Introducción a UML 2.0 (Diagramas Estructurales)

Objetivos de Aprendizaje

Al finalizar esta unidad ud. debería ser capaz de:

- Explicar los cambios básicos sufridos por la especificación UML 2.0.
- Explicar de forma básica los diagramas de UML 2.0 actualizados en el aspecto estructural.
- Explicar de forma básica los nuevos diagramas incluidos en la especificación UML 2.0 en el aspecto estructural.
- Enumerar diferencias entre diagramas de la especificación UML 2.0 y la especificación anterior.

Ramas de UML 2.0



Ramas de UML 2.0...1

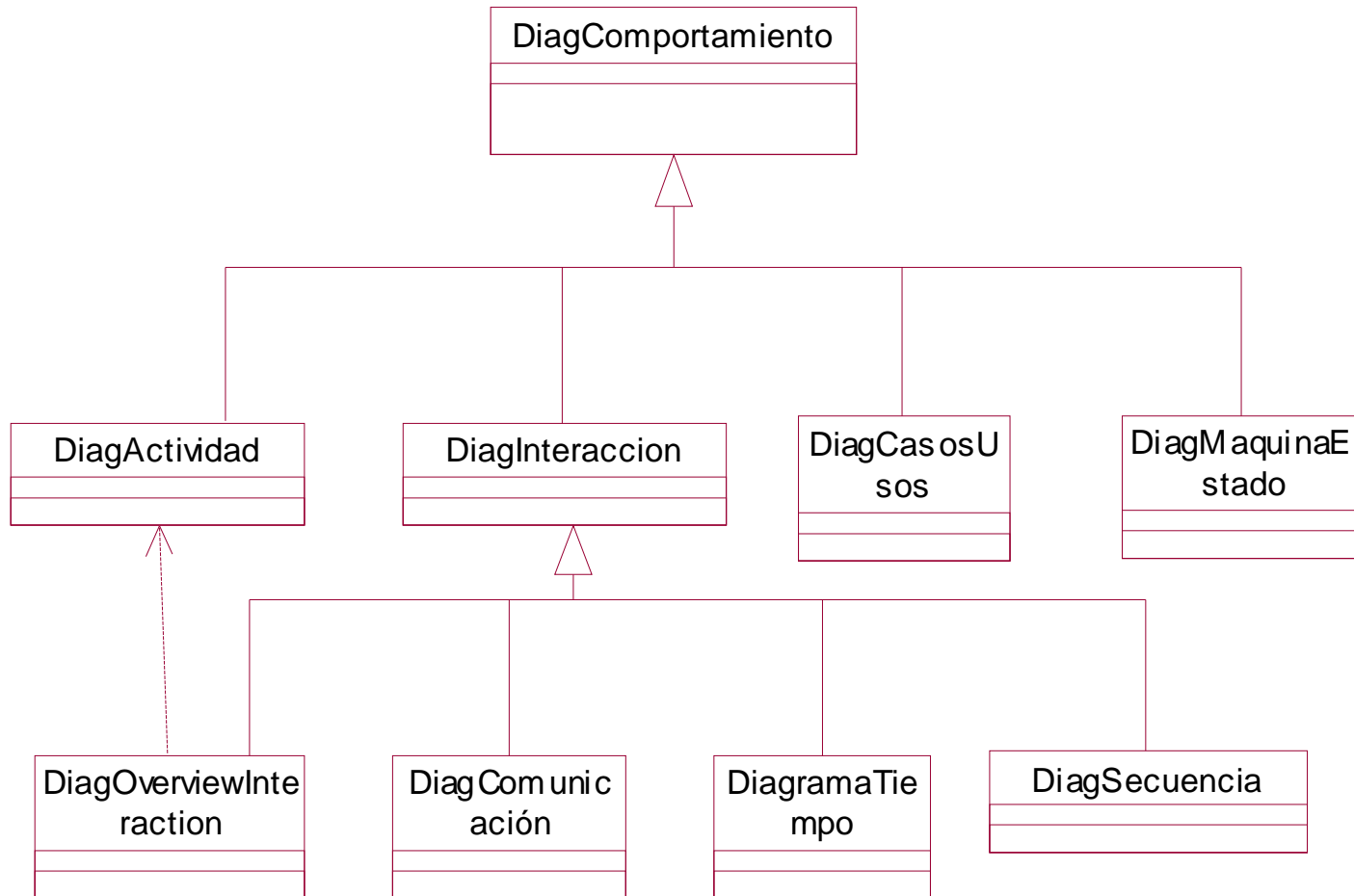


Diagrama de Paquetes

- Proporcionan un espacio de nombres a los elementos asociados.
- Un elemento puede pertenecer a más de un paquete.
- La agrupación de los elementos del modelo no necesariamente debe coincidir con la ubicación física de los elementos del sistema.
- Los paquetes se organizan jerárquicamente, siendo el paquete raíz el que contiene todo el sistema.

Diagrama de Paquetes...1

- **Contenido de Paquetes**

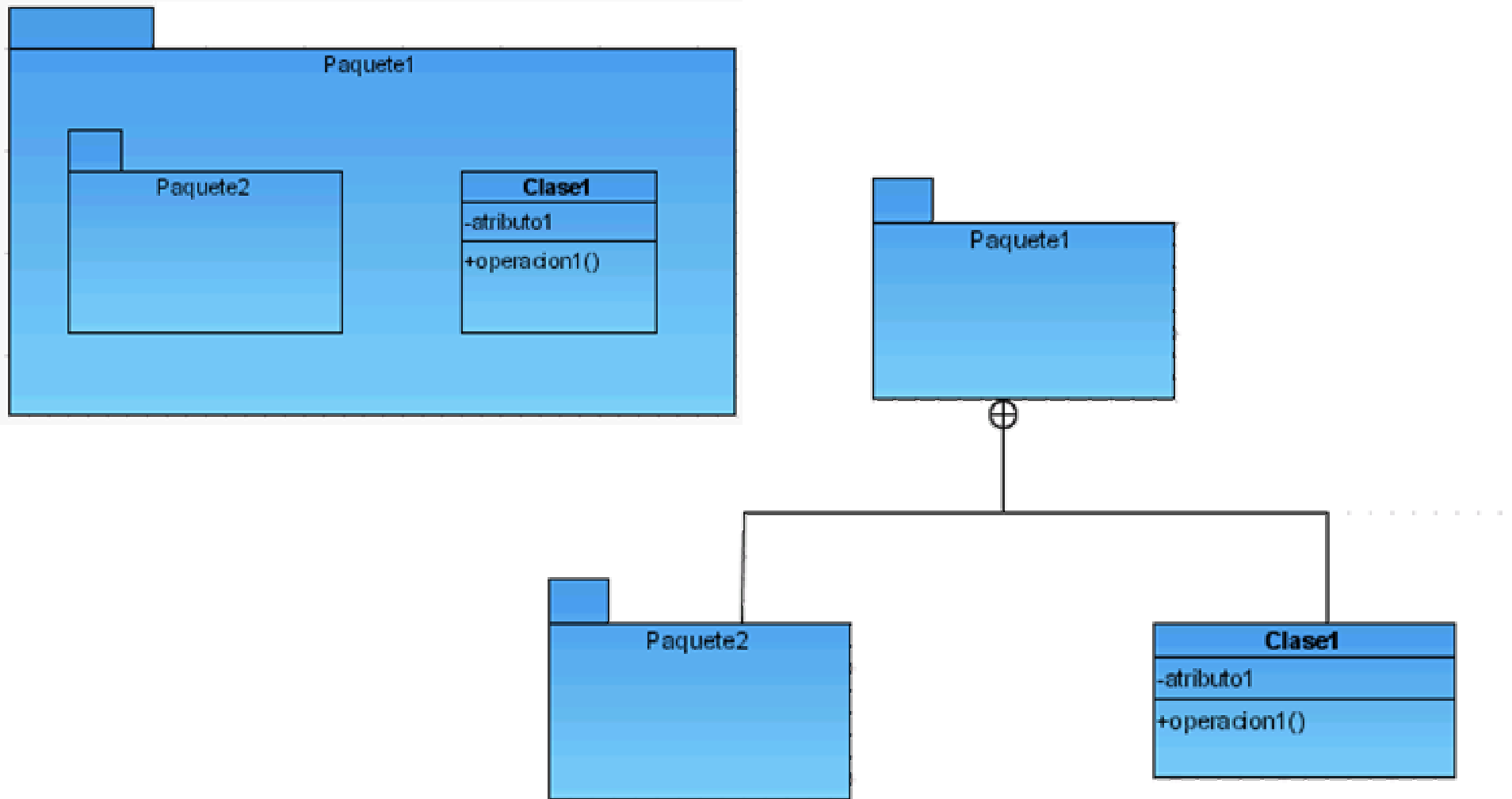


Diagrama de Paquetes...2

- Relaciones de Paquetes

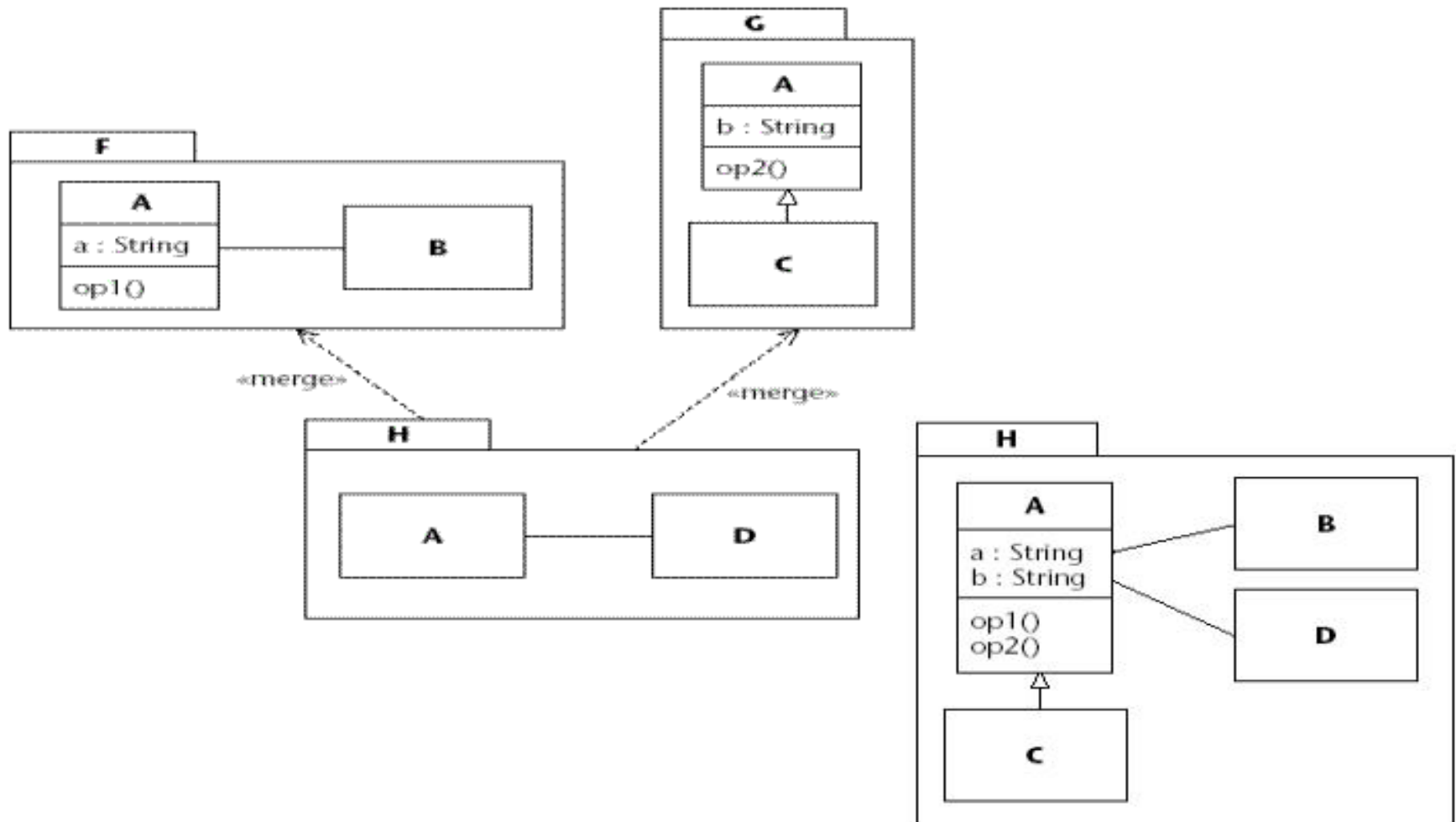


Diagrama de Clases

- Características y Operaciones**

Las características y operaciones son ahora asignadas directamente a la clase no a un clasificador.

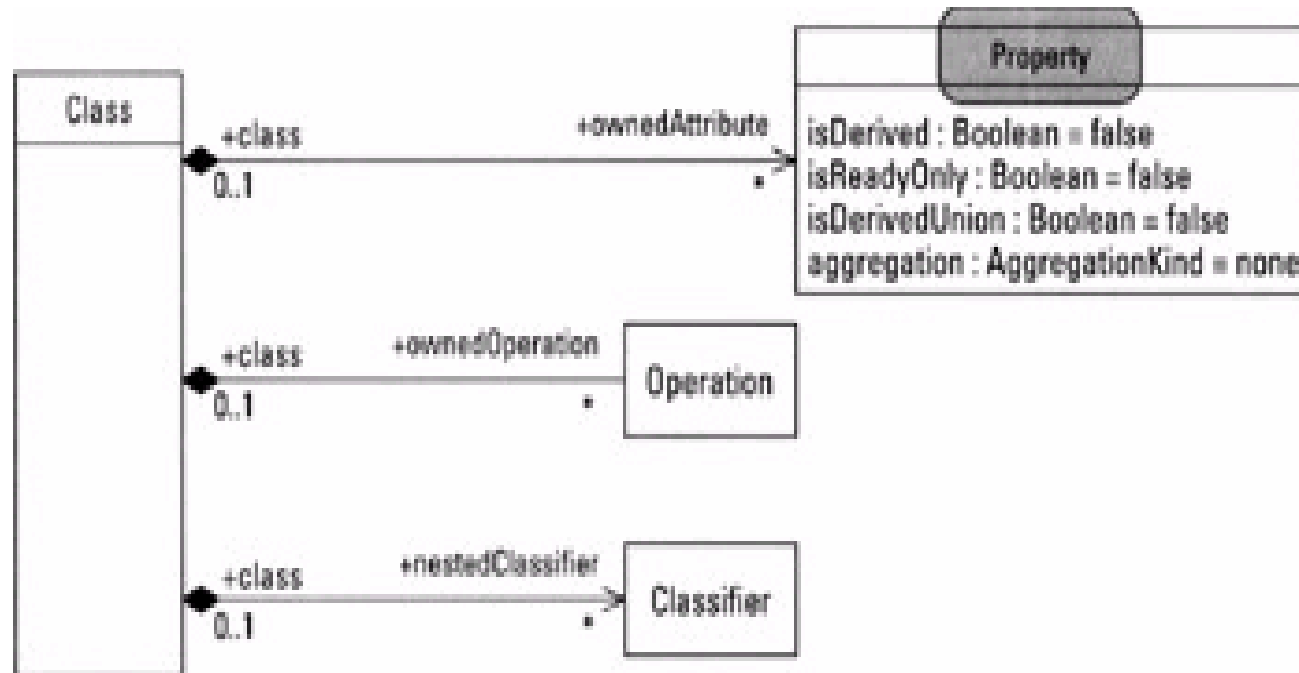


Diagrama de Clases...1

- **Visibilidad**

UML 2.0 incorpora un nuevo tipo de visibilidad llamada visibilidad de paquetes.

