

A large, antique-style globe is positioned on the left side of the slide. It has a weathered, yellowish-brown surface with visible map details and a brass-colored metal ring around its equator.

UML – Lenguaje de Modelado Unificado

Escuela de Computación - Facultad de Ciencias UCV - Profa. Zulma González

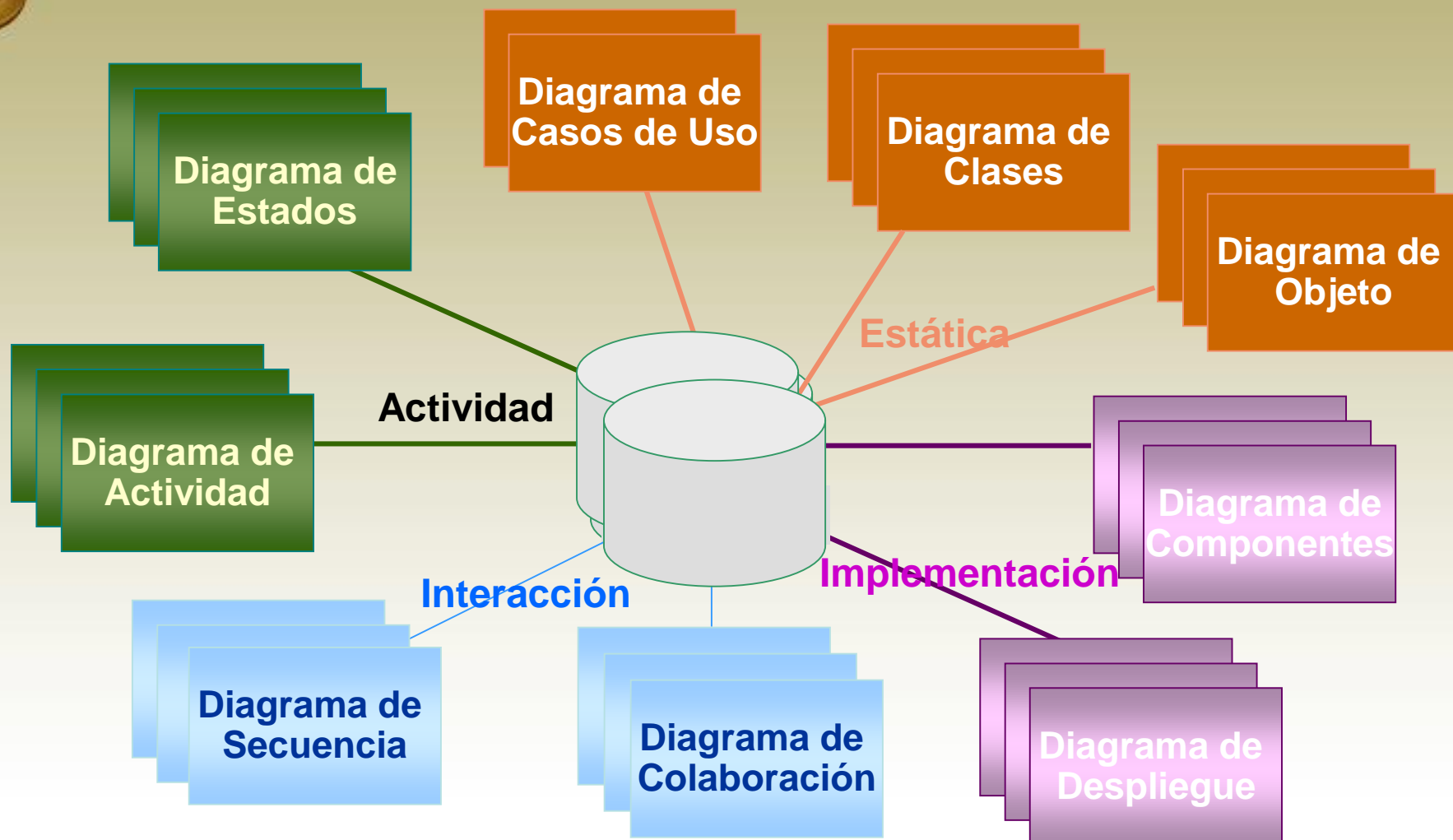


UML

- UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.
- Uno de los objetivos principales de la creación de UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una **notación y semántica común**.
- **Proyecta** las diferentes **vistas** de un sistema.



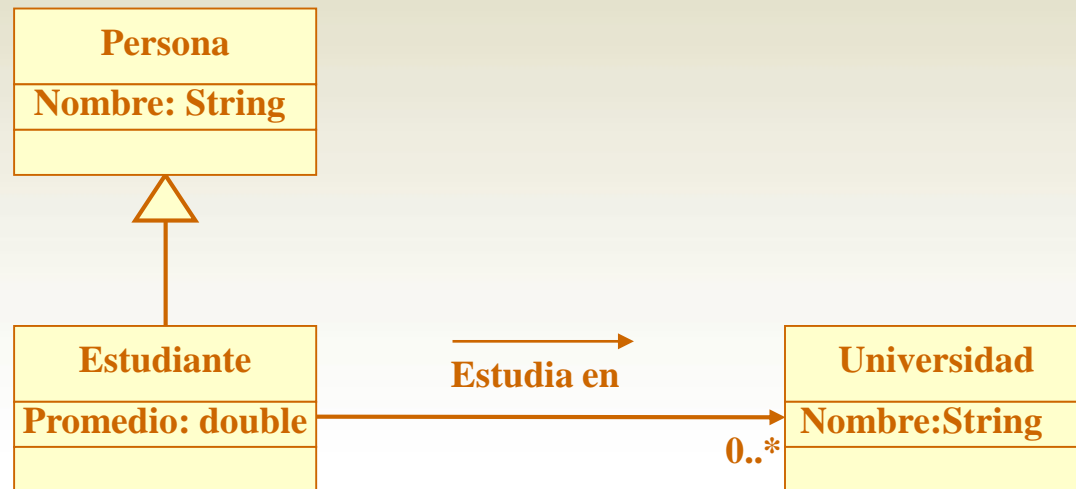
Vistas





Estructura

- **Diagrama de Clases:**
 - Describe la **estructura estática** del sistema.
 - Presenta el conjunto de clases, interfaces y colaboraciones, así como sus relaciones, cubriendo la vista de diseño estática del sistema.





Estructura

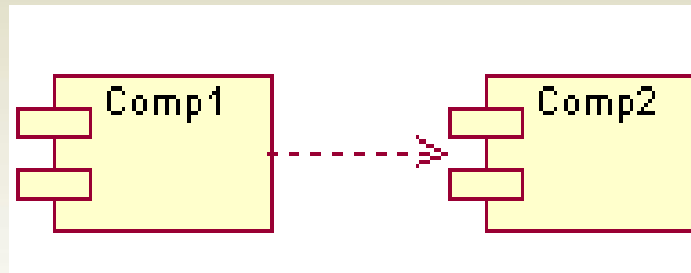
- **Diagrama de Objetos:**
 - Análogo al diagrama de clases, muestra un conjunto de objetos y sus relaciones, en un instante dado.





Estructura

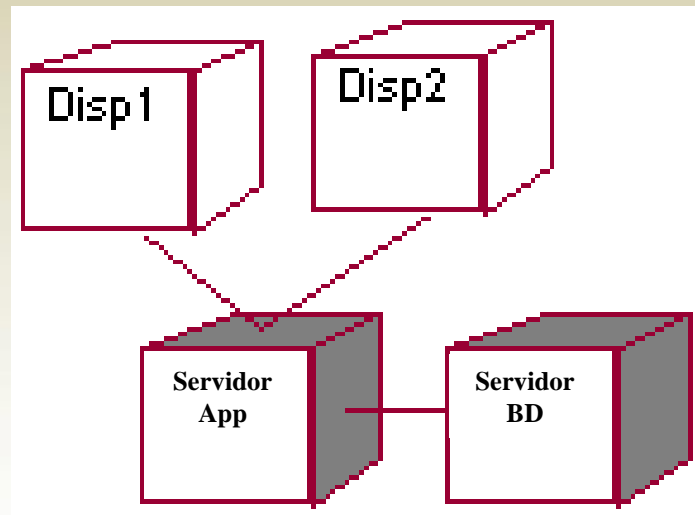
- **Diagrama de Componentes:**
 - Muestra la **organización** y **dependencias** de un conjunto de **componentes**.
 - Cubren la vista de implementación de un sistema, y describen la interacción entre componentes de Software.





Estructura

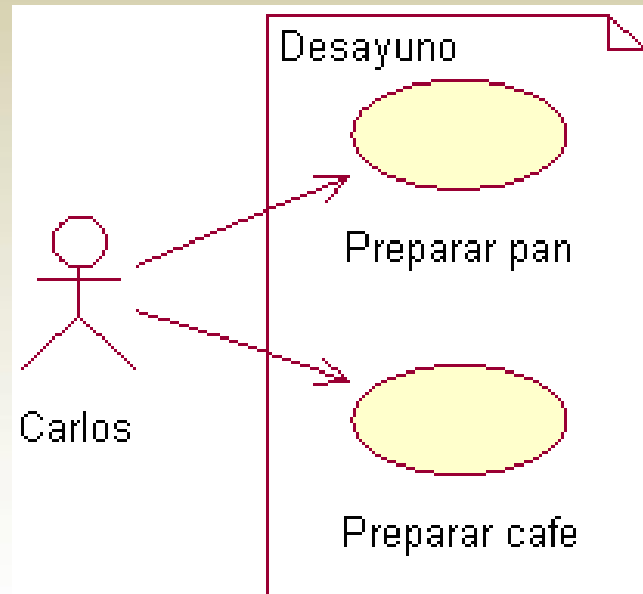
- **Diagrama de Despliegue:**
 - Describe la **disposición del hardware**.
 - Muestra la configuración del hardware del sistema, los nodos de proceso y los componentes empleados por éstos.





Comportamiento

- **Diagrama de Casos de Uso:**
 - Describe las **funcionalidades** del sistema a partir de las interacciones del usuario.

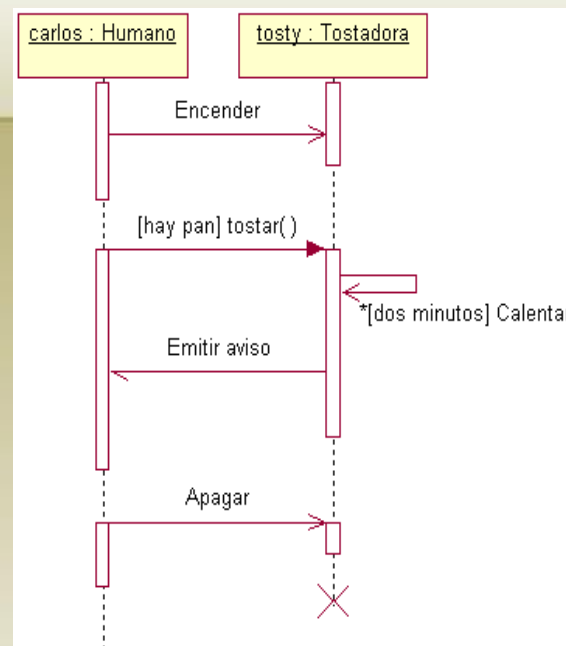




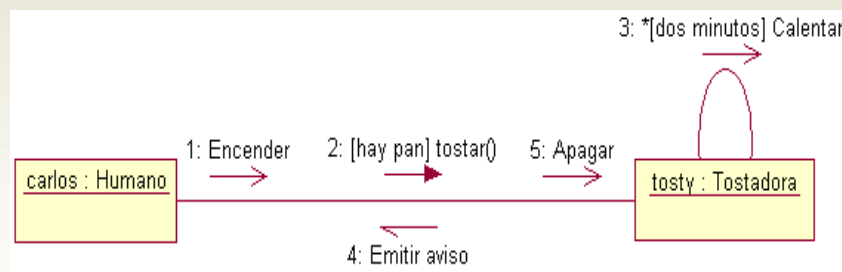
Comportamiento

- **Diagrama de Secuencia y de Colaboración:**

- Son diagramas de **interacción**, muestran un conjunto de objetos y sus relaciones, así como los mensajes que se intercambian entre ellos.
- El diagrama de **secuencia** resalta la ordenación temporal de los mensajes, mientras que el de **colaboración** resalta la organización estructural de los objetos, ambos siendo equivalentes o isomorfos.