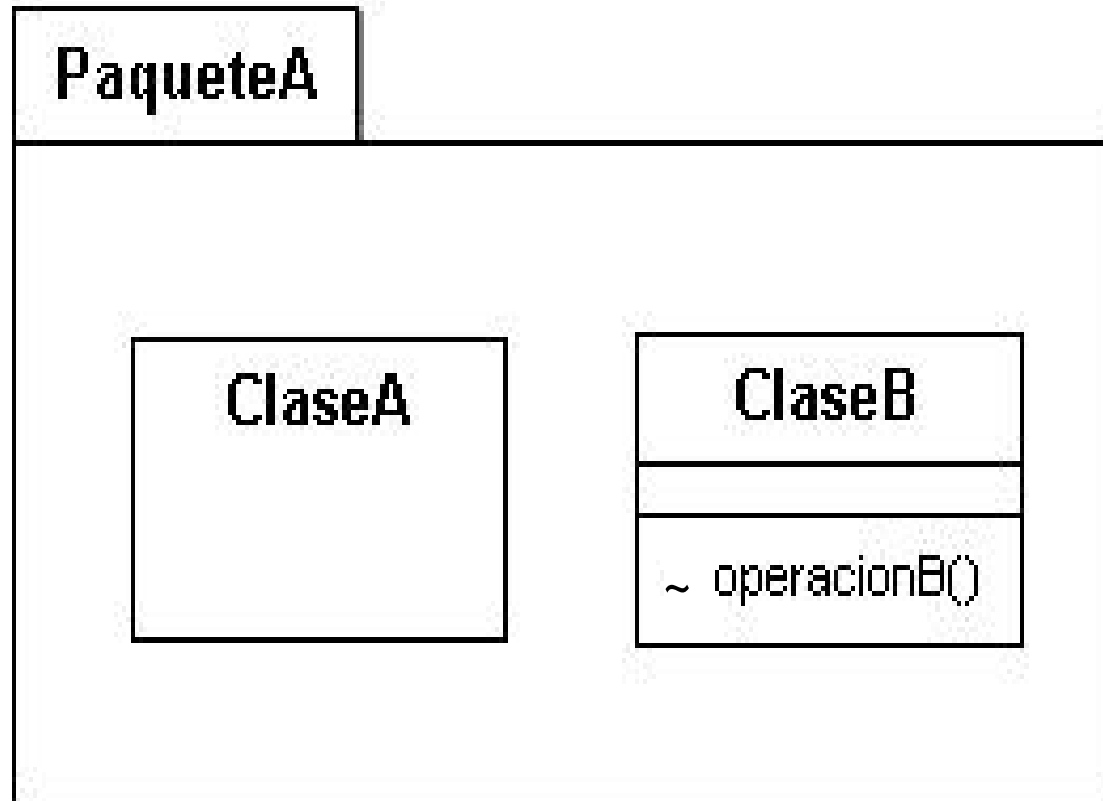


Diagrama de Clases...2

- **Relación de Asociación**

- Las Clases A y B muestran una asociación binaria con ambos extremos navegables.
- Las Clases C y D muestran una asociación binaria donde ningunos de los extremos son navegables.
- Las Clases E y F muestran una asociación binaria donde la navegabilidad no se especifica.
- Las Clases G y H muestran una asociación binaria con uno de los extremos navegables y uno no navegable.
- La Clases I y J muestran una asociación binaria con uno de los extremos navegables y el otro extremo sin especificar.

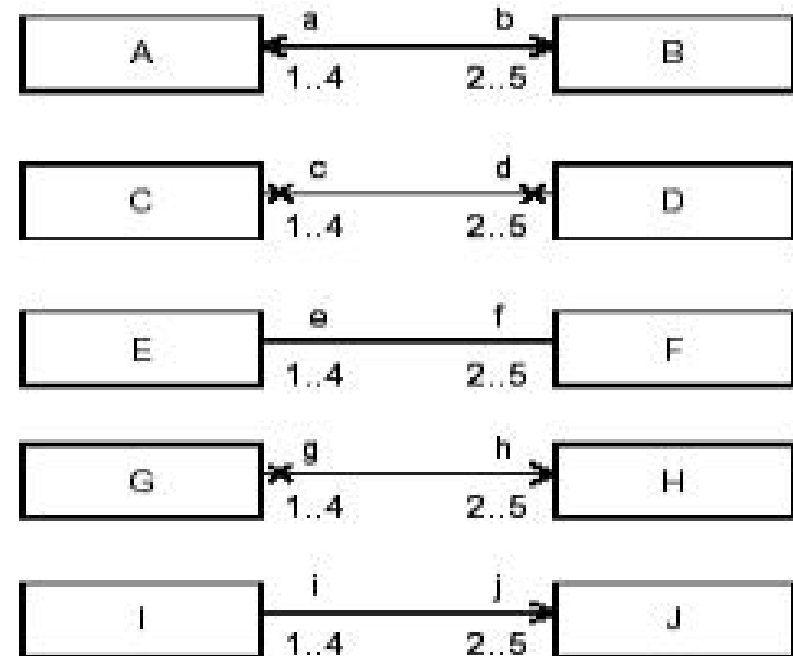


Diagrama de Clases...3

- **Relación de Dependencia**

- Las instancias de una clase pueden sustituir a las instancias de otra clase en tiempo de ejecución.
- La dependencia por sustitución no se basa en la generalización / especialización, más bien es otra manera en que un clasificador podría ser implementado.
- La sustitución no implica la herencia de la estructura de la clase, sino solamente la conformidad de contratos públicos disponibles.
- Se requiere que las interfaces puestas en ejecución por el clasificador del contrato, también sean puestas en ejecución por el clasificador que lo sustituye.

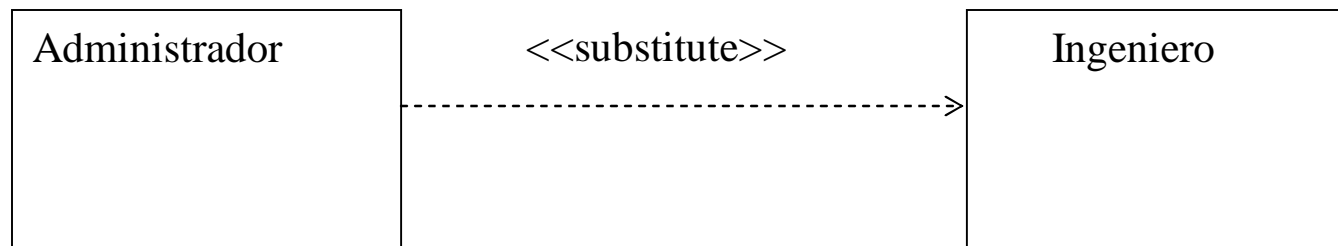
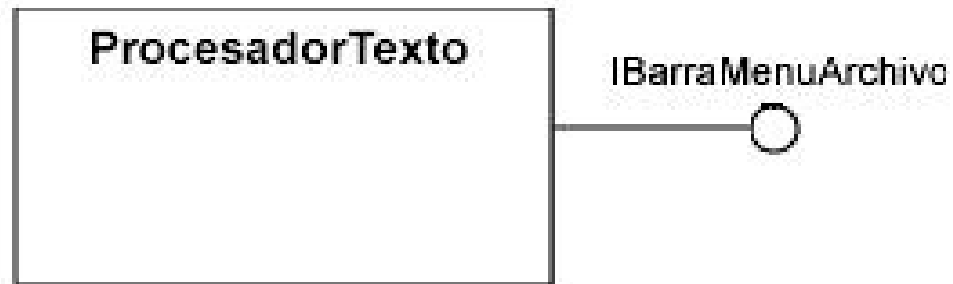


Diagrama de Clases...4 Interfaz

- **Interfaz que ofrece servicios**



- **Interfaz que requiere servicios**

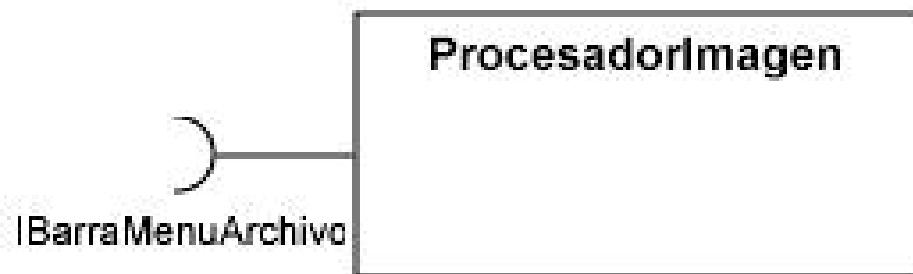


Diagrama de Clases...5 Interfaz

- **Interconexión entre una interfaz que provee servicios y una que requiere servicios**



Diagrama de Composición y Estructura

- Representa la estructura interna de un clasificador.
- Describe las conexiones entre elementos que trabajan juntos en un clasificador (caso de uso, objeto, colaboración, clase, entre otros).
- **Parte**
 - Declara que una instancia de un clasificador puede tener un conjunto de instancias por composición.
 - Si la instancia que contiene las partes es destruida, todas las instancias contenidas también se destruyen.
 - Dentro de una clase, las partes se modelan con un rectángulo.

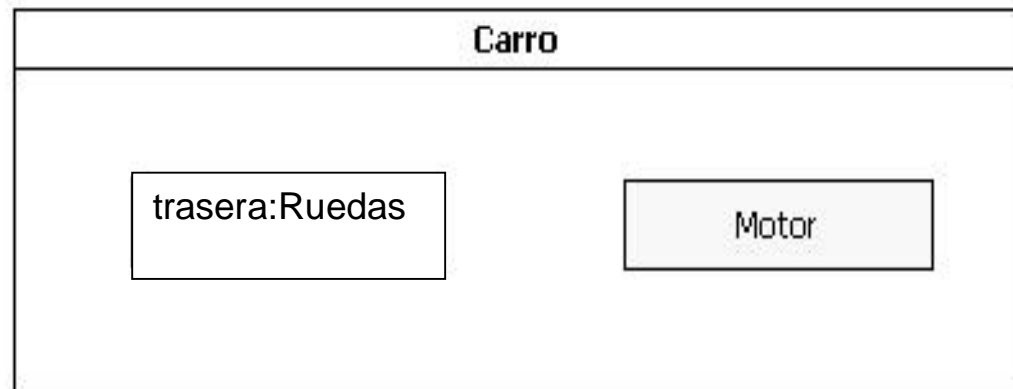


Diagrama de Composición y Estructura...1

- **Conectores**

- Son las líneas que permiten la comunicación entre las instancias de las partes en una estructura de composición.

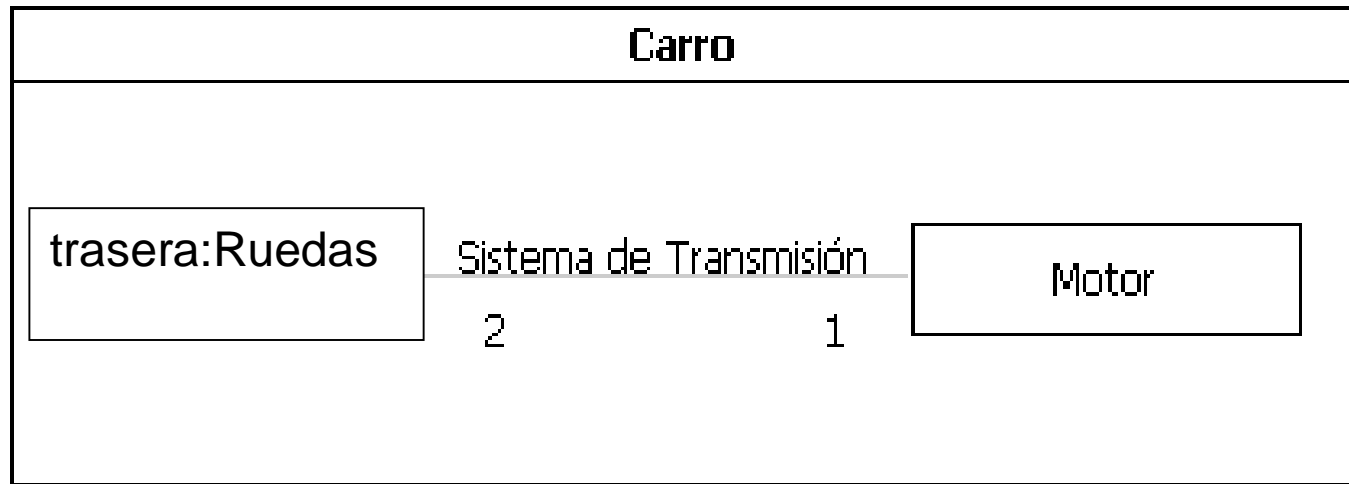


Diagrama de Composición y Estructura...2

- **Puerto**

- Puntos de contactos entre los conectores y las partes.
- Se modelan como un cuadro pequeño en un borde de la parte.
- Definen un punto de interacción entre las partes, que puede ser en forma de interfaz que ofrece servicios o de interfaz que requiere servicios.

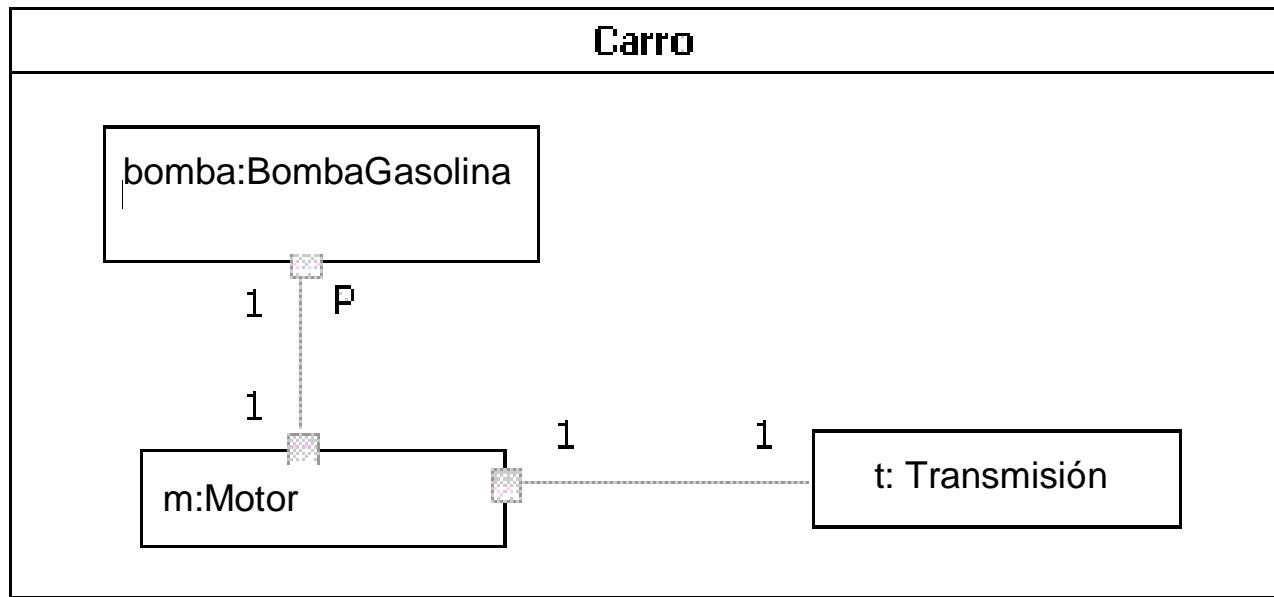


Diagrama de Composición y Estructura...3

- **Partes como Atributos**
 - El diagrama de Composición y estructura permite ver las partes como simples atributos.

Carro
m: Motor bomba: BombaGasolina t: Transmisión

Diagrama de Composición y Estructura...4

- Relación entre Composición en UML 1.X y Partes en UML 2.0

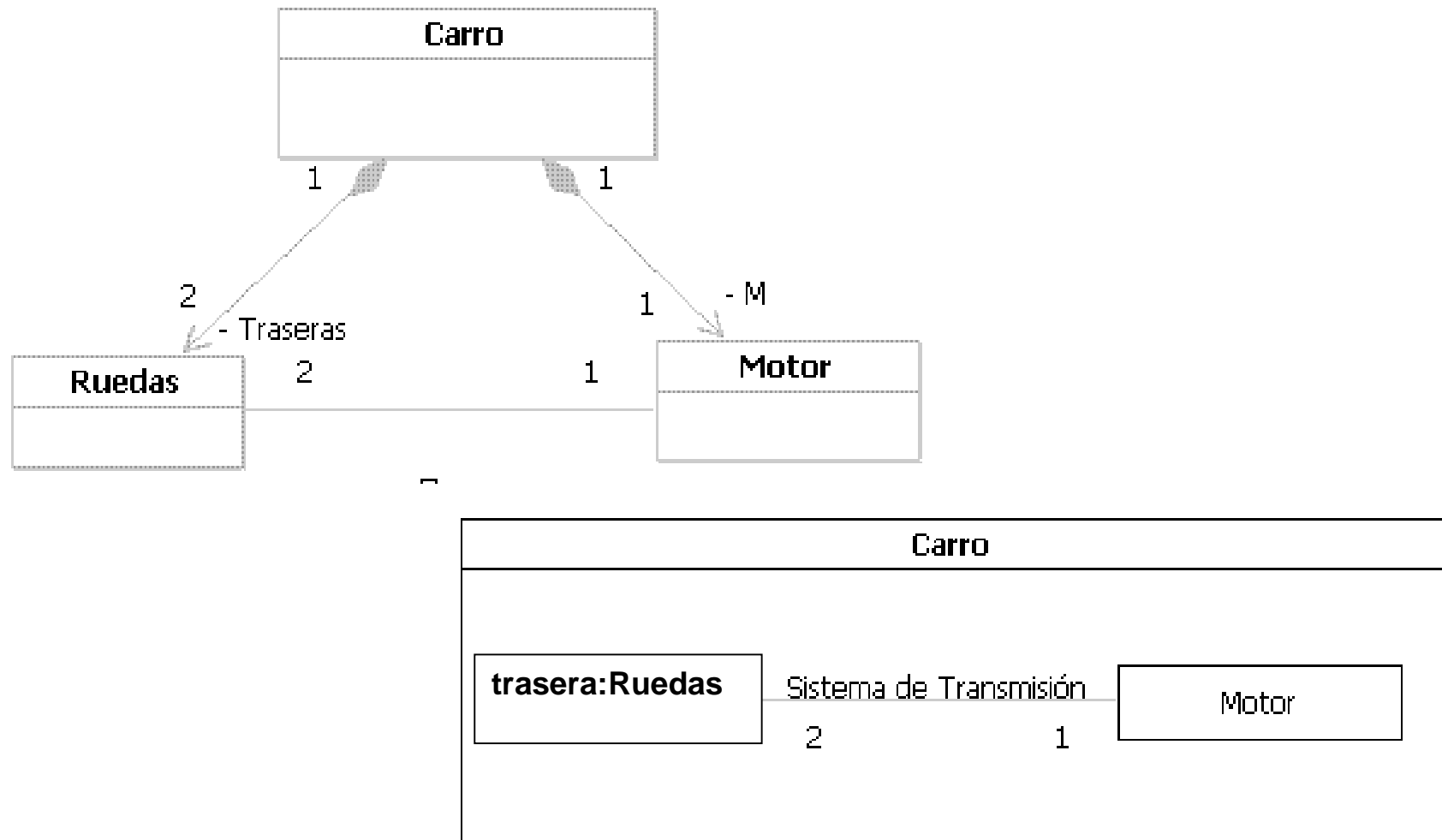


Diagrama de Componentes

- **Componente**

- UML 2.0 simplificó la representación antigua, eliminando los dos rectángulos del borde izquierdo por el estereotipo <<component>> en el centro del rectángulo.
- El símbolo antiguo puede ser utilizado de manera alternativa en la esquina superior derecha del rectángulo.



Diagrama de Componentes...1

- **Interfaces**

- Al igual que las clases, los componentes pueden tener interfaces que ofrecen servicios e interfaces que requieren servicios.

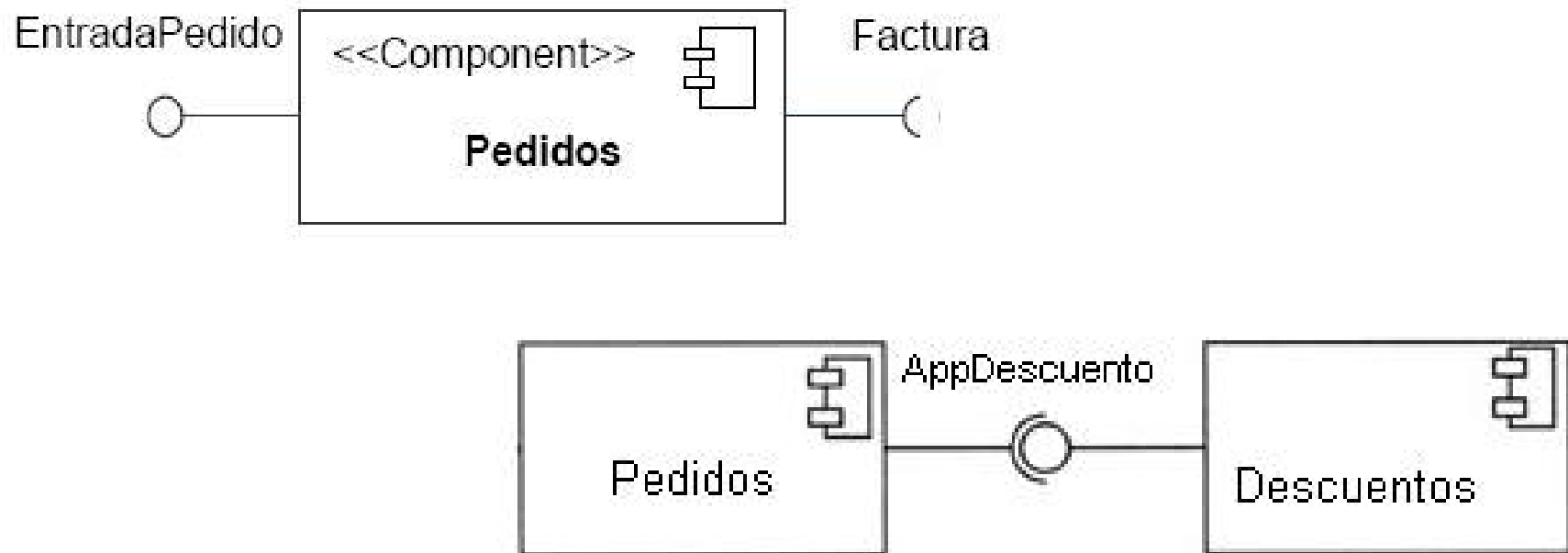


Diagrama de Componentes...2

Vista de un Componente

Vista Caja Negra



Vista Caja Blanca

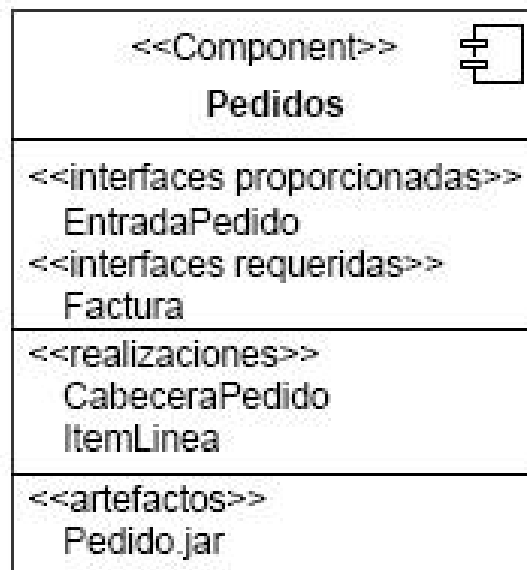
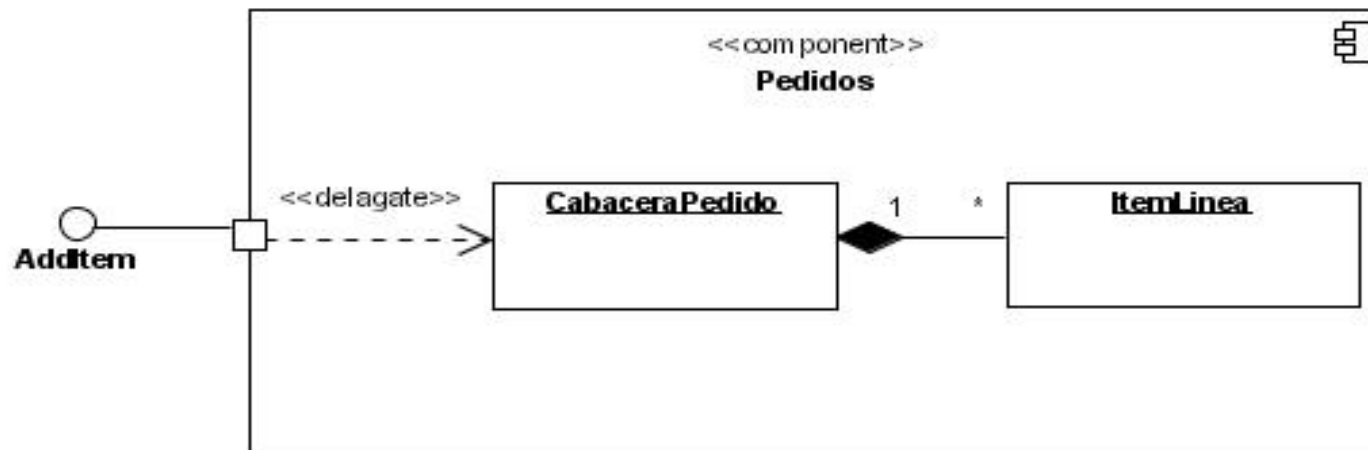


Diagrama de Componentes...3

Conectores

Conector de Delegación



Conector de Ensamblaje



Diagrama de Componentes...4

- **Artefacto**

- Representa una pieza física de información, tal como un modelo, un archivo, una tabla, entre otros; que es usado o producido por un software en el desarrollo del proceso.
- Los artefactos son realizados generalmente por componentes, este tipo de relación se conoce con el estereotipo <<manifest>> y representa una realización física concreta de uno o más elementos de un modelo por un artefacto.



Diagrama de Despliegue

Los diagramas de despliegue describen la topología física del sistema: la estructura de las unidades de hardware y el software que se ejecuta en cada unidad.

- **Nodo**

- Un nodo es una subclase de una clase más no de un clasificador como era en versiones anteriores de UML.
- Pueden anidarse, permitiendo mayor comprensión de los diagramas.

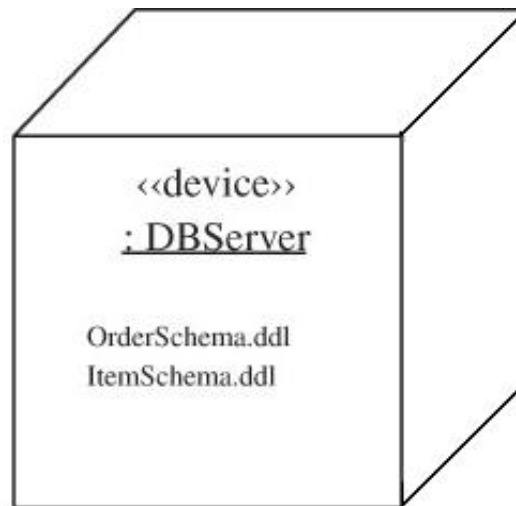


Diagrama de Despliegue...1

- **Tipos de Nodos**

- **Nodo dispositivo:** Es un recurso computacional físico con capacidad de procesamiento, sobre el cual los artefactos pueden ser desplegados para su ejecución.
- **Nodo de ambiente de ejecución:** nodo que define un conjunto de servicios que soportan software.

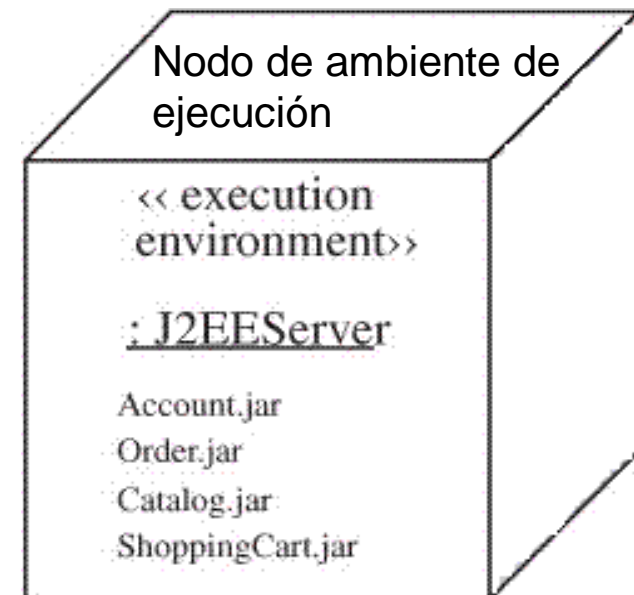
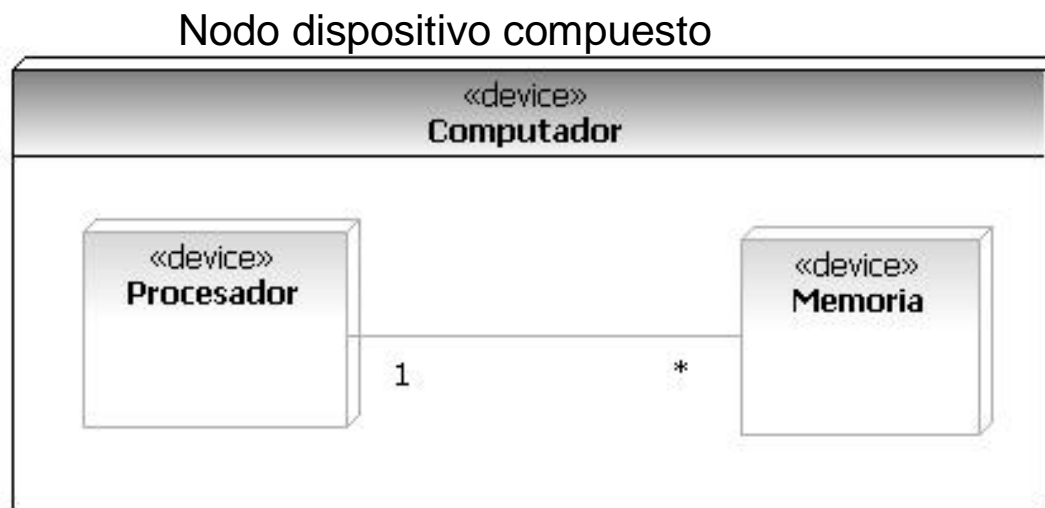


Diagrama de Despliegue...2

- **Atributos y Operaciones en Nodos**
 - Los nodos al ser una subclase de la clase pueden tener atributos y operaciones.
 - Los nodos al igual que las clases también pueden ser instanciados.