Diag. Secuencia

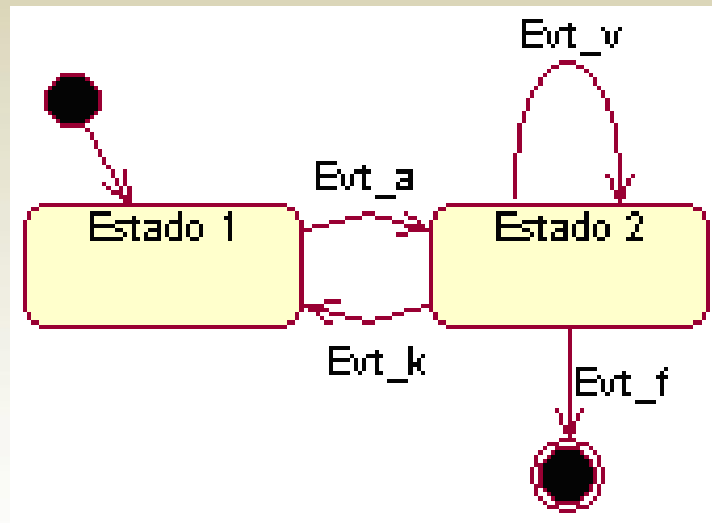


Diag. Colaboración



Comportamiento

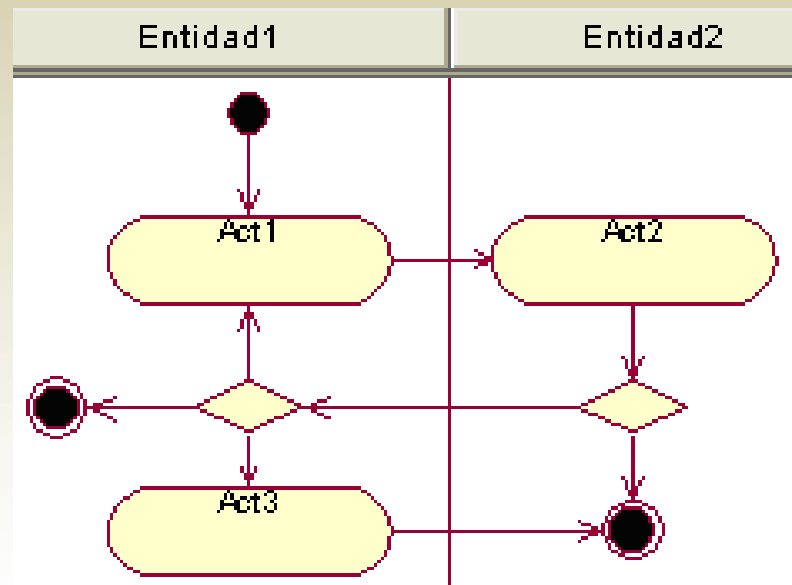
- **Diagrama de Estados:**
 - Muestra una **máquina de estados** de un objeto, con sus estados, transiciones, eventos y actividades.
 - Modelan comportamientos reactivos en base a eventos.





Comportamiento

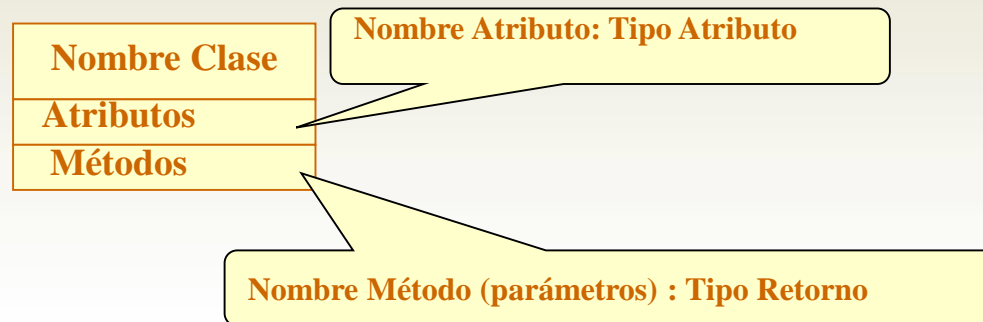
- **Diagrama de Actividades:**
 - Describe el **flujo de trabajo**, muestra las actividades, su secuenciamiento y coordinación





Diagramas de Clases

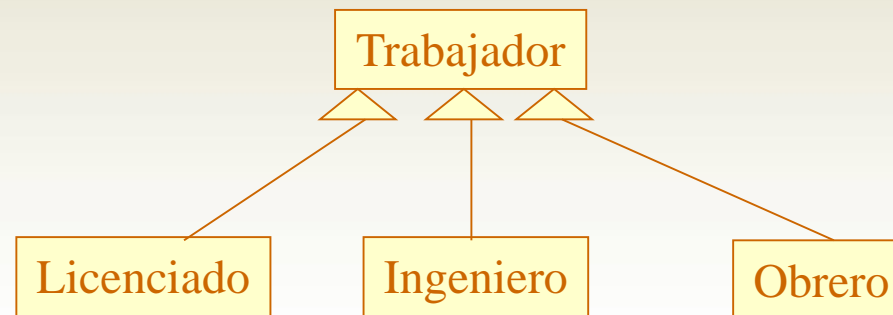
- Un **diagrama de clase** describe la estructura estática del sistema, mostrando sus clases y las relaciones entre estas.
- Una **clase** es la definición de un conjunto de objetos con características y comportamiento similares.





Diagramas de Clases

- *Clases Abstractas:*
 - Una **clase abstracta** es aquella que no tiene instancias directas pero cuyas clases descendientes tienen instancias directas
- *Clases Concretas:*
 - Son todas aquellas clases que pueden ser instanciadas.





Diagramas de Clases

- **Restricciones (*constraints*):**
 - *Restricciones de Atributos:* Permiten adicionar reglas en atributos.

Nombre Clase
Lista de Atributos {restricción}

Rectángulo
Ancho Largo Area
{Area = Ancho * Largo} {Ancho = Largo}

- *Restricciones en relaciones:* Reglas a las relaciones.



Diagramas de Clases

- Estereotipos: <<estereotipo>>
 - Es un *metatipo*, cuya utilidad es definir a otros tipos.
 - Define el propósito del conjunto de elementos a modelar con el mismo estereotipo.
 - Es opcional.

<<estereotipo>>
Nombre Clase

<<interfaz>>
Ventana_Principal

<<modelo>>
Persona

<<control>>
Main



Diagramas de Clases

- **Relaciones entre clases**
 - Conexión semántica entre elementos del modelo.
- Tipos de Relaciones entre clases:
 - Asociación.
 - Agregación.
 - Composición.
 - Generalización / Especialización.
 - Dependencia.



Diagramas de Clases

- **Asociación:**
 - Relación o invocación significativa entre dos o más clases.



- Según notación UML, la asociación comprende:
 - *Descripción*, o nombre de la relación.
 - *Rol*: Responsabilidad de la clase en la relación.
 - *Multiplicidad*: Indica cuantos objetos pueden participar en la relación.
 - 0 ó más: *
 - 1 o más: 1..*
 - De 2 a 4: 2..4
 - Sólo 7: 7



Diagramas de Clases

- **Asociación:**

- *Grado de la Asociación:* Se determina por el número de clases conectadas por la misma asociación. Las asociaciones pueden ser binarias ternarias o de mayor grado.
- Ejemplo de una *asociación ternaria*:

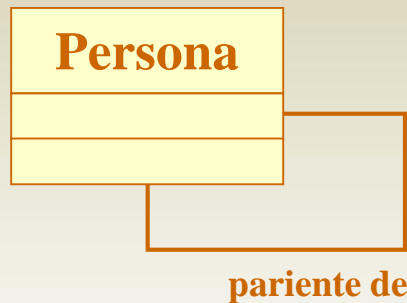




Diagrama de Clases

- **Asociación:**

- Las *asociaciones* pueden ser *reflexivas*, es decir pueden relacionar distintos objetos de una misma clase.





Diagramas de Clases

- **Agregación:** (“es parte de”, “contiene”)
 - Asociación que especifica relación *Parte de* entre el agregado (Todo) y el componente (Parte).





Diagramas de Clases

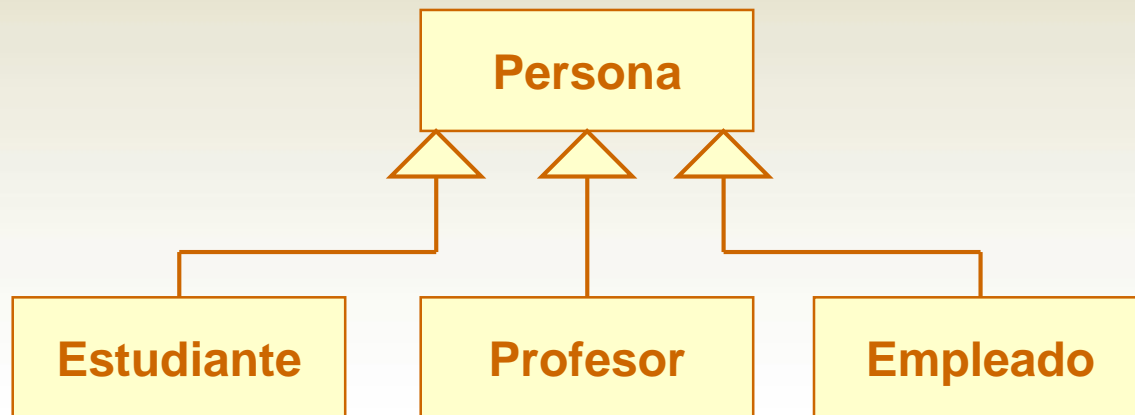
- **Composición:** (“compuesto por”)
 - Relación de agregación especial donde las partes no pueden existir sin que exista el objeto todo.





Diagramas de Clases

- **Generalización / Especialización:** (“es un”)
 - *Generalización:* Se crea una clase (superclase), que generaliza las propiedades comunes de varias clases.
 - *Especialización:* Dada una clase, se crea(n) otra(s) clase(s) (subclase) que especializa(n) la clase dada, agregando las diferencias.





Diagramas de Clases

- **Dependencia:**

- Es una conexión entre clases que indica que un cambio en una clase B puede afectar a otra clase A que la *usa*.





Diagramas de Clases

- **Clase Asociativa:**
 - Asociación entre clases, donde la relación posee atributos propios.
 - Cada enlace es una instancia de clase.