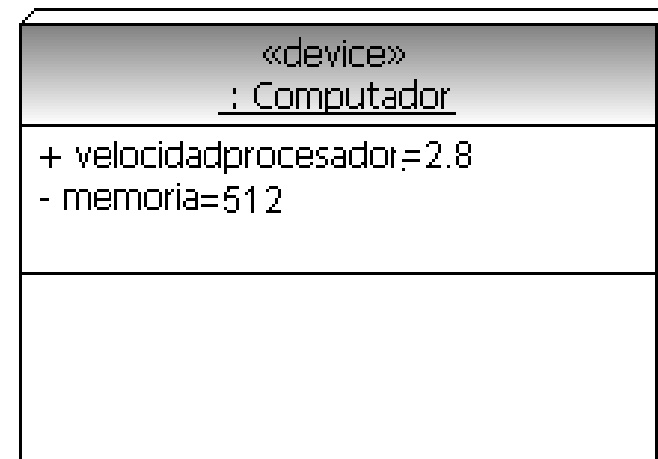
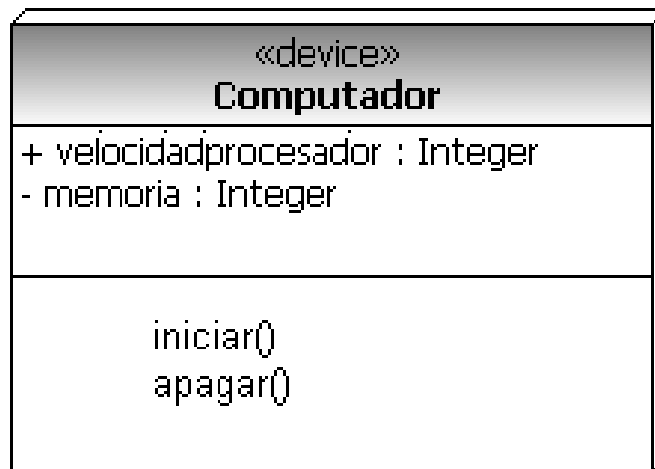


Diagrama de Despliegue...3

- **Rutas de Comunicación**
 - Especifican la relación entre nodos, el número de nodos que pueden ser conectados (multiplicidad) y la naturaleza de la conexión colocándole a la ruta un estereotipo.

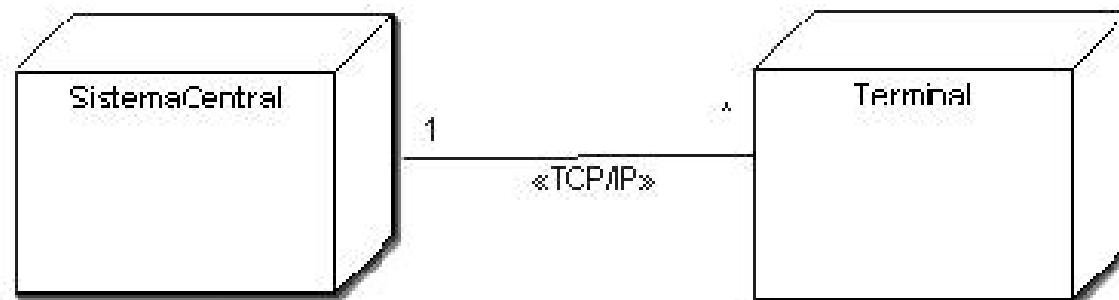
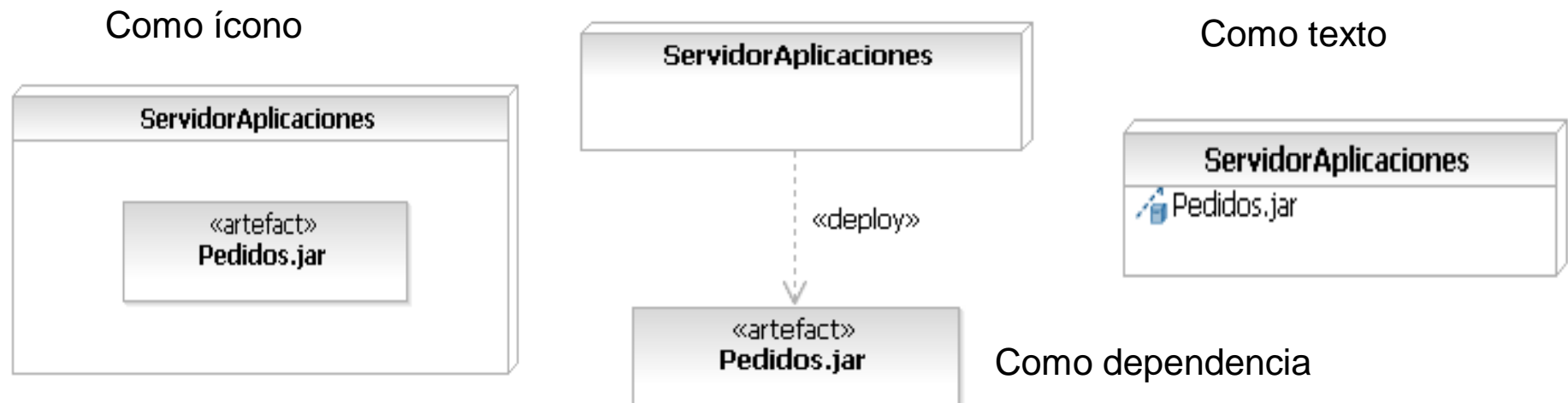


Diagrama de Despliegue...4

- **Artefacto Desplegado**

- Es una instancia de un artefacto que se ha desplegado en un nodo.
- Para UML 2.0 los componentes son representados como artefactos que implementan los requerimientos especificados por el componente.
- Los artefactos de despliegue se modelan como dependencias, las cuales tiene 3 maneras de representarse: **Contenido el nodo como ícono, contenido en el nodo como texto y como dependencia del nodo.**



Resumen

Ahora que ud. ha completado esta unidad, debe ser capaz de:

- Explicar los cambios básicos sufridos por la especificación UML 2.0 Explicar acerca de interfaces, roles y paquetes.
- Explicar de forma básica los diagramas de UML 2.0 actualizados en el aspecto estructural.
- Explicar de forma básica los nuevos diagramas incluidos en la especificación UML 2.0 en el aspecto estructural.
- Enumerar diferencias entre diagramas de la especificación UML 2.0 y la especificación anterior.

Introducción a UML 2.0 (Diagramas de Comportamiento)

Objetivos de Aprendizaje

Al finalizar esta unidad ud. debería ser capaz de:

- Explicar de forma básica los diagramas de UML 2.0 actualizados en el aspecto de comportamiento.
- Explicar de forma básica los nuevos diagramas incluidos en la especificación UML 2.0 en el aspecto de comportamiento.
- Enumerar diferencias entre diagramas de la especificación UML 2.0 y la especificación anterior.

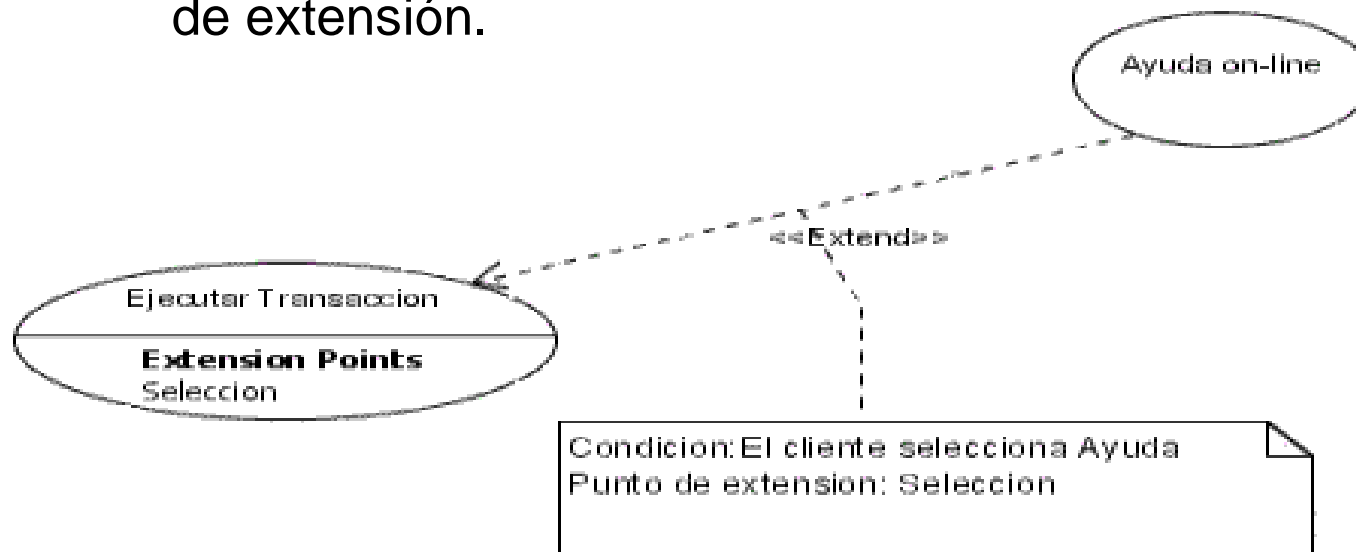
Diagramas de Casos de Uso

- Permiten modelar como los usuarios esperan utilizar el sistema.
- Describen quienes son los usuarios relevantes del sistema, los servicios que requieren los sistemas y los servicios que proporciona el sistema.
- Pueden ser aplicados a distintos tipos de desarrollos incluyendo sistemas manuales, pero se utilizan comúnmente para sistemas y subsistemas.

Diagramas de Casos de Uso...1

- **Punto de Extensión**

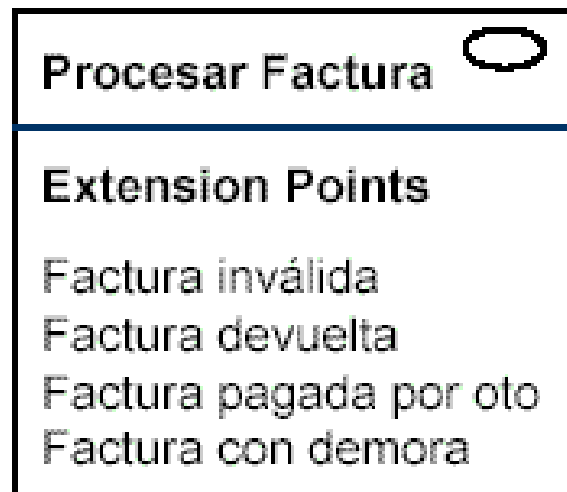
- Es el punto en el cual se agrega un caso de uso.
- Puede colocarse condiciones en notas adjuntas.
- La condición debe ser verdadera para que la extensión se realice, además es única para todos los puntos de extensión que tengan ese caso de uso.
- Si la condición se cumple, se ejecutan todos los fragmentos del caso de uso extensor correspondientes a todos los puntos de extensión.



Diagramas de Casos de Uso...2

- **Punto de Extensión**

- Existe una nueva notación para un caso de uso cuando existen muchos puntos de extensión, esta notación es llamada notación de clasificador.



Diagramas de Actividad

Representa el comportamiento del sistema mediante un modelo de flujo de datos y flujo de control.

- **Actividades**

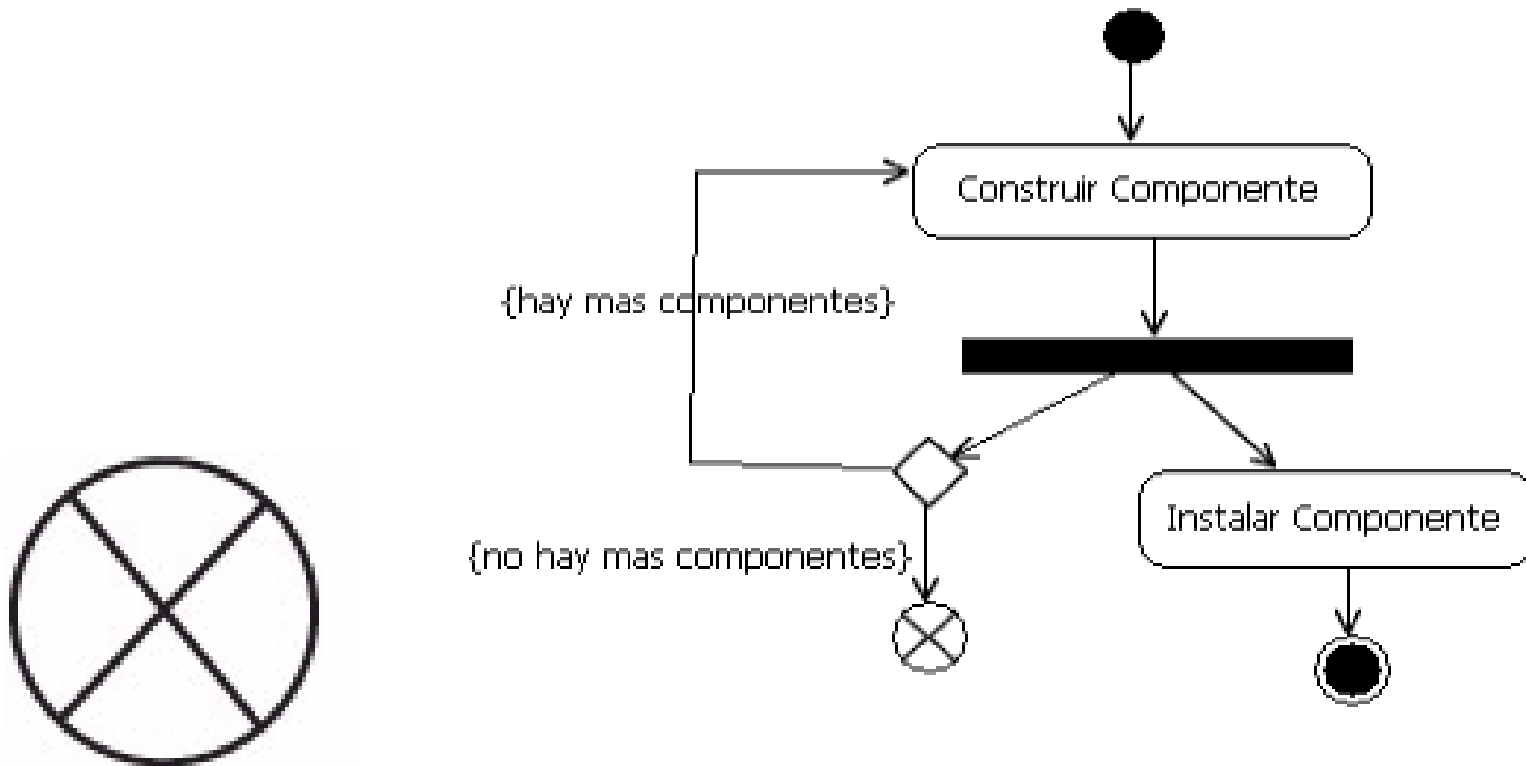
- Coordinan el comportamiento de las acciones mediante modelos de flujos de datos y de control.
- Las actividades pueden ser invocadas por las acciones.
- Un clasificador puede ser el contexto de una actividad, por ejemplo modelar el comportamiento de un objeto.

- **Acciones**

- Representan pasos simples, es decir, no pueden descomponerse dentro de una actividad.
- Su ejecución es controlada por la actividad que lo contiene a través de un flujo de control.
- Las acciones pueden pasar valores en la forma de flujo de objetos.

Diagramas de Actividad...1

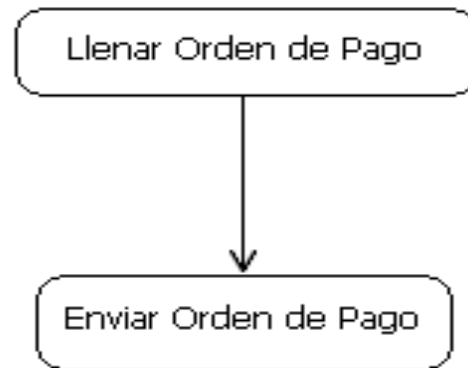
- **Nodos de Actividad Inicial, Final y Final de Flujo**



Diagramas de Actividad...2

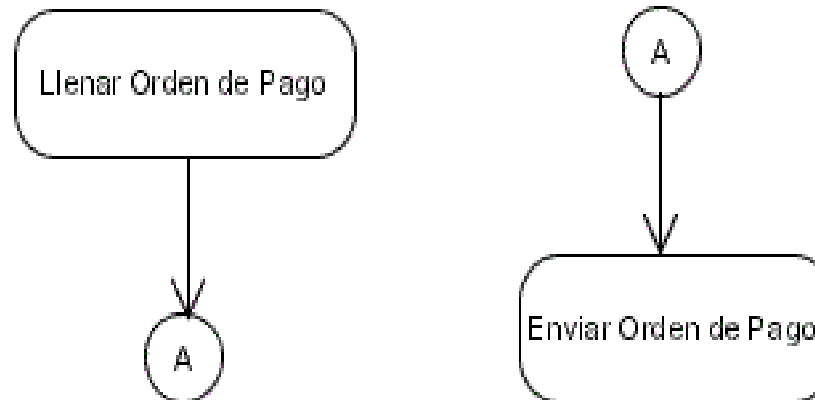
- **Arcos de Actividad**

- Modelan las conexiones entre una actividad y otra o una acción y otra.



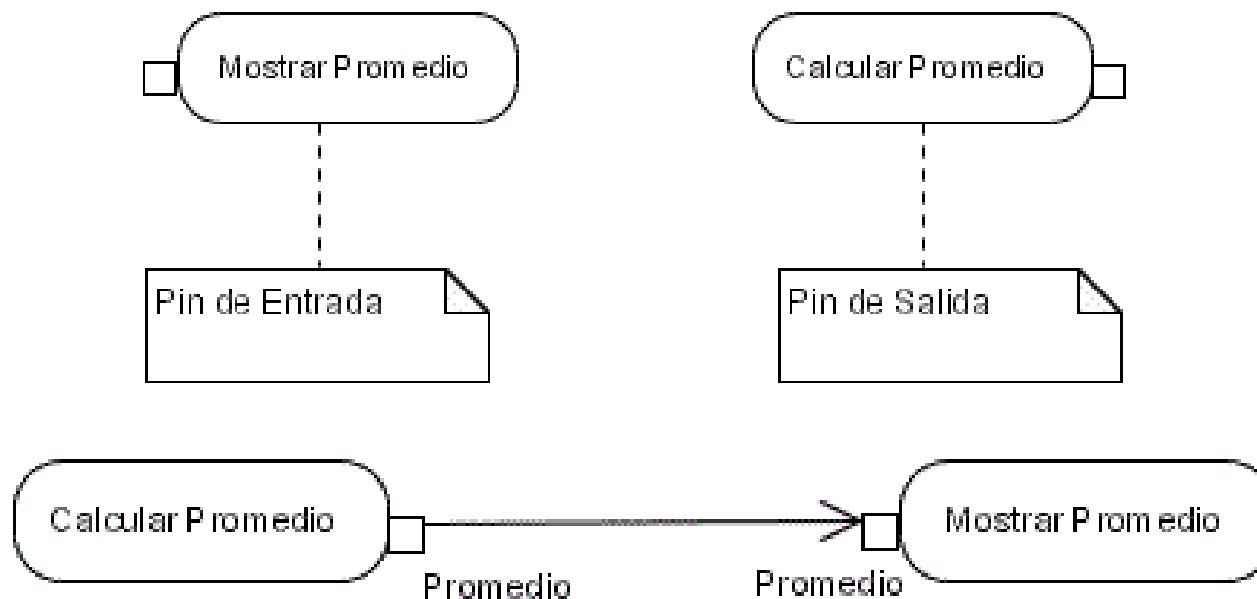
Un arco de actividad puede estar adornado con los siguientes elementos:

- Nombre
- Conectores



Diagramas de Actividad...3

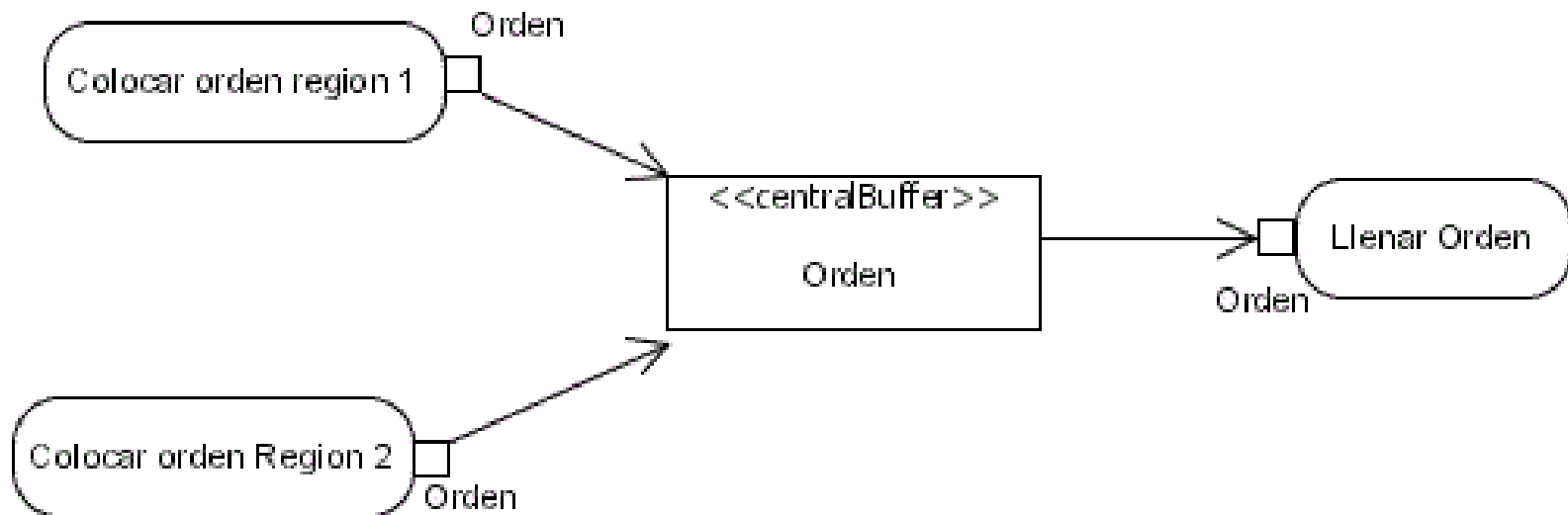
- **Pines de Entrada y Salida en Acciones**
 - Son nodos de objetos que contienen valores.
 - Se dividen en: Pin de Entrada y Pin de Salida.
 - Para que pueda conectarse un pin de entrada con un pin de salida, ambos debe ser del mismo tipo de datos.



Diagramas de Actividad...4

- **Nodo de Buffer Central**

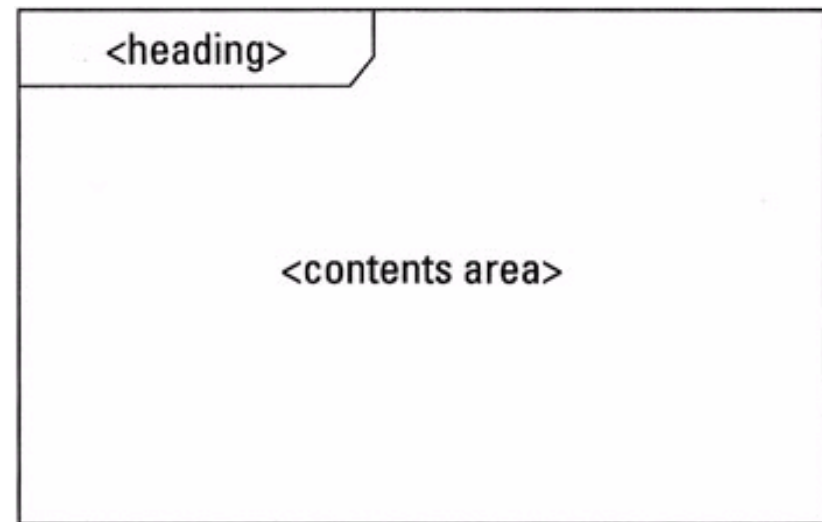
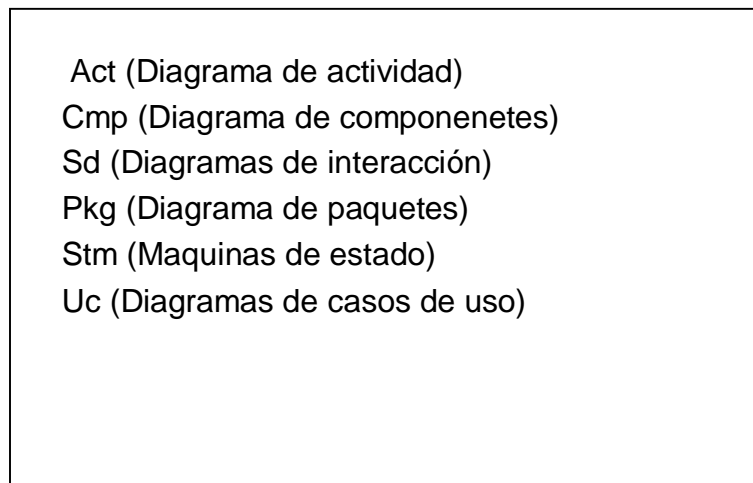
- Este tipo de nodo puede recibir datos de cualquier número de pines.
- Se dividen en: Pin de Entrada y Pin de Salida.
- Los pines de salida y de entrada pueden conectarse central en el nodo de buffer central para proporcionar o para extraer datos según los requisitos de las acciones asociadas.



Diagramas de Secuencia

- **Marcos**

- Son contenedores de diagramas, ellos proveen un contexto portable a los diagramas.
- Un marco puede estar dentro de otro marco, esto permite que el diagrama que incluye reutilice con eficacia el diagrama incluido.
- Sintaxis opcional del heading: [<tipo>] <nombre>[<parámetros.>].



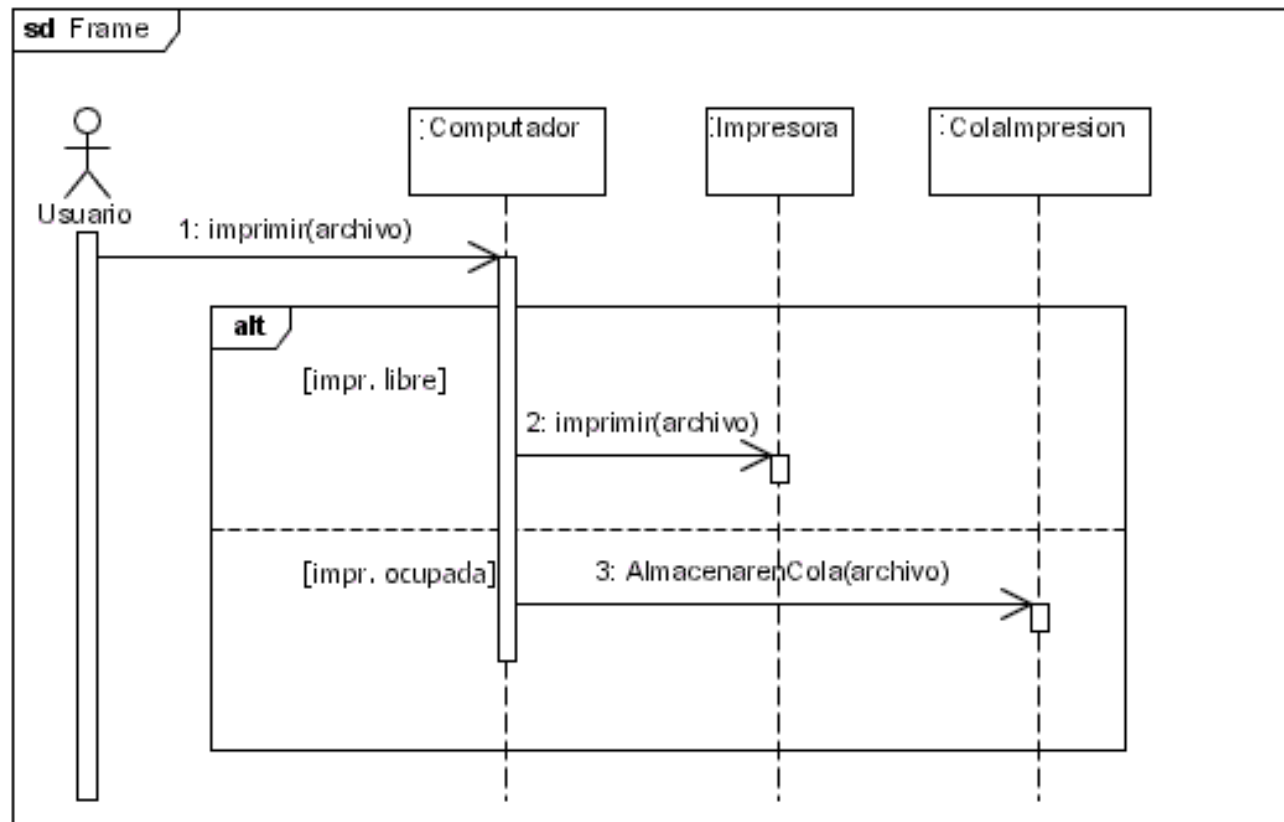
Diagramas de Secuencia...1

- **Fragmentos Combinados**
 - Encapsula porciones de un diagrama de secuencia por medio de un marco.
 - El nombre que aparece en la <<header>> del marco describe la función del fragmento y son llamados operándos de interacción.

Diagramas de Secuencia...2

- **Fragmentos Combinados...1**

- **Alternativos:** Funciona como una estructura condicional en cualquier lenguaje de programación, se ofrecen dos alternativas y el flujo de interacción entrará solo por una de ellas.