Diagramas de Objetos

- Un **diagrama de objetos** muestra un conjunto de objetos y sus relaciones en un momento determinado.

Nombre del Objeto: Nombre de la Clase

Atributo: Valor



Diagramas de Objetos

- Ejemplo:





Diagramas de Casos de Uso

- Los diagramas de Casos de Uso describen lo que hace un sistema, enfatizando el **qué** en vez del cómo.
- Describen las funcionalidades del sistema a partir de las interacciones del usuario.
Es decir, describen un uso del sistema y cómo este interactúa con el usuario.
- Se emplean para visualizar el comportamiento del sistema.



Diagramas de Casos de Uso

Actores

- **Entidad externa** que interactúa con el sistema.
- Entidades distintas a los usuarios de sistema.
- En algunos casos, representan cierta función que un usuario va a realizar en el sistema.

Personas

Sistema



Nombre del Actor

Componente de Software

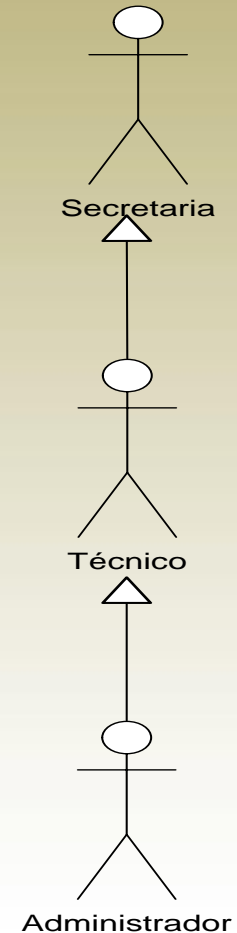
Organización



Diagramas de Casos de Uso

Actores

- Relaciones entre actores:
 - Generalización:
Cuando diferentes actores realizan roles similares, pueden heredar de un actor común.

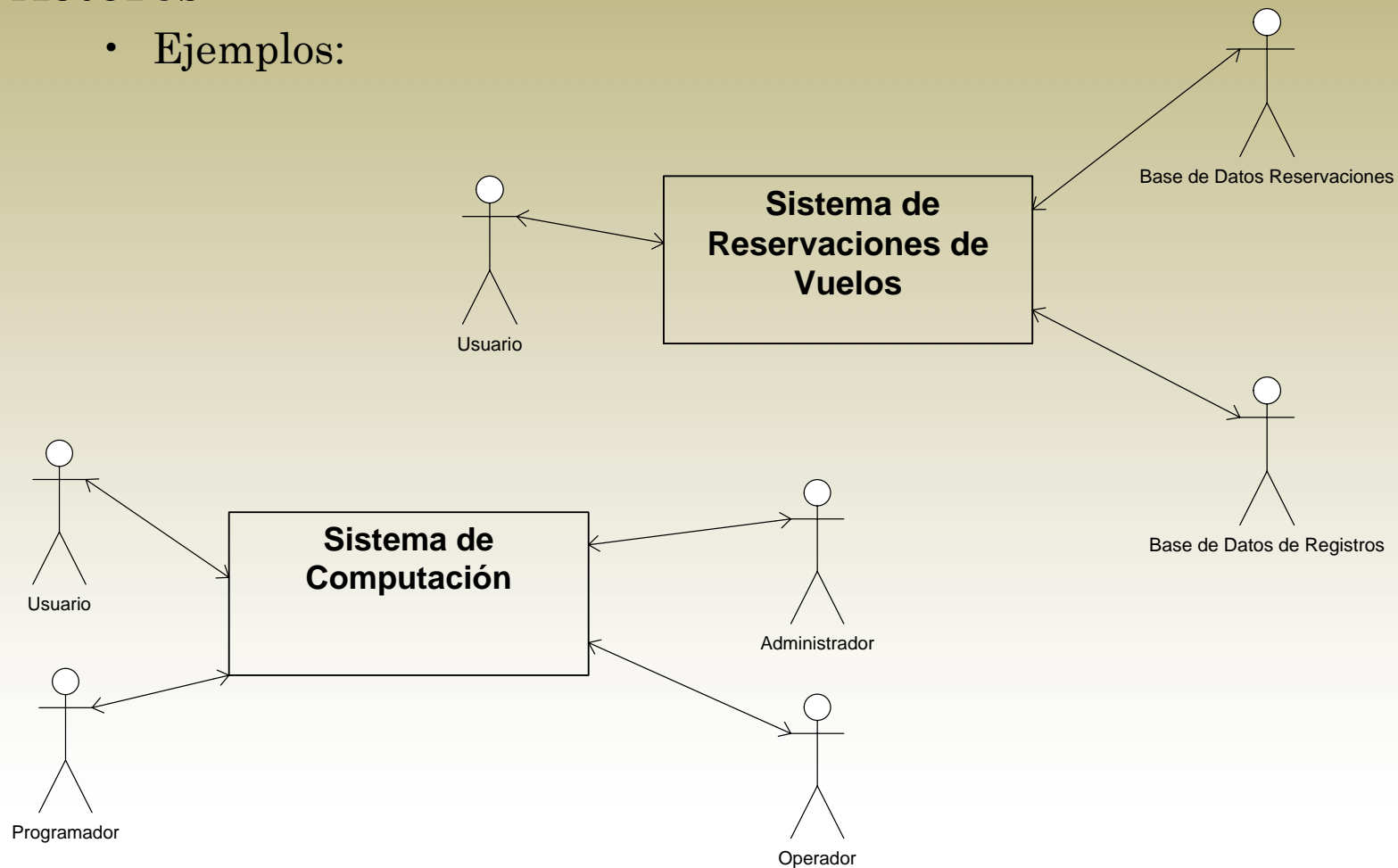




Diagramas de Casos de Uso

Actores

- Ejemplos:





Diagramas de Casos de Uso

Casos de Uso

- Un caso de uso define una **funcionalidad** del sistema.
- Cada caso de uso constituye un **flujo de eventos**, que especifican la interacción que toma lugar entre el actor y el sistema.
- Cada caso de uso **produce** un **resultado** observable y válido para el actor involucrado en la secuencia de acciones.

Nombre del Caso de
Uso

Verbos

Son Acciones



Diagramas de Casos de Uso

Relaciones entre Casos de Uso:

1. Generalización.
2. Extensión.
3. Inclusión.



Diagramas de Casos de Uso

1. Generalización:

- Relación que define la especialización de un caso de uso.
- Los casos de uso **abstractos** describirán las partes similares y no podrán ser instanciados independientemente.
- Los casos de uso **concretos** describirán el comportamiento específico.

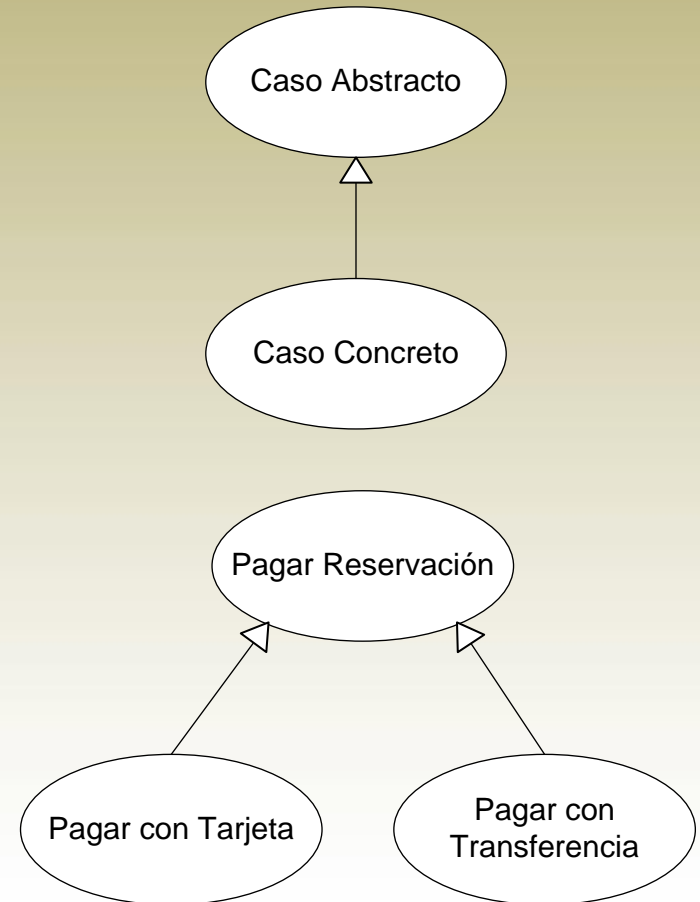
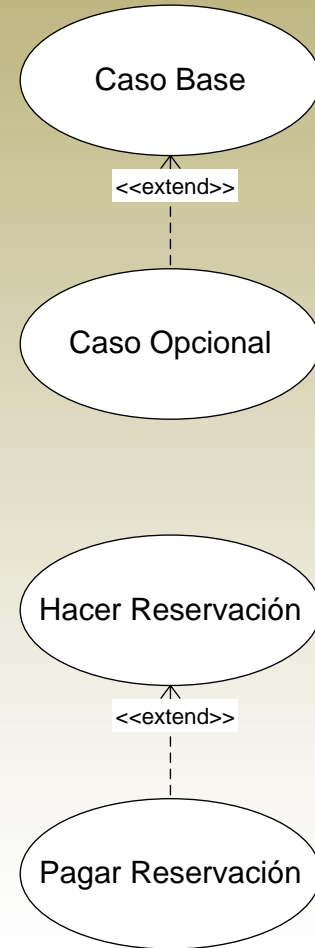




Diagrama de Casos de Uso

2. Extensión: (<<extend>>)

- Especifica como un caso de uso puede insertarse en otro para extender la funcionalidad de un caso de uso base.
- *El Caso Opcional es una extensión del Caso Base:*
Una instancia del caso de uso **Base** *puede incluir* el comportamiento especificado por el Caso **Opcional**.