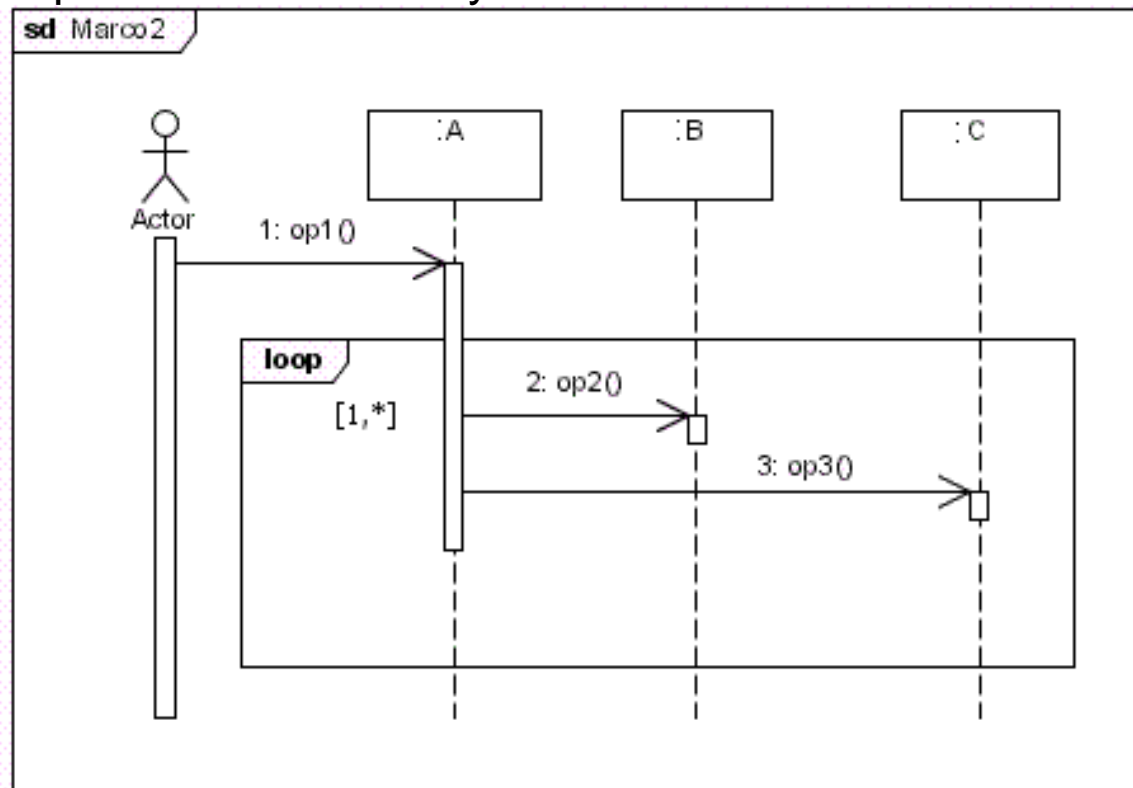


Diagramas de Secuencia...3

- **Fragmentos Combinados...2**

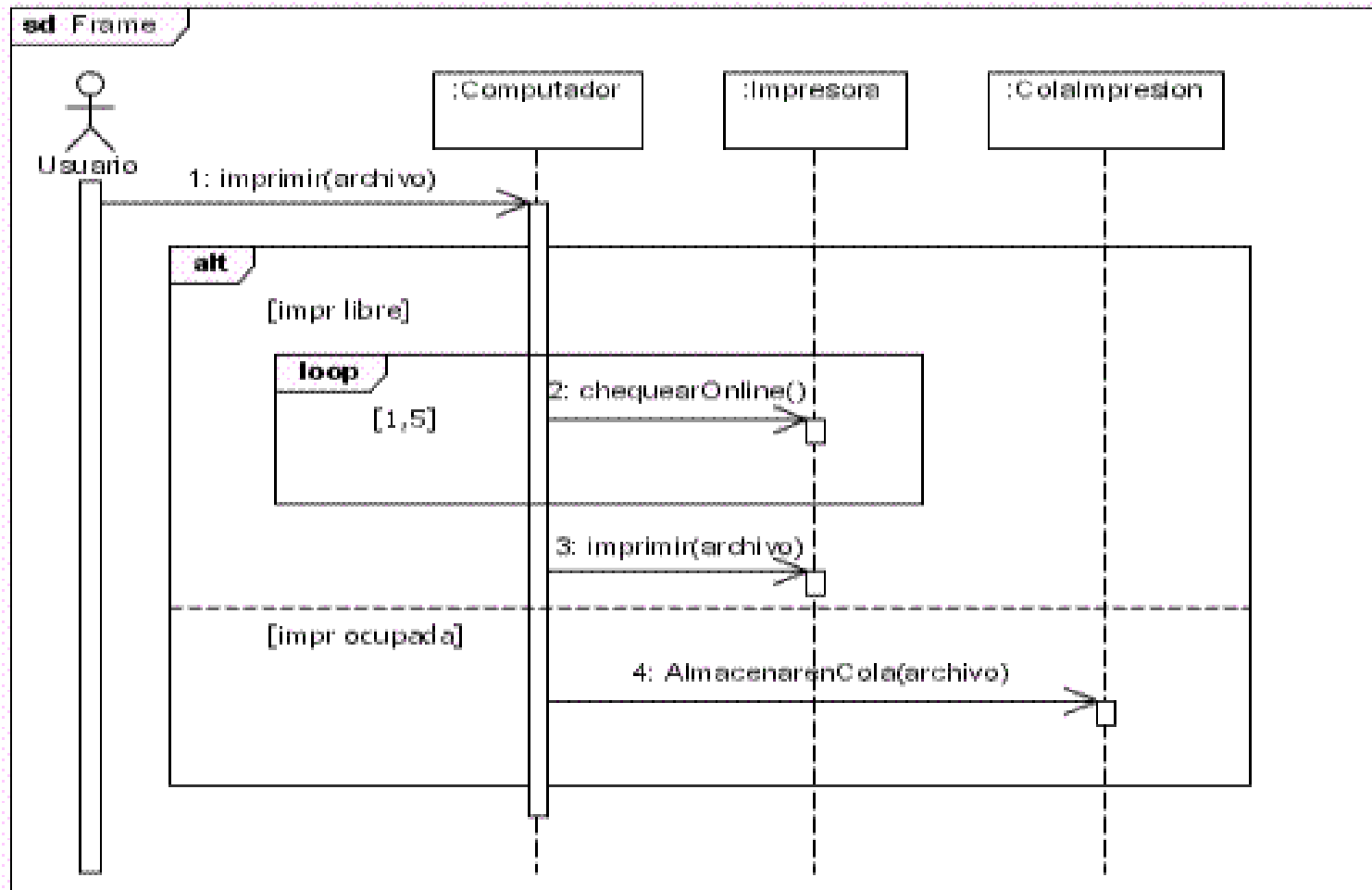
- **Ciclo (Loop)**

- Indica que el fragmento de interacción será ejecutada una cierta cantidad de veces.
 - El número de veces que se ejecuta es determinado por los parámetros “minint y maxint”.



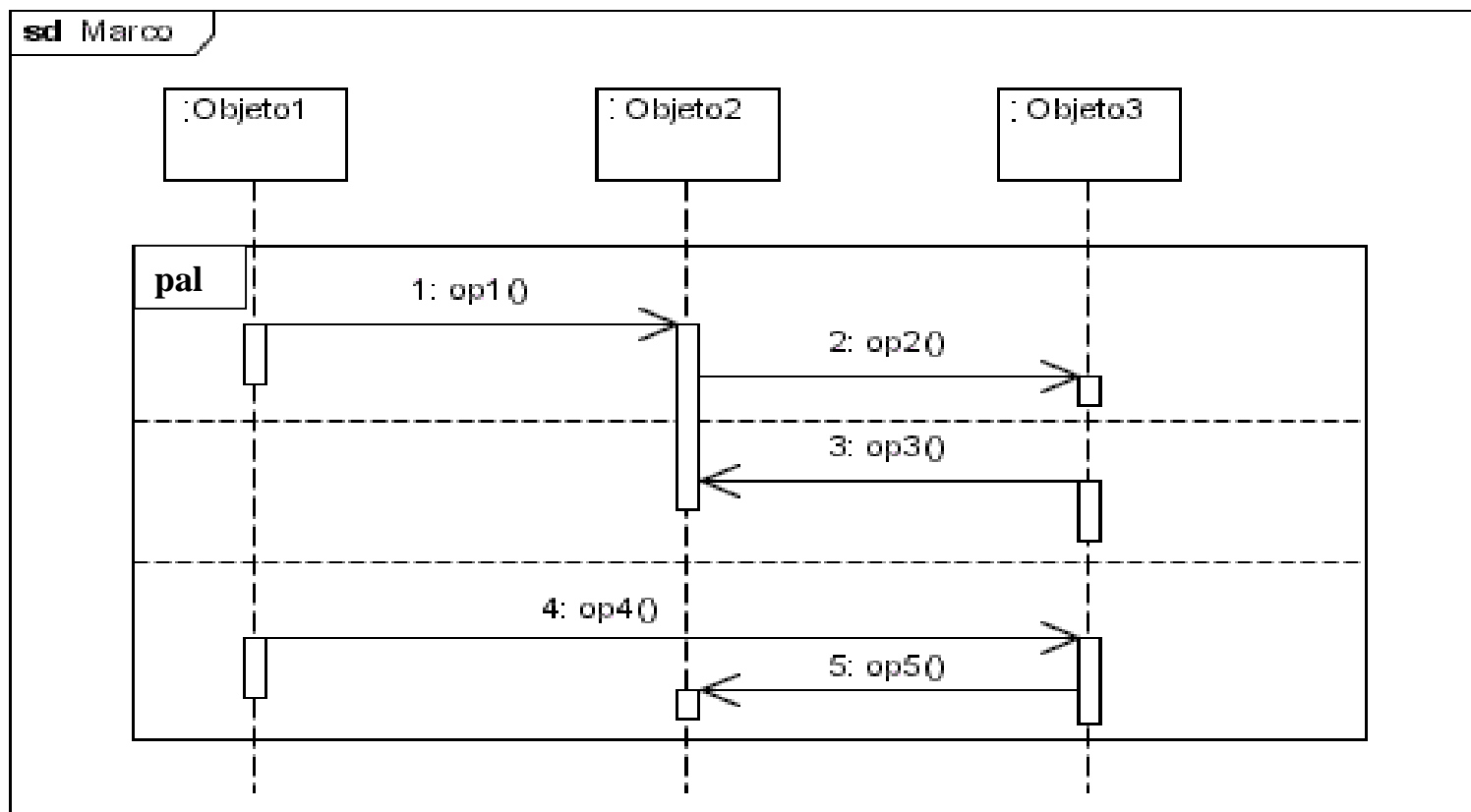
Diagramas de Secuencia...4

- **Fragmentos Combinados...3**
 - **Anidamiento de Fragmentos**



Diagramas de Secuencia...5

- **Fragmentos Combinados...4**
 - **Paralelo:** Cada una de las operaciones que están dentro de cada fragmento se ejecutan paralelamente.

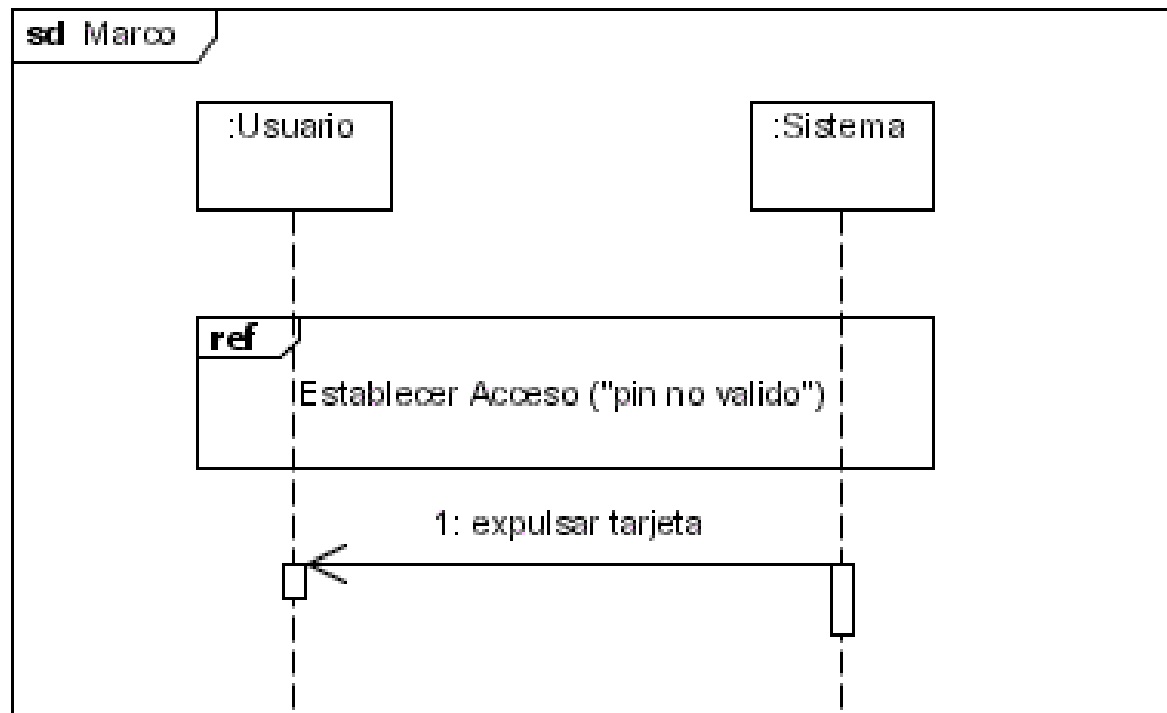


Diagramas de Secuencia...6

- **Fragmentos Combinados...5**

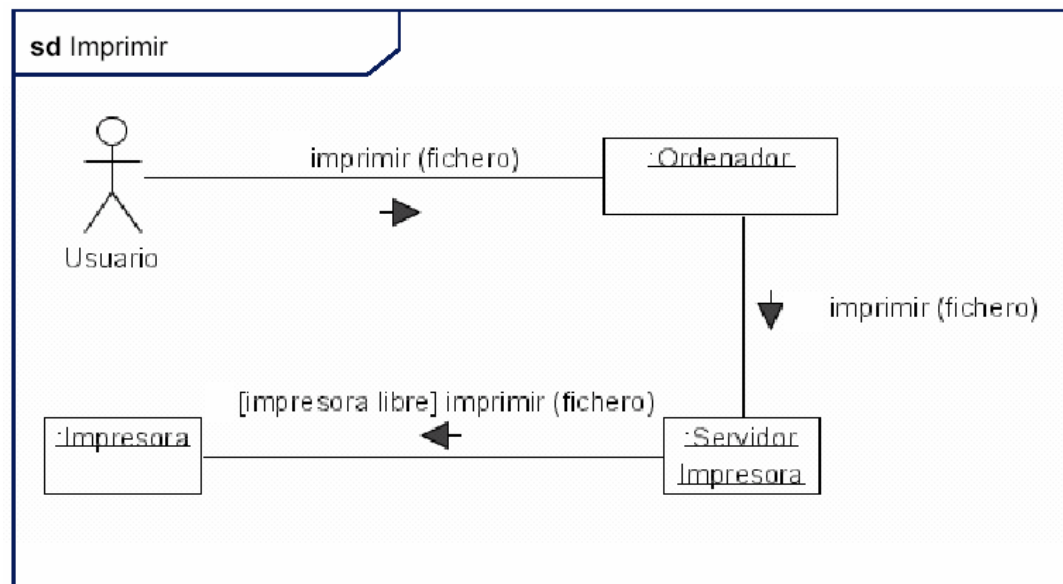
- **Ocurrencia de Interacción**

- Es un símbolo que refiere a una interacción que puede ser utilizada dentro de otra interacción.
 - El símbolo es un marco que en el <<header>> lleva la palabra “ref”.