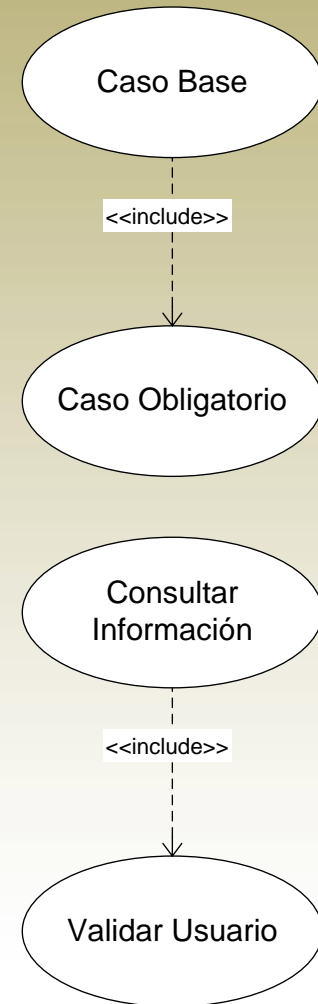




Diagramas de Casos de Uso

3. Inclusión: (<<include>>)

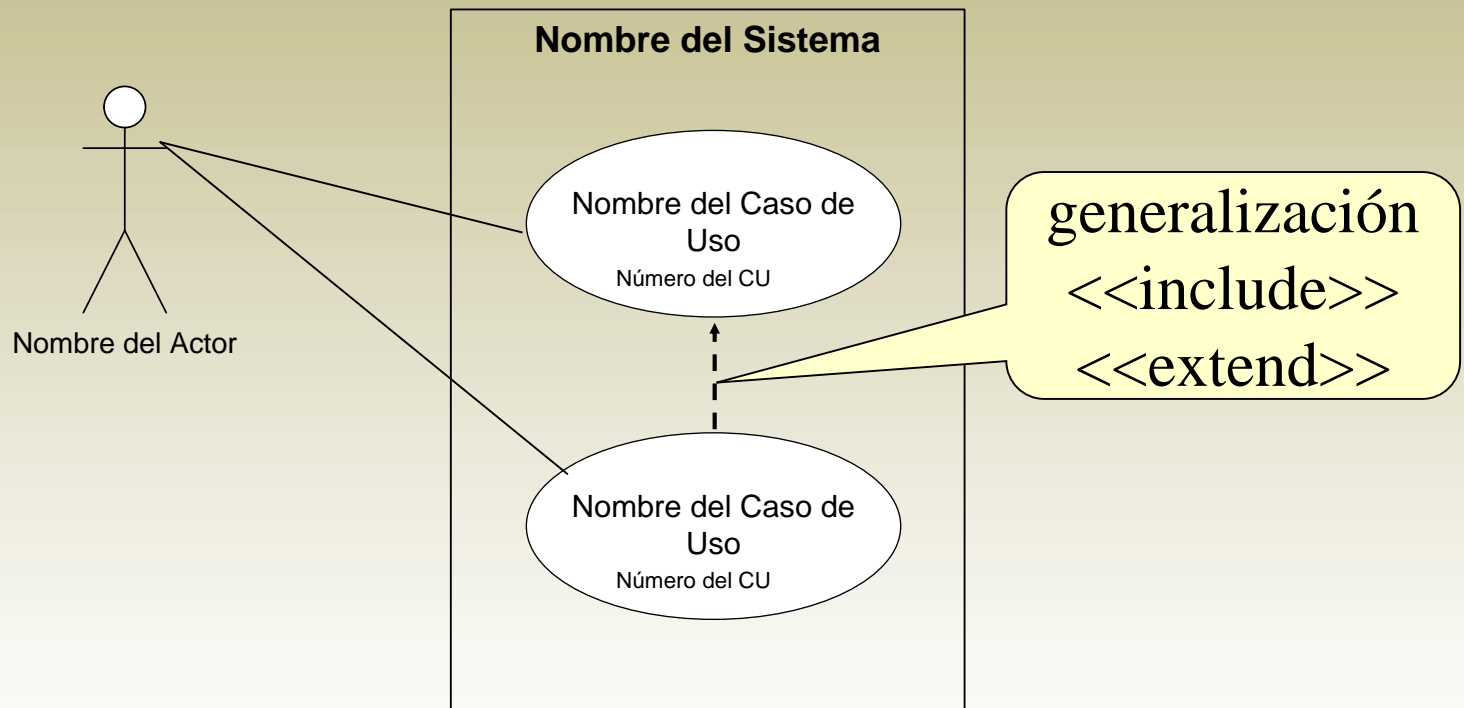
- La inclusión define como un caso de uso es parte obligatoria de un caso de uso base.
- *Un Caso Base incluye un Caso Obligatorio:*
Una instancia de un caso base **siempre** incluye el comportamiento especificado por un caso de uso obligatorio.





Diagramas de Casos de Uso

- Notación

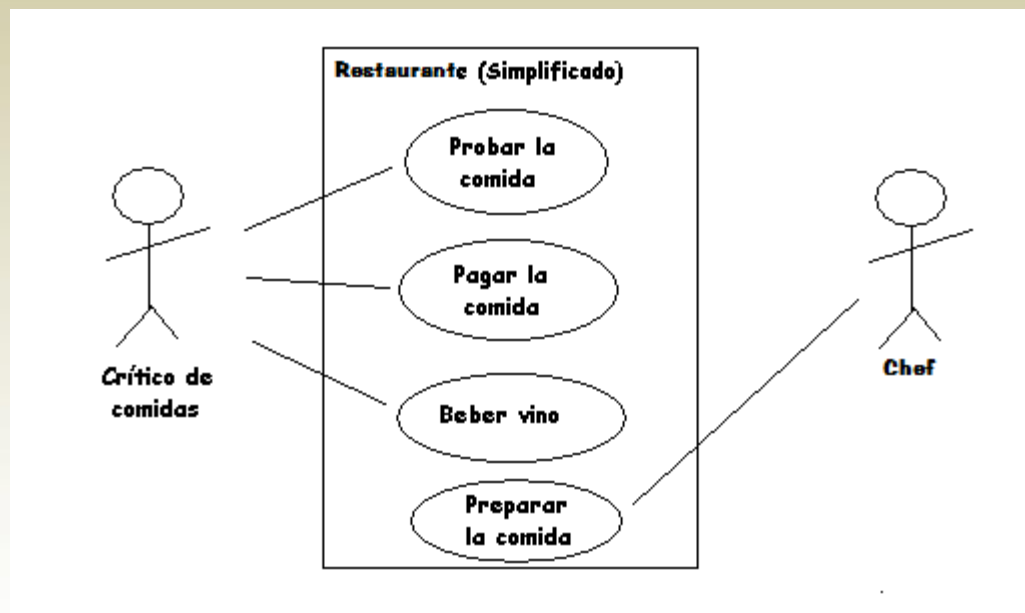




Diagramas de Casos de Uso

Caso de Uso

- Ejemplo:





Diagramas de Casos de Uso

Especificación de un Caso de Uso: (Documentación)

- Nombre del Caso de Uso.
- Actores.
- Propósito.
- Precondiciones.
- Flujo de Eventos Principal.
- Sub Flujos.
- Excepciones.
- Postcondiciones.



Diagramas de Actividades

- El **diagrama de actividades** define la lógica de los procedimientos, los procesos del negocio y flujos de trabajo del sistema.
- Un diagrama de actividades demuestra la *serie de actividades* que deben ser realizadas en un caso de uso, así como las distintas rutas que pueden irse desencadenando en el caso de uso.



Diagramas de Actividades

- **Actividad:**

Representa una acción que será realizada por el sistema.

Nombre de la Actividad

- Actividad *Inicial* ●

- Actividad *Final* ●

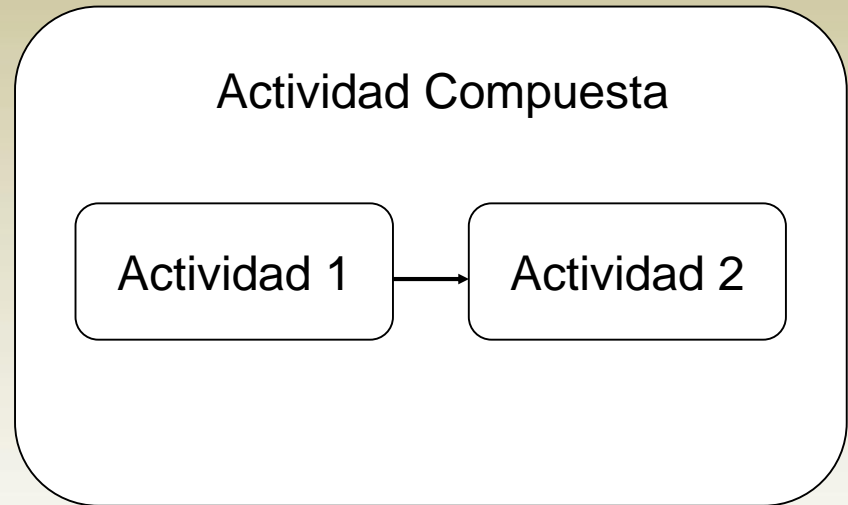
Verificar Password del Usuario



Diagrama de Actividades

- Sub Actividad:

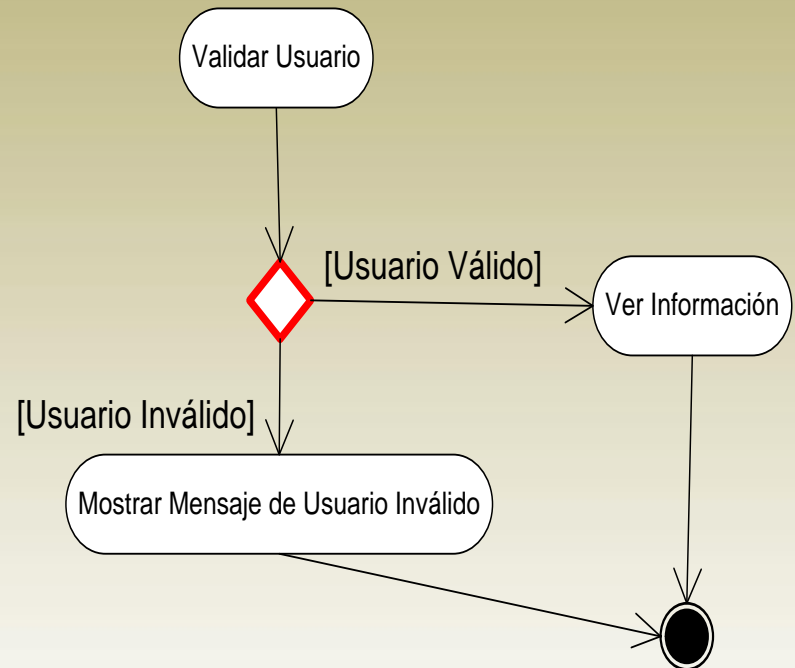
Una acción puede ser descompuesta en varias actividades.





Diagramas de Actividades

- **Ramificación (Branch):**
 - Una ramificación surge cuando existe la posibilidad que ocurra más de una transición (resultado) al terminar determinada actividad.
 - Este elemento es representado a través de un rombo.