Diagramas de Comunicación

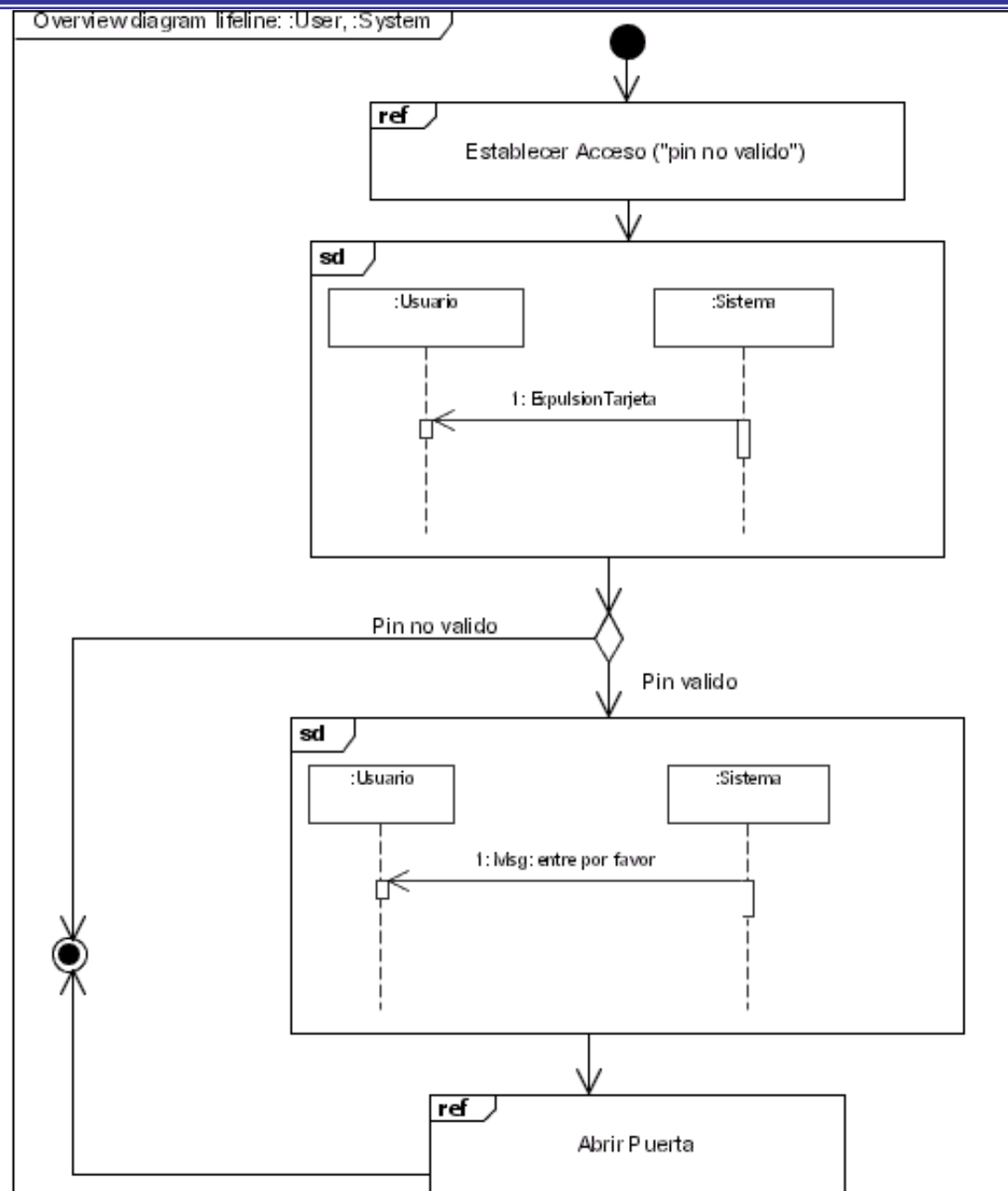
- Se centran en la interacción entre los objetos siguiendo una secuencia por medio de un esquema de números.
- Se representan dentro de un marco con el nombre del diagrama precedido del prefijo sd.
- **Características:**
 - Solo se representa el rectángulo de la línea de vida (objeto).
 - Los mensajes se colocan cerca de los enlaces.
 - Los enlaces se representan con una flecha y una etiqueta que contiene el nombre del mensaje e información adicional.



Diagramas de Visión General de Interacciones

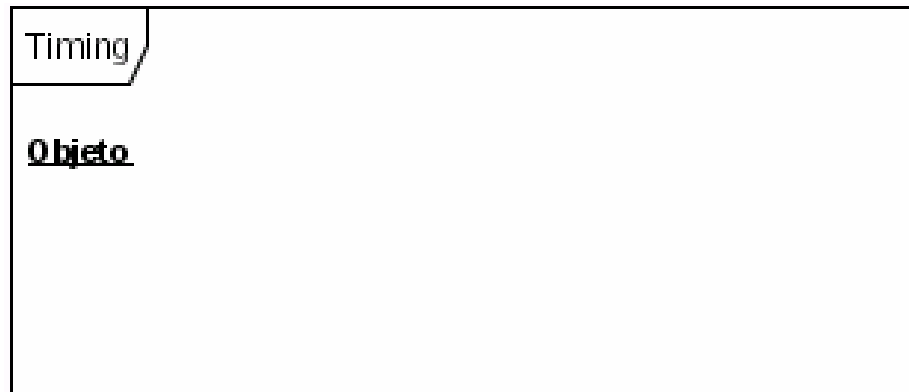
- Es una variante de los diagramas de actividad en los que los nodos son interacciones u ocurrencias de interacciones.
- **Características:**
 - Las interacciones u ocurrencias de interacción sustituyen a los nodos objetos de los diagramas de actividad y se consideran formas especiales de invocación de actividades.
 - Los fragmentos combinados alternativos se representan con un nodo de decisión y su correspondiente nodo fusión. Los enlaces se representan con una flecha y una etiqueta que contiene el nombre del mensaje e información adicional.
 - Los fragmentos combinados paralelos se representan con un nodo *fork* y su correspondientes nodo *join*.
 - Se pueden anidar estructuras.

Diagramas de Visión General de Interacciones...1



Diagramas de Tiempo

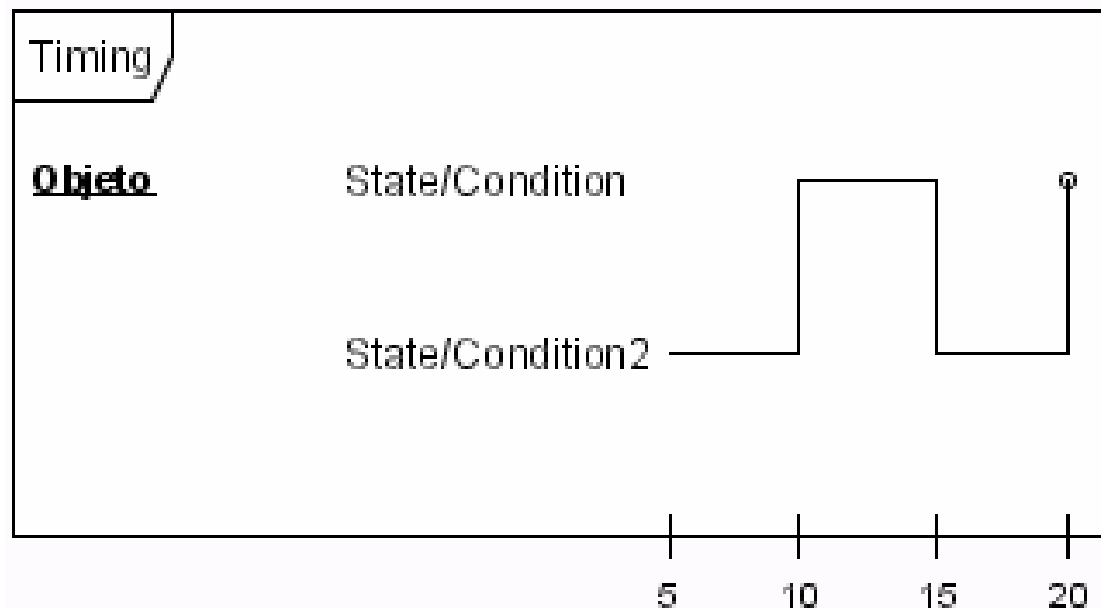
- Proporcionan una forma de mostrar los objetos activos y sus cambios de estado durante sus interacciones con otros objetos activos y con otros recursos del sistema.
- **Características:**
 - Para representarlos se utilizan marcos.
 - El eje X representa las unidades de tiempo y el eje Y representa los estados y transiciones.
 - Permite la representación de diferentes tipos de mensajes.
 - Los mensajes se pueden dividir en etiquetas para mejorar la legibilidad de los diagramas.



Diagramas de Tiempo...1

- **Marcas de Tiempo**

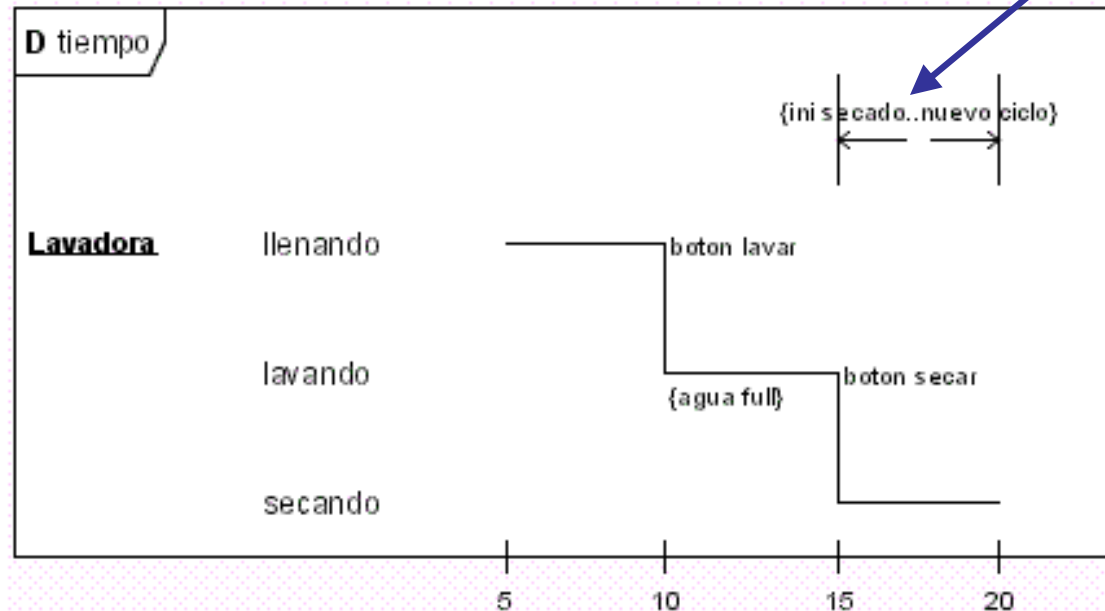
- Permiten saber el tiempo durado de un objeto en un estado específico.
- Se colocan en el eje de las X dentro del marco y se estila colocarlas con intervalos periódicos, por ejemplo 5 10 15 20



Diagramas de Tiempo...2

- **Restricciones de Duración**

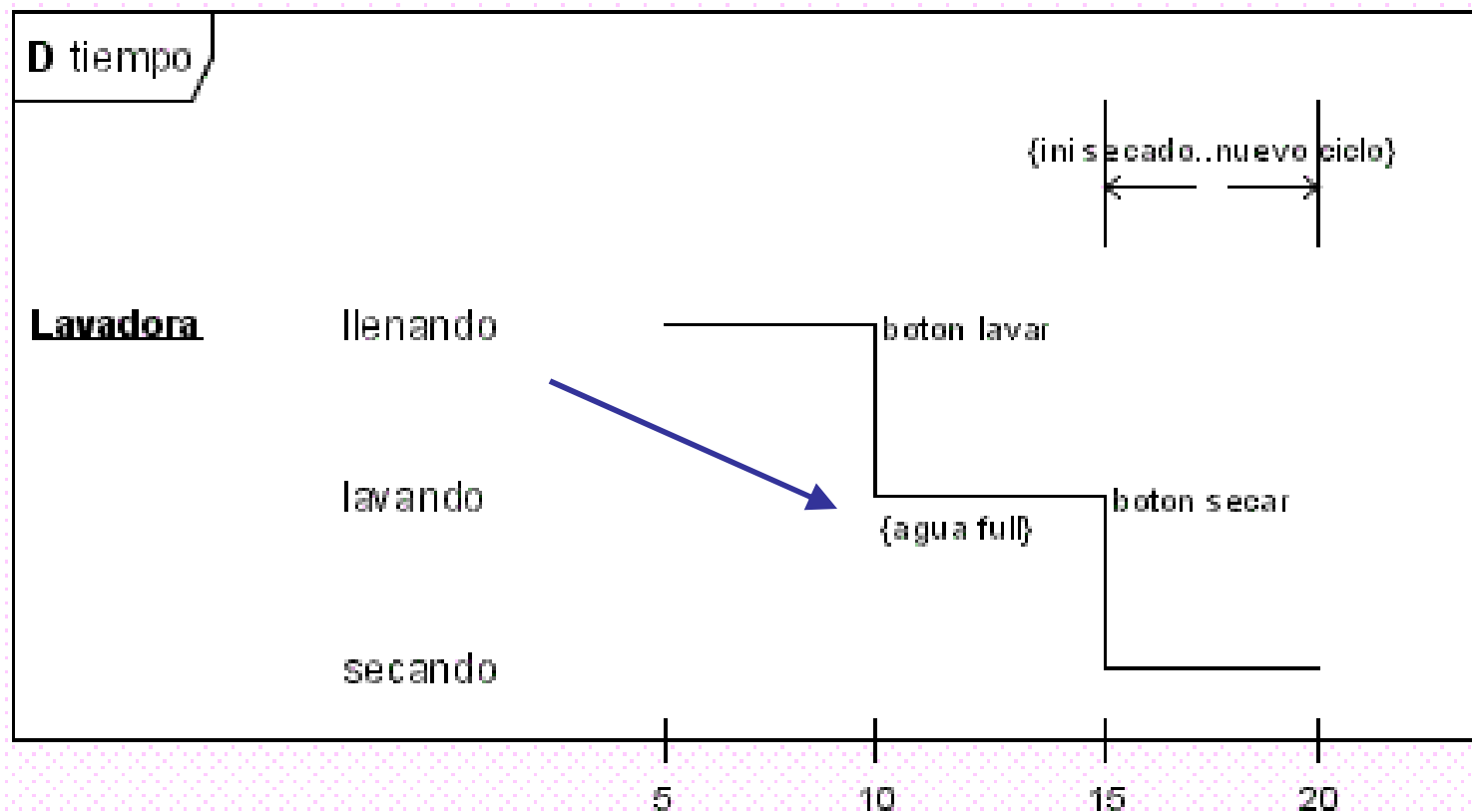
- Se agregan para restringir la longitud de tiempo entre los eventos.
- Tiene un límite de duración inferior y un límite de duración superior.
- Se colocan entre llaves y separados por dos puntos, por ejemplo: {Límite de duración inferior..Límite de duración superior}



Diagramas de Tiempo...3

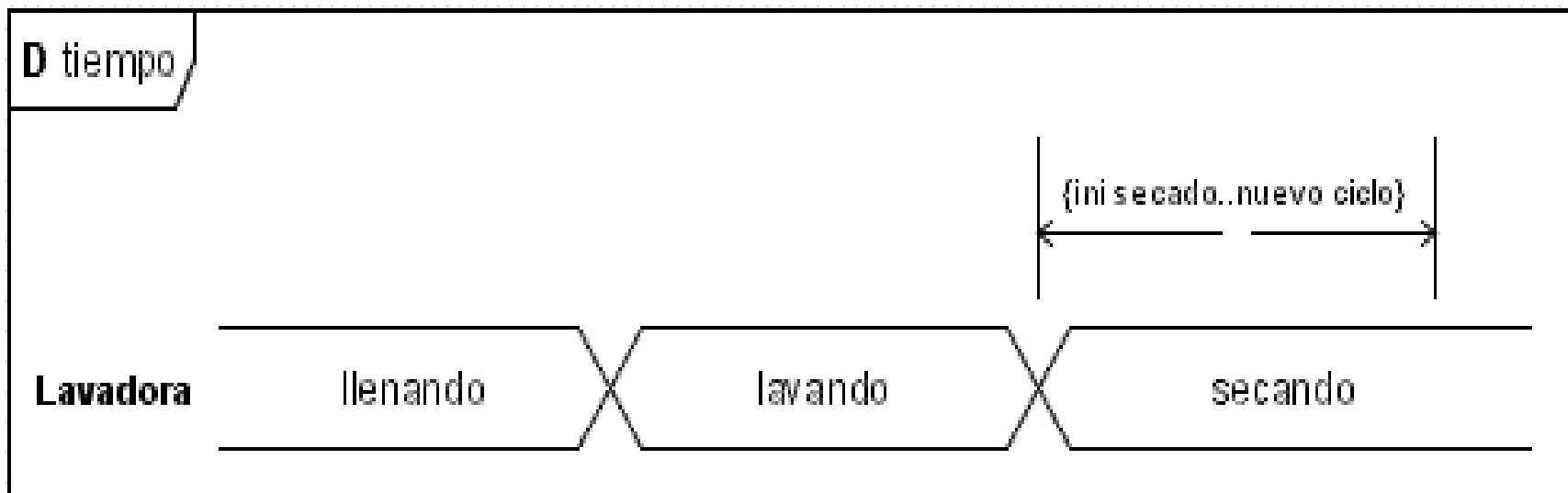
- **Restricciones de Tiempo**

- Son restricciones colocadas justo en una transición, es decir, en una marca de tiempo. Permite limitar la transición.