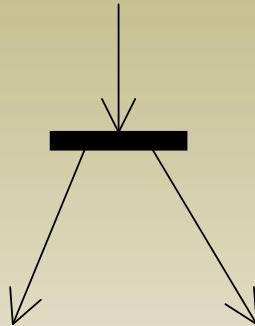




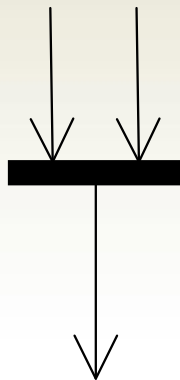
Diagramas de Actividades

- **Especificaciones Join**

- *División*



- *Unión*





Diagramas de Actividades

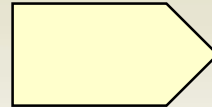
- **Señales**

Algunas acciones responden a señales

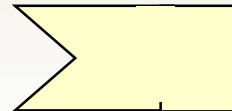
- Señales de tiempo



- Envío de señal



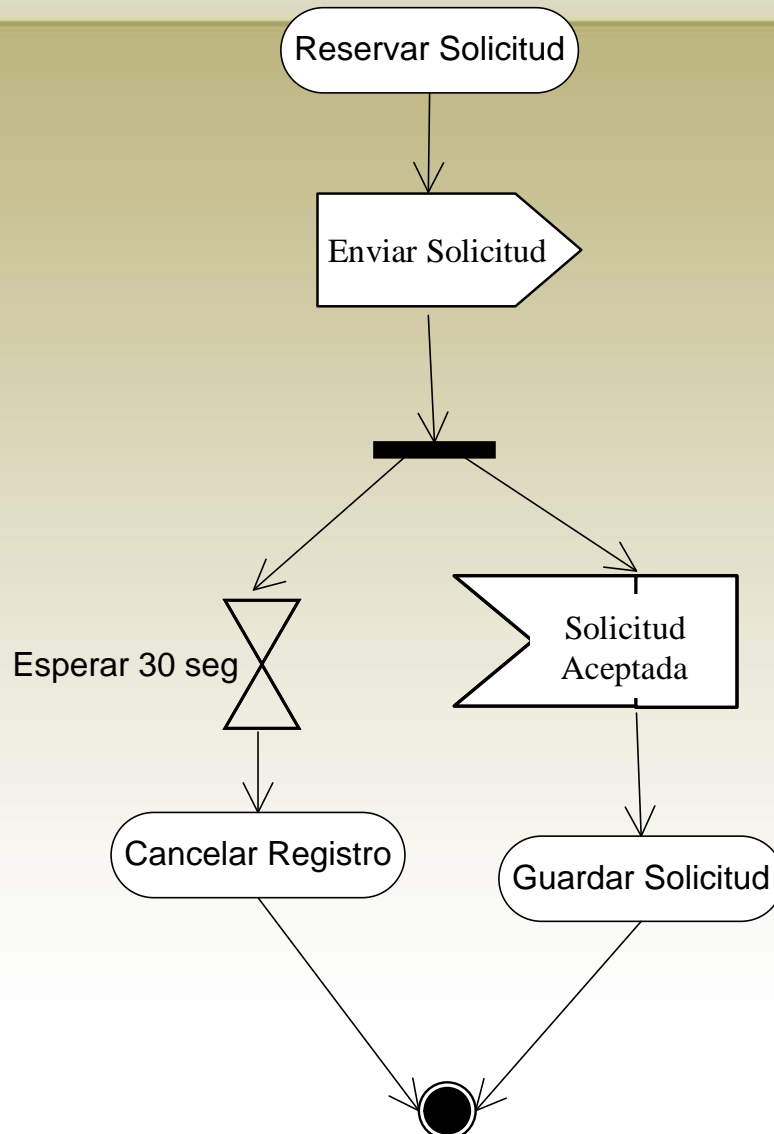
- Recepción de señal





Diagramas de Actividades

- Señales
 - Ejemplo:

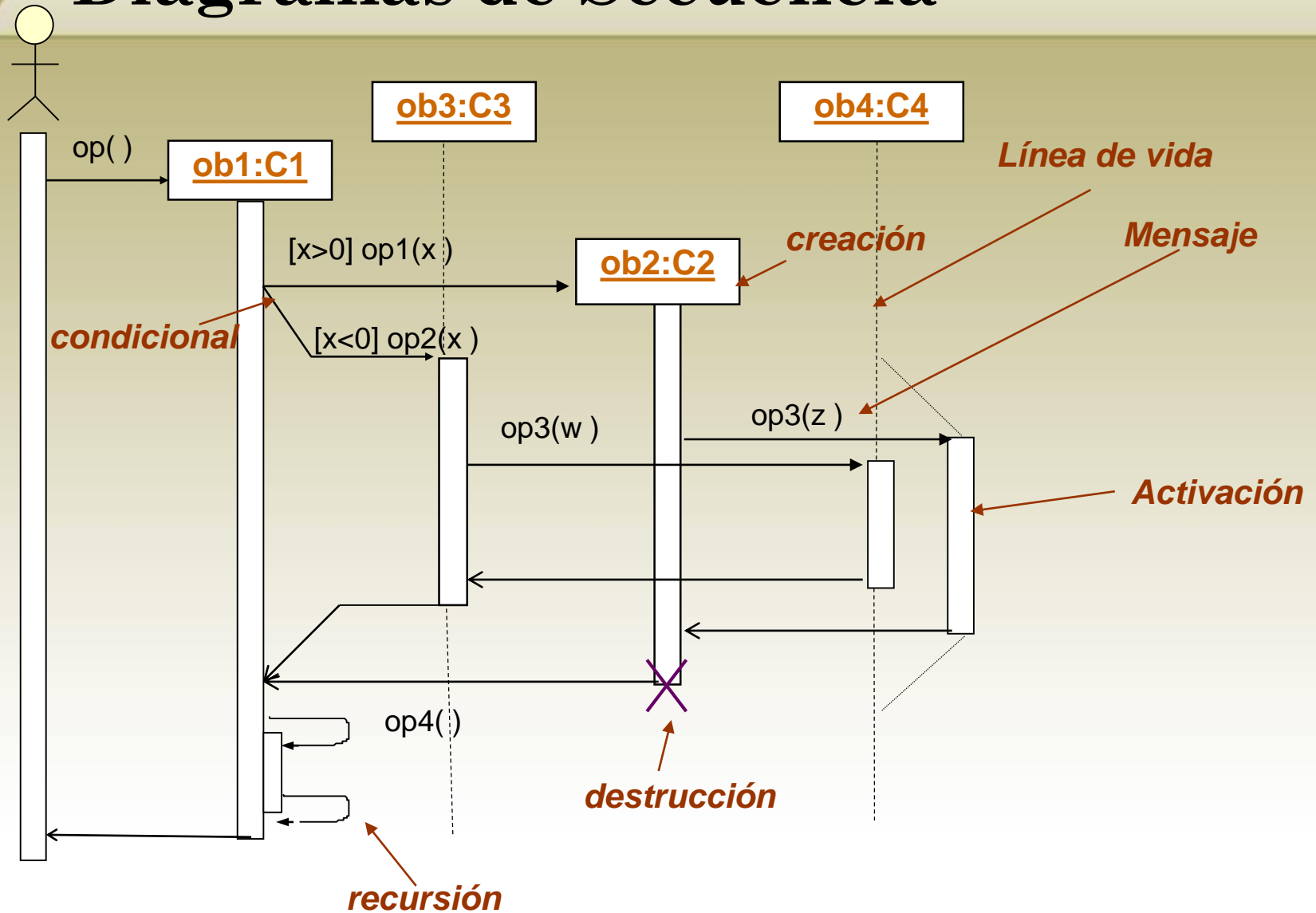




Diagramas de Secuencia

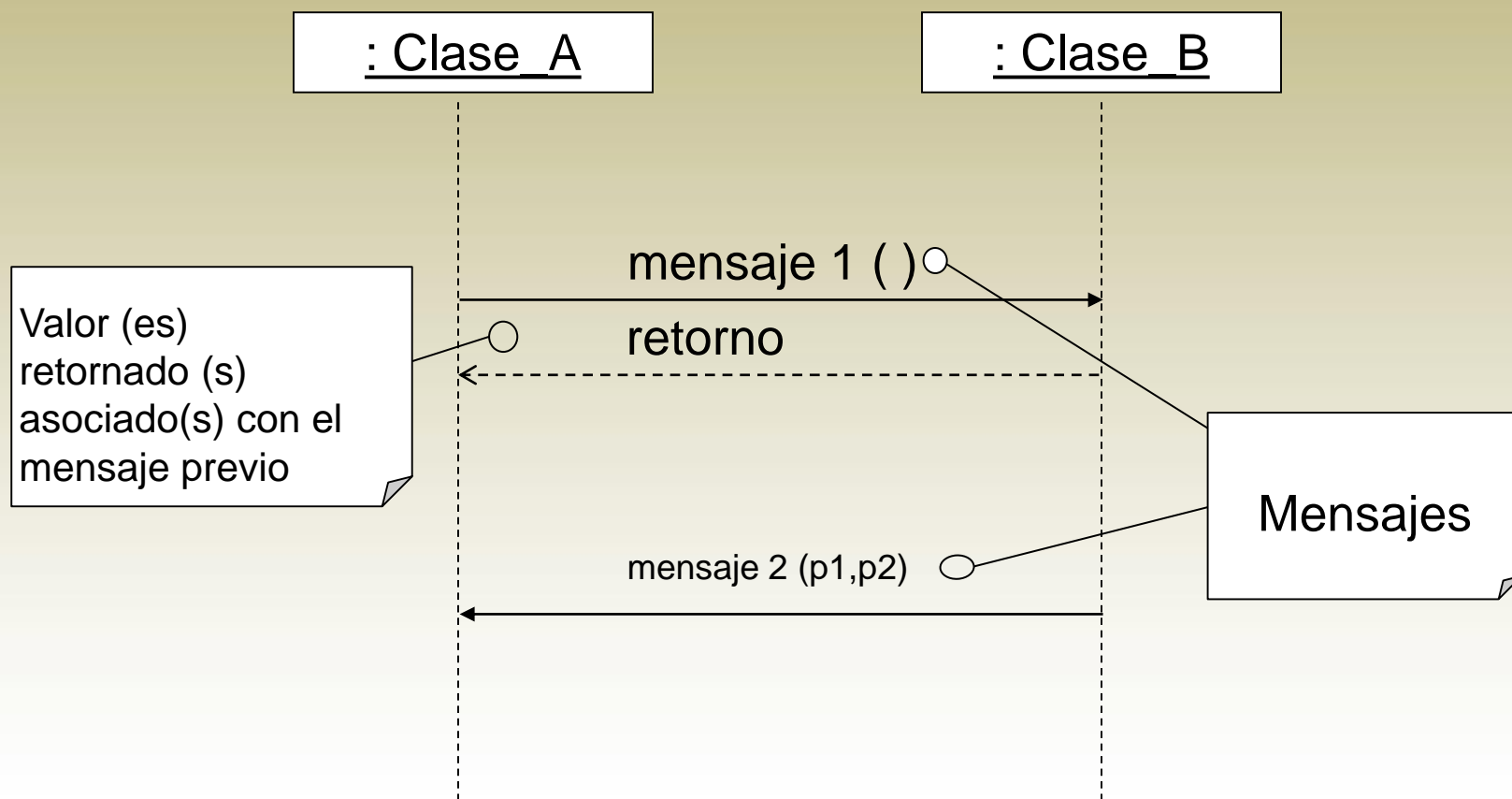
- Los diagramas de secuencia describen como **colaboran y se comunican (interacción)** los objetos del sistema.
- Muestra los objetos que participan en una interacción, el intercambio de mensajes y su ordenamiento en el tiempo.
- Un diagrama de secuencia es una representación que muestra, para un escenario de un caso de uso, los *eventos* que generan los actores, su *orden* y posibles eventos internos en el sistema