Diagramas de Tiempo...4

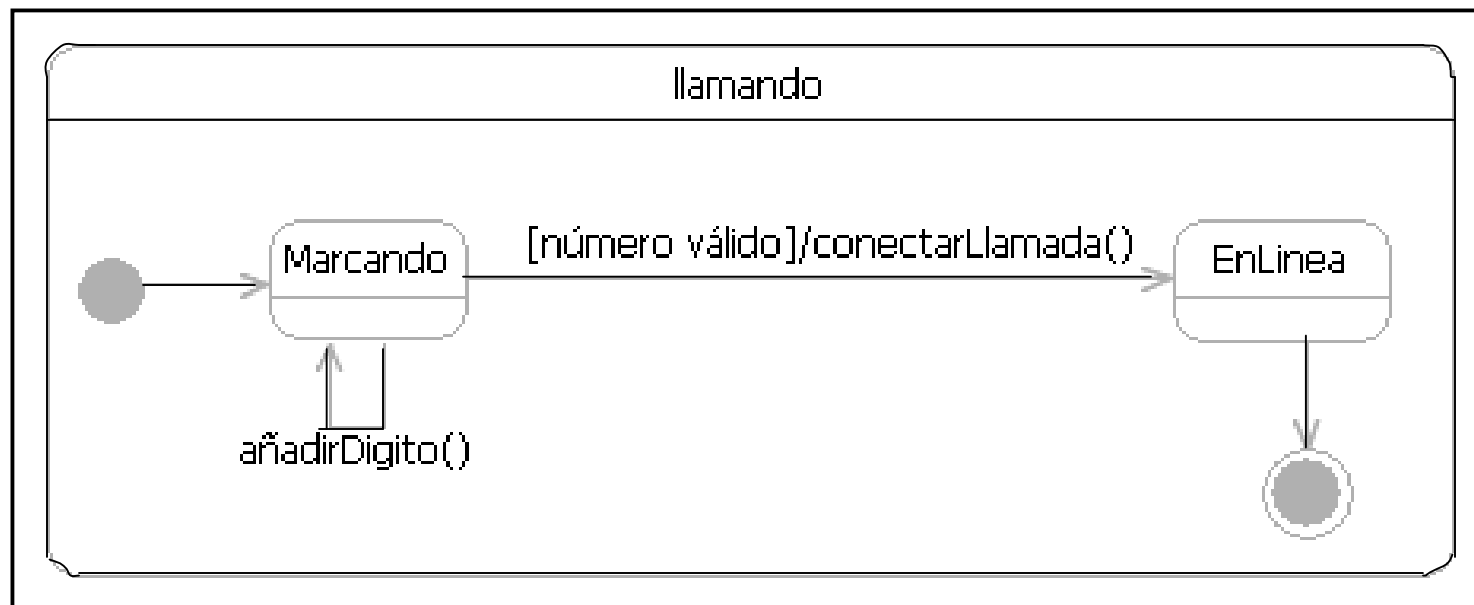
- **Vista Alternativa**



Diagramas de Máquinas de Estado

- **Estados Compuestos**

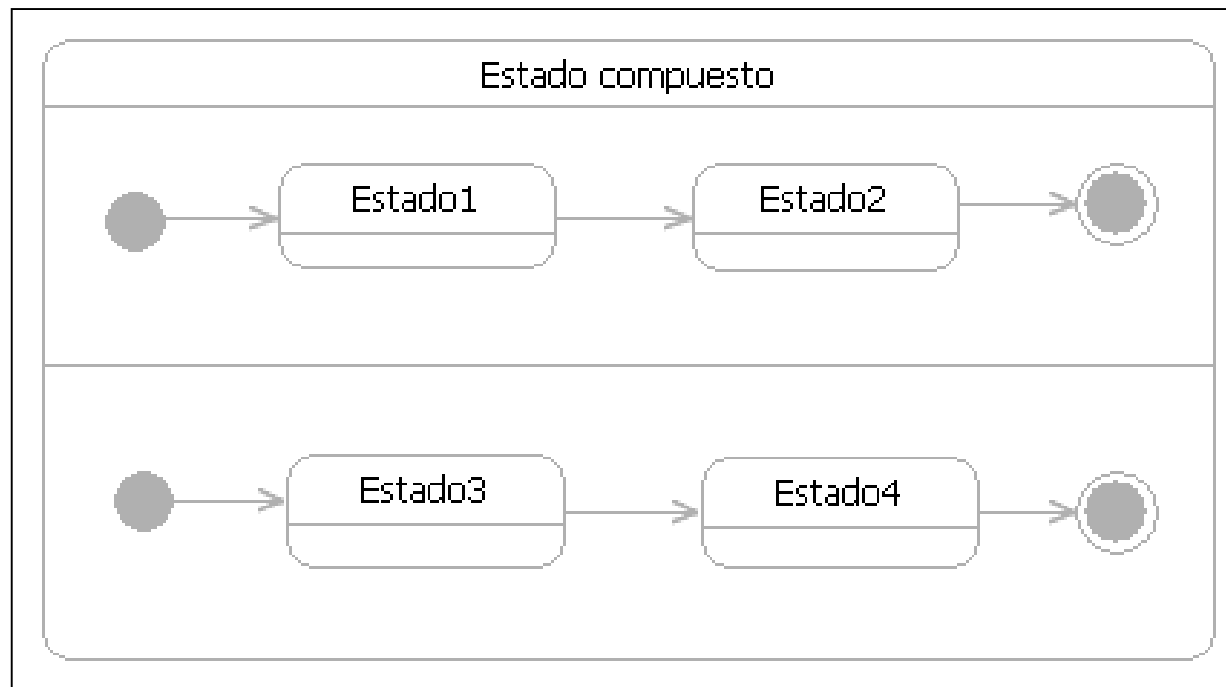
- Un estado puede ser descompuesto para proveer una descripción más específica de los comportamientos de los objetos.
- Aparte de proporcionar la posibilidad a los modelos de dar mucho más detalle, le permite tener estados concurrentes.



Diagramas de Máquinas de Estado...1

- **Regiones**

- Son partes en la que se divide un estado compuesto para permitir en muchos casos la concurrencia entre estados.
- Los estados son mutuamente exclusivos entre regiones, es decir un estado solamente puede pertenecer a una sola región.

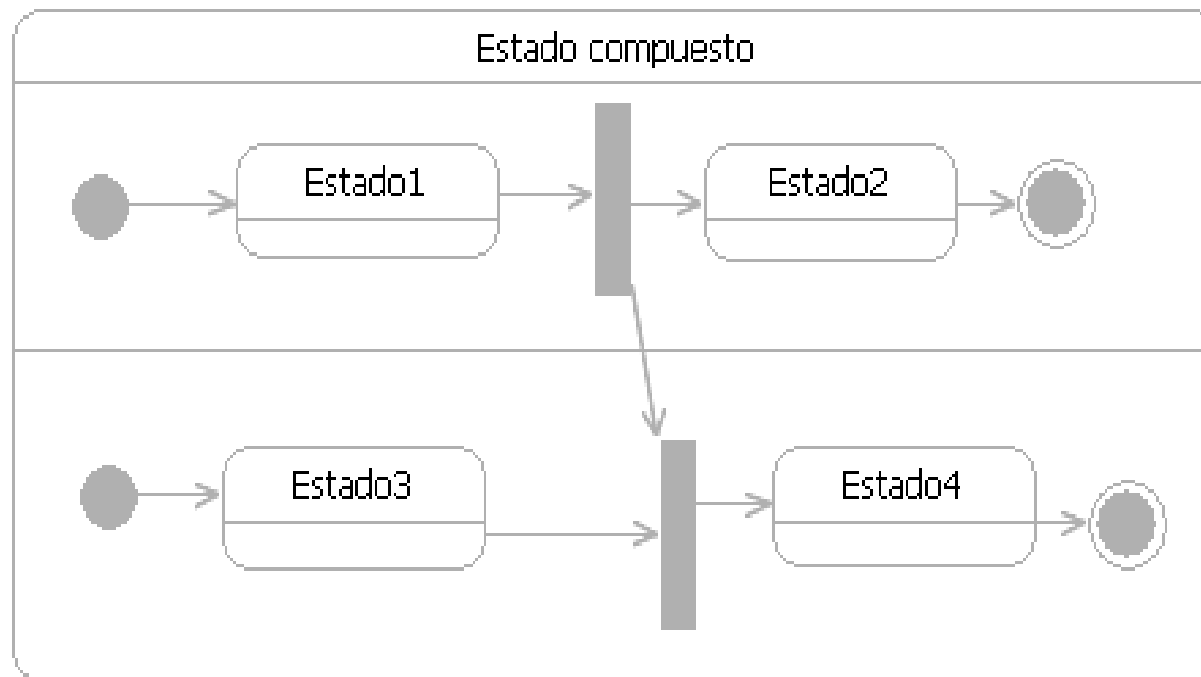


Diagramas de Máquinas de Estado...2

- **Unión y Bifurcación**

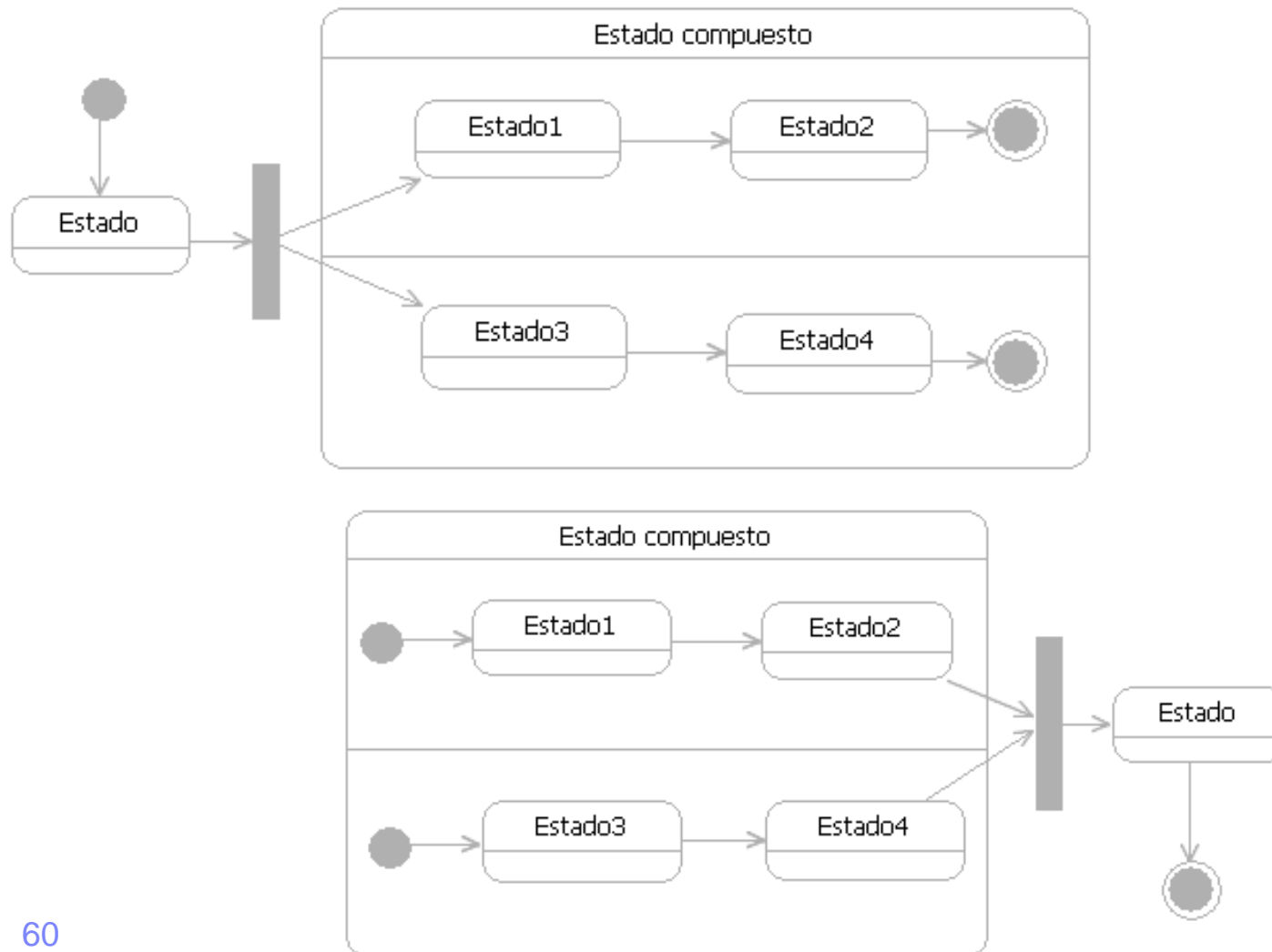
- Son dos tipos de pseudos estados que permiten aplicar la concurrencia y relacionar las regiones de los estados compuestos.

Unión y bifurcación dentro de regiones



Diagramas de Máquinas de Estado...3

Unión y bifurcación fuera de regiones



Resumen

Ahora que ud. ha completado esta unidad, debe ser capaz de:

- Explicar de forma básica los diagramas de UML 2.0 actualizados en el aspecto de comportamiento.
- Explicar de forma básica los nuevos diagramas incluidos en la especificación UML 2.0 en el aspecto de comportamiento.
- Enumerar diferencias entre diagramas de la especificación UML 2.0 y la especificación anterior.