Diagramas de Secuencia





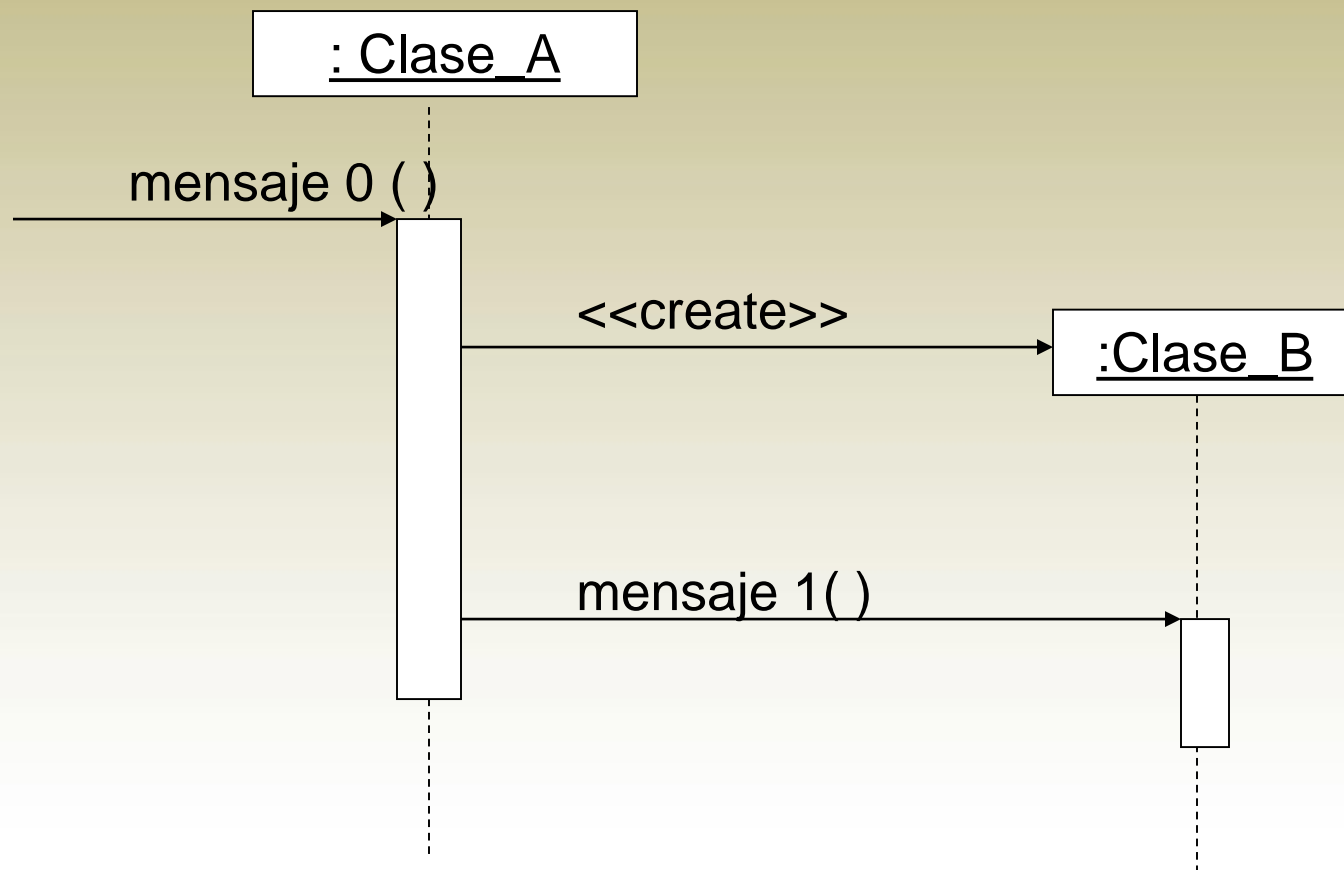
Diagramas de Secuencia





Diagramas de Secuencia

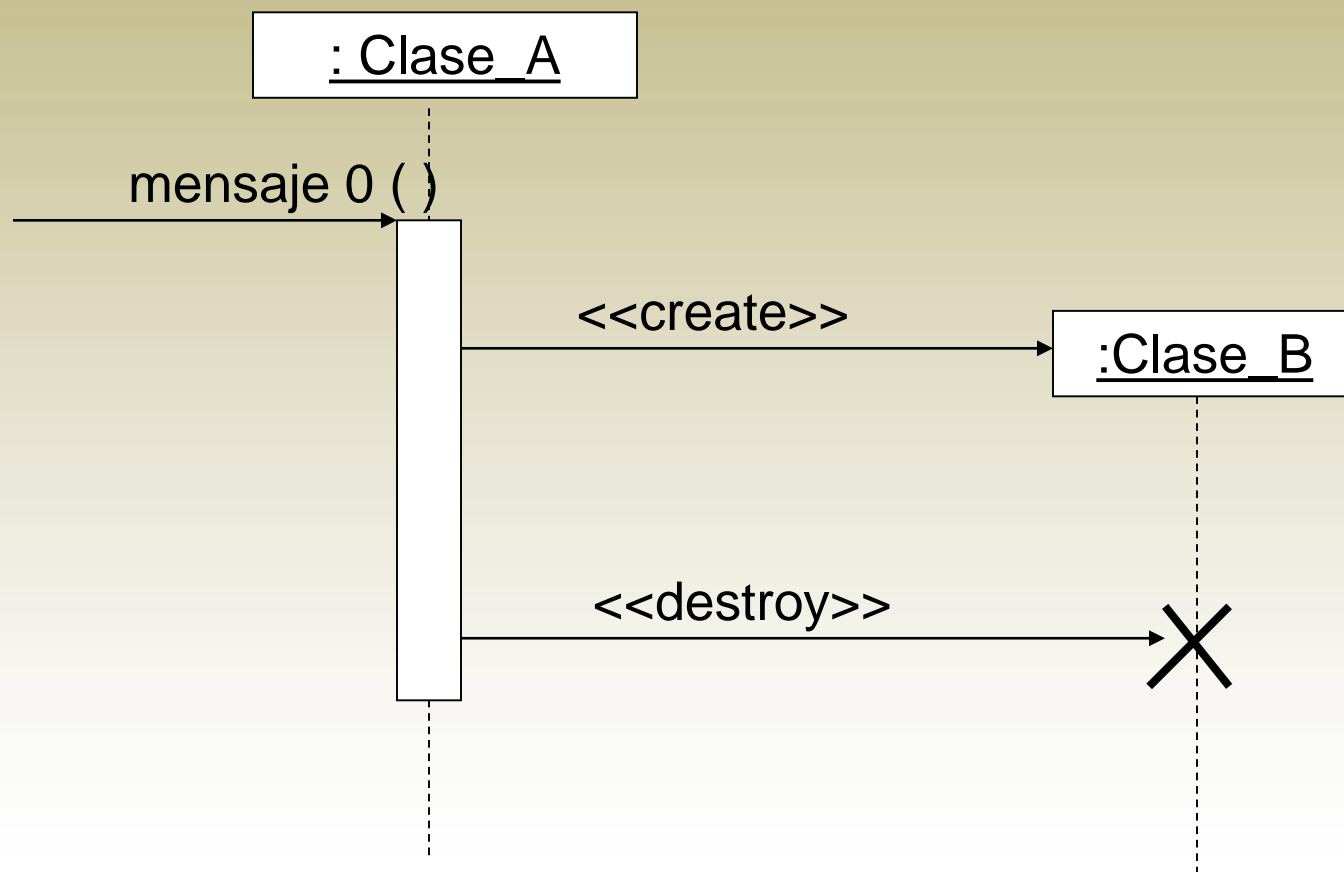
- Crear Objetos: **<<create>>**





Diagramas de Secuencia

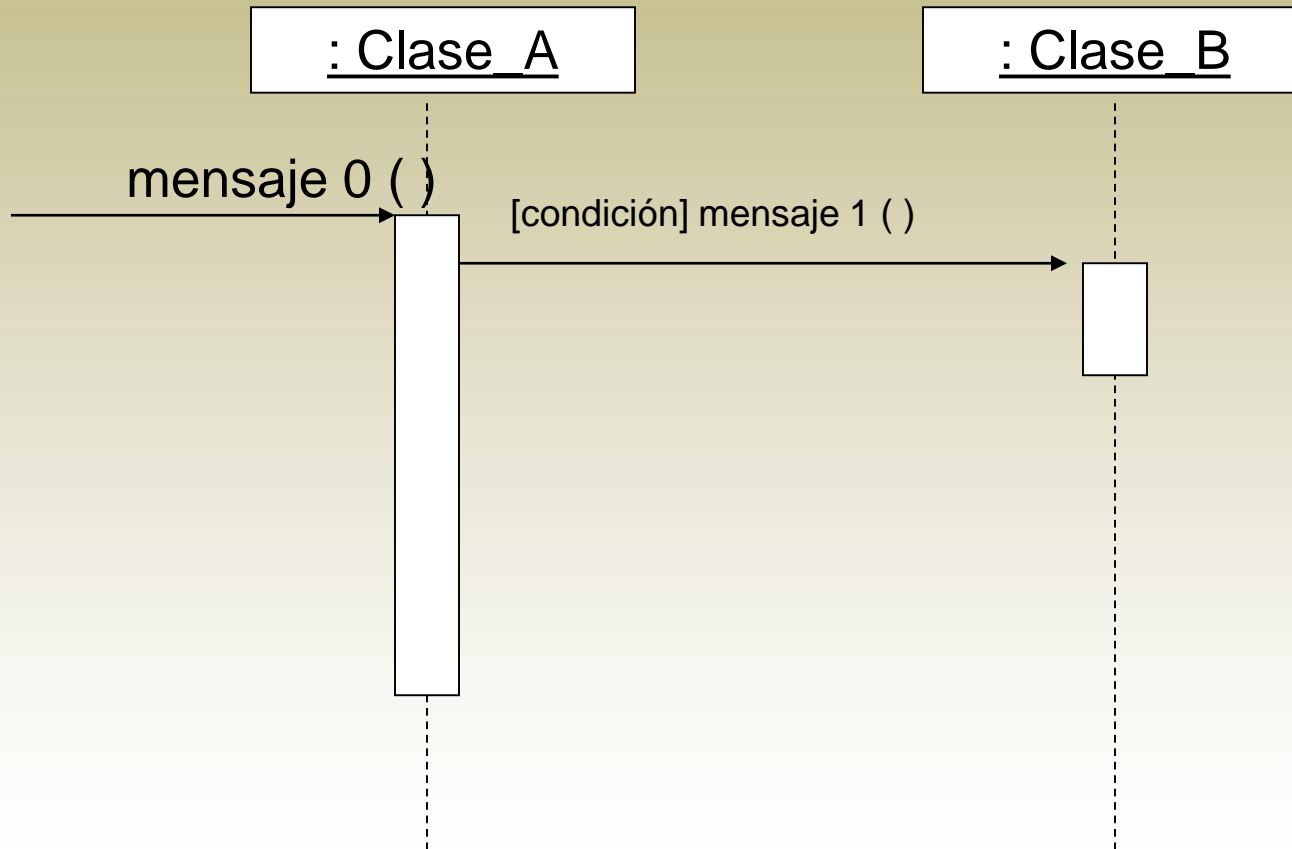
- Destruir Objetos: **<<destroy>>**





Diagramas de Secuencia

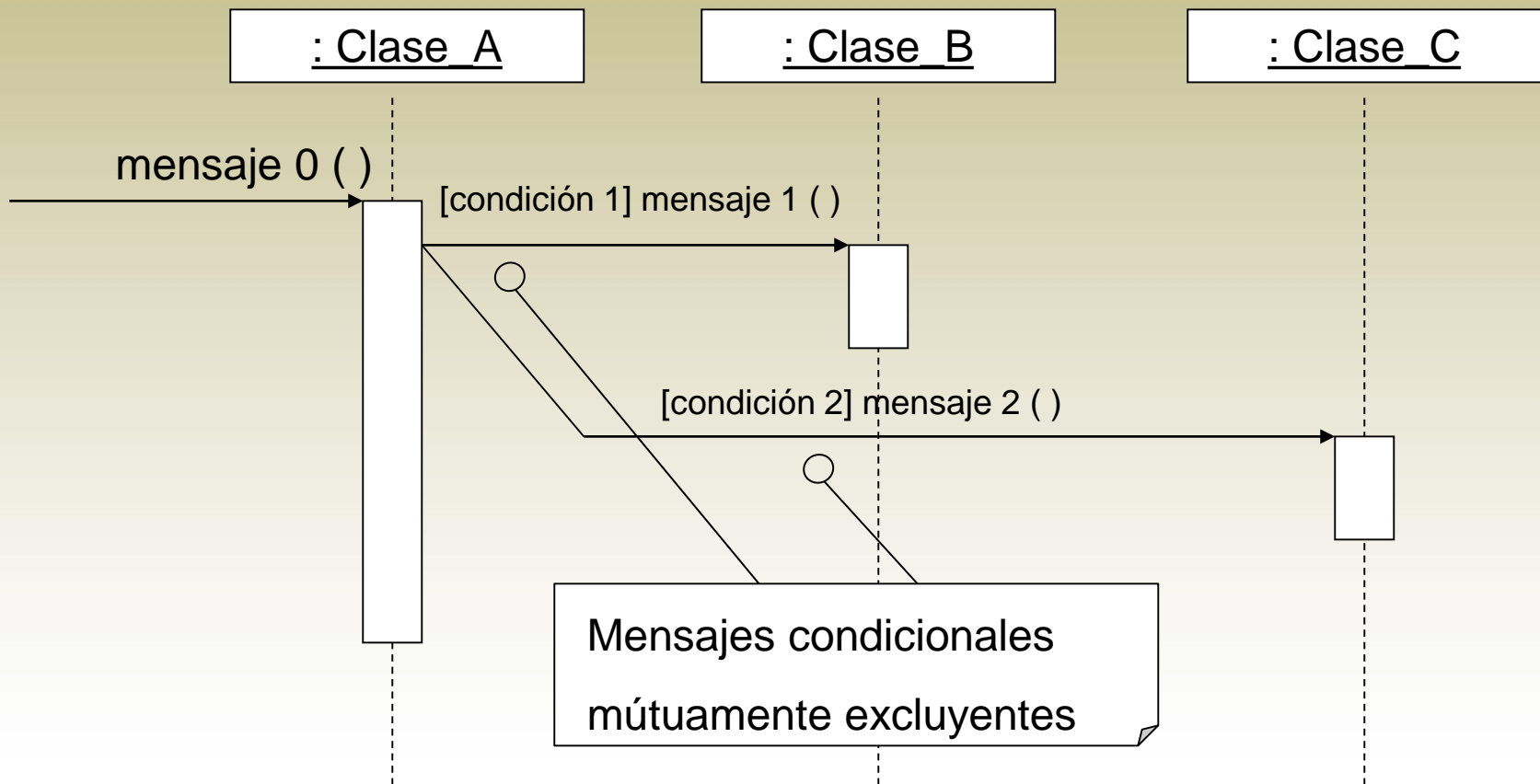
- Mensajes Condicionales





Diagramas de Secuencia

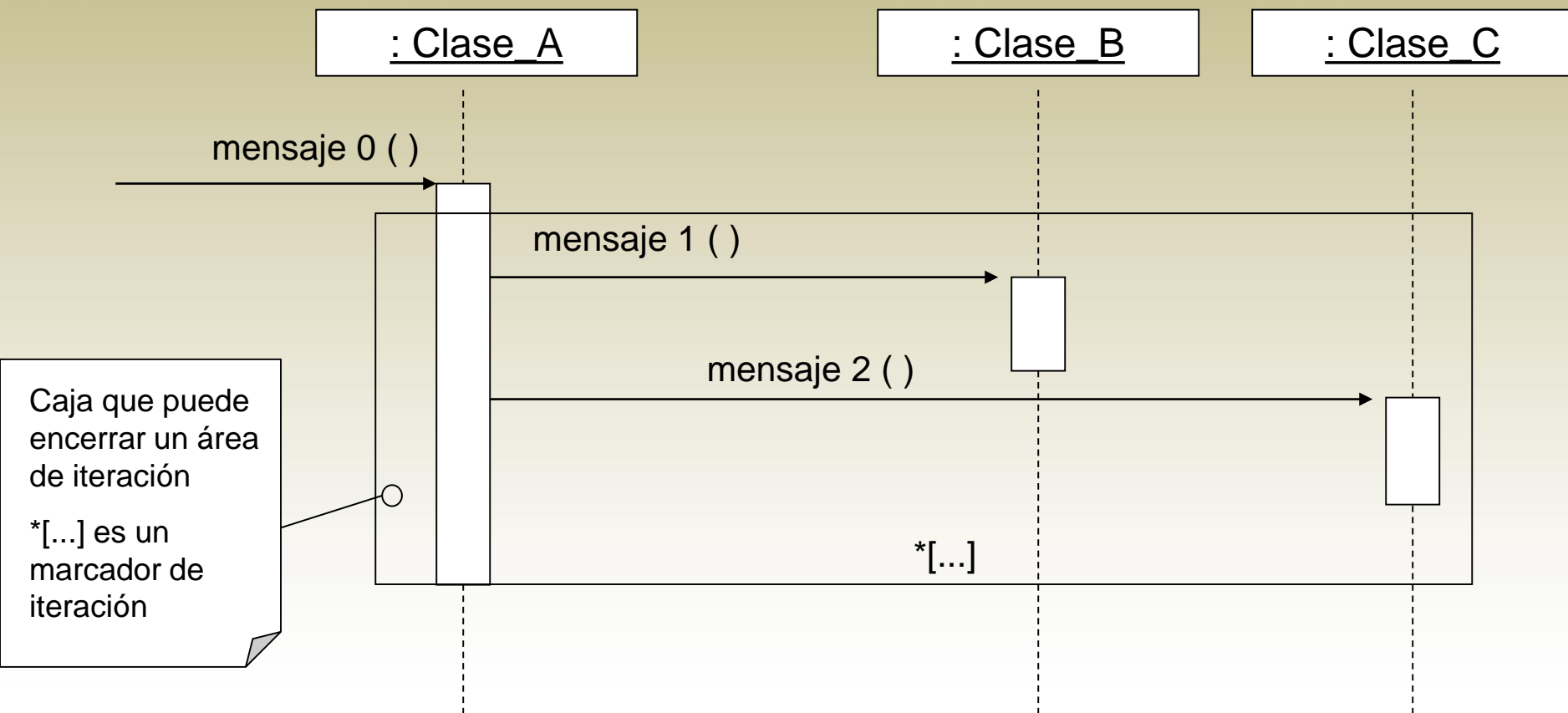
- Mensajes Condicionales Excluyentes





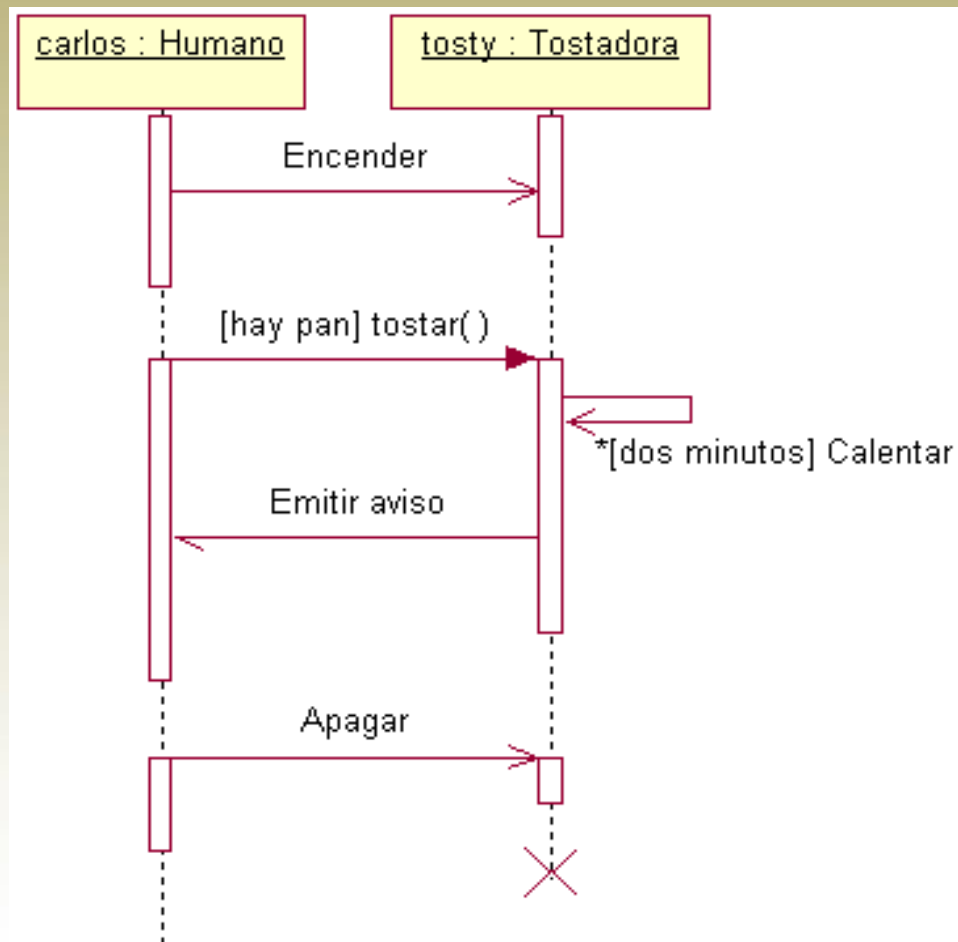
Diagramas de Secuencia

- Caja de Iteración:





Diagramas de Secuencia



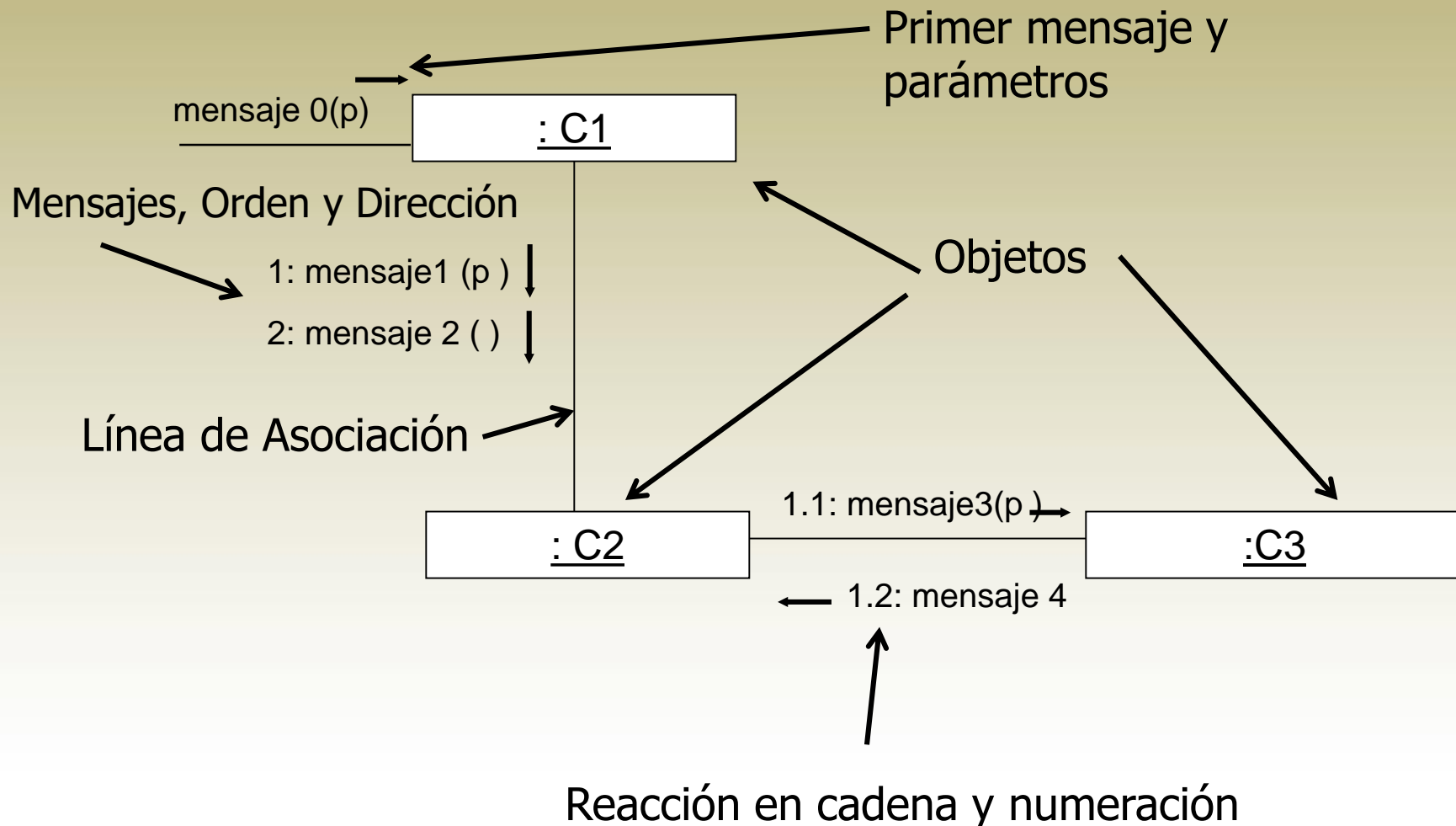


Diagramas de Colaboración

- Un Diagrama de Colaboración describe la **interacción** entre los objetos, numerando la secuencia de mensajes.



Diagramas de Colaboración





Diagramas de Colaboración

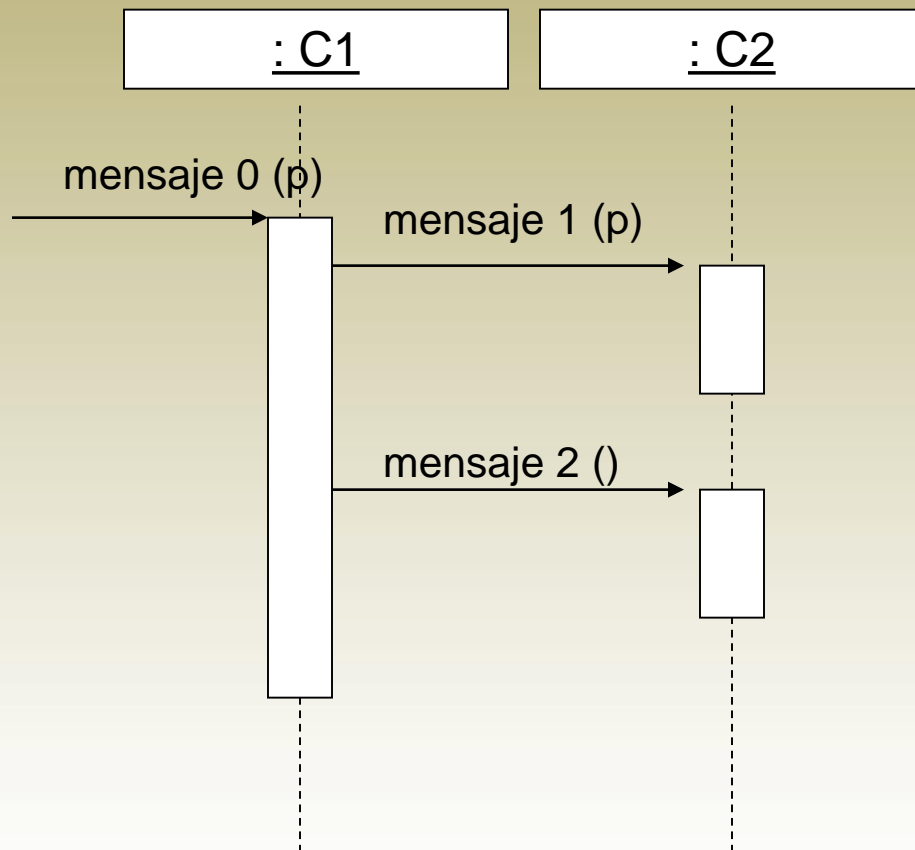


Diagrama de Secuencia

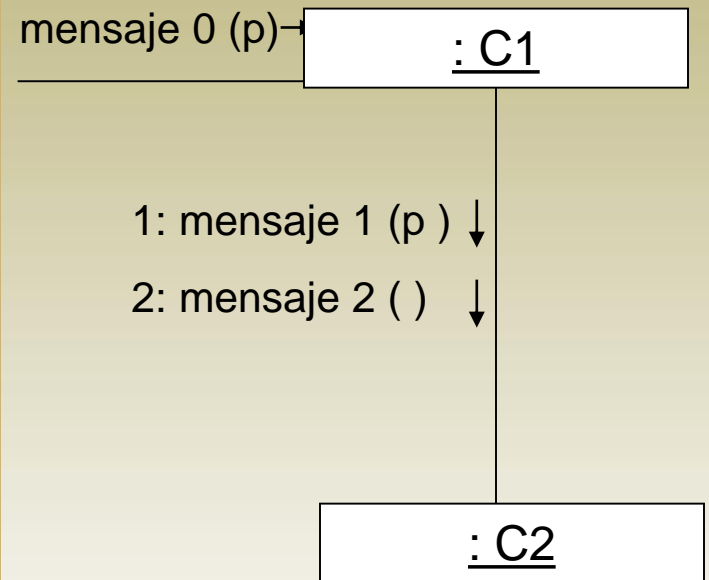
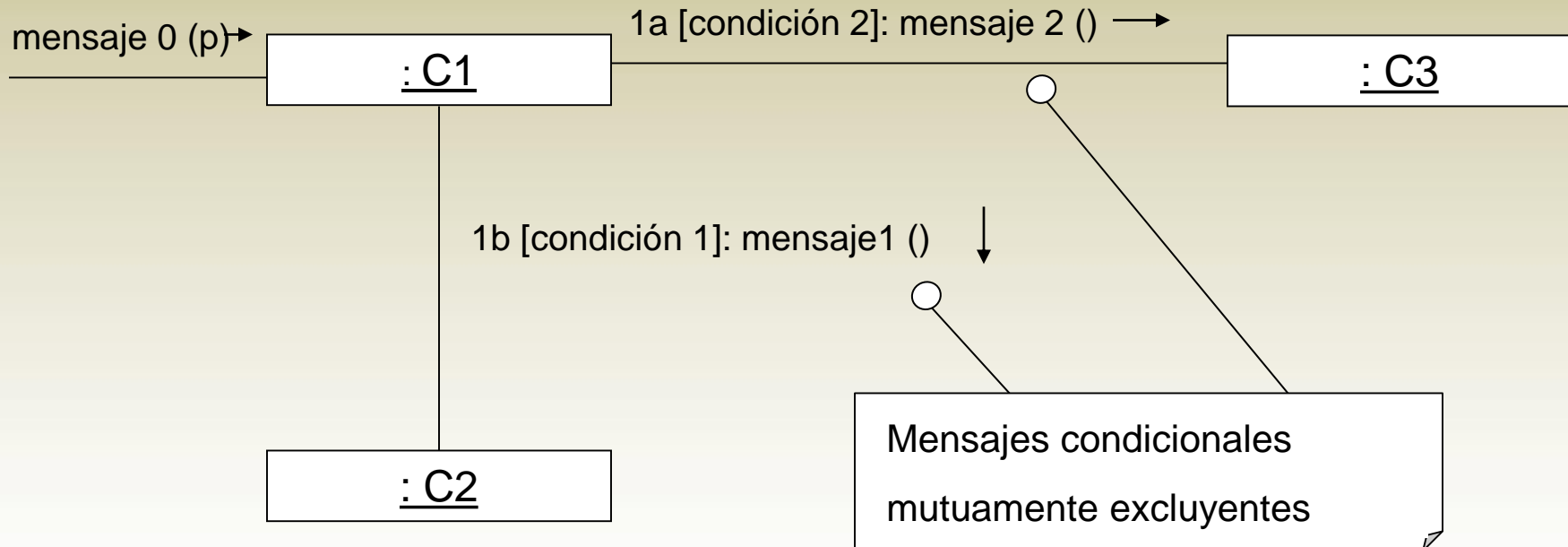


Diagrama de Colaboración



Diagramas de Colaboración

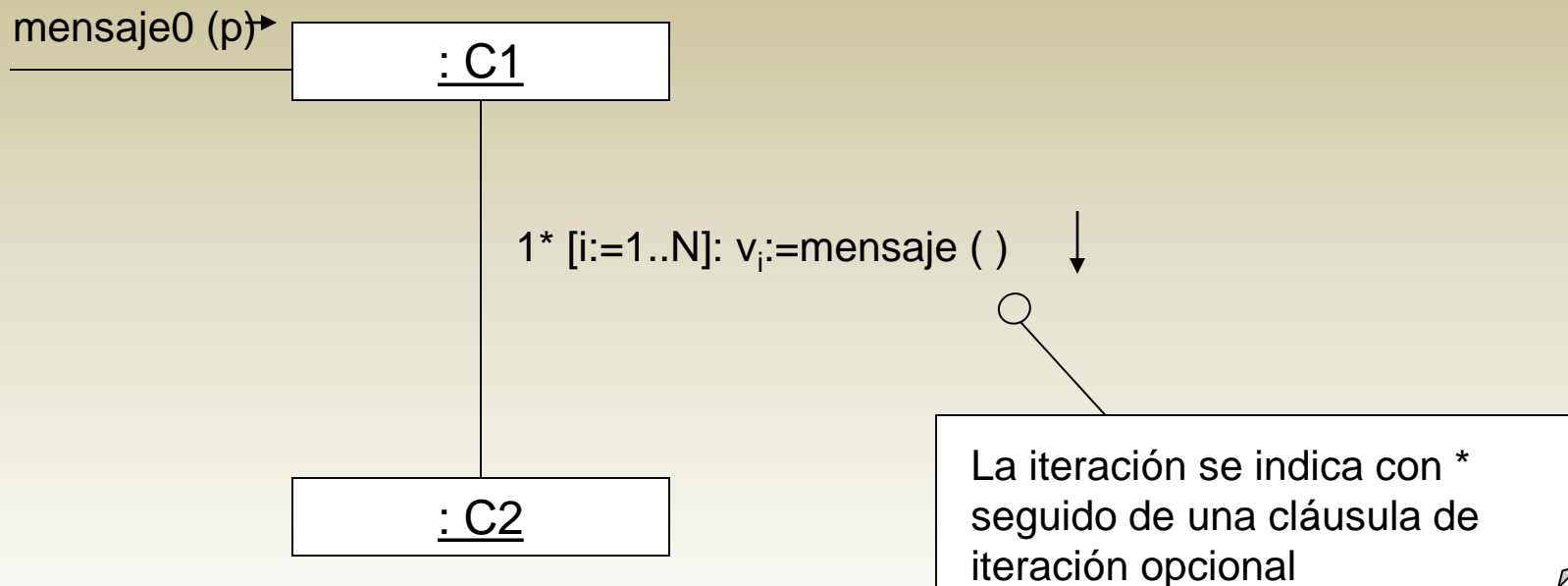
- Mensajes Excluyentes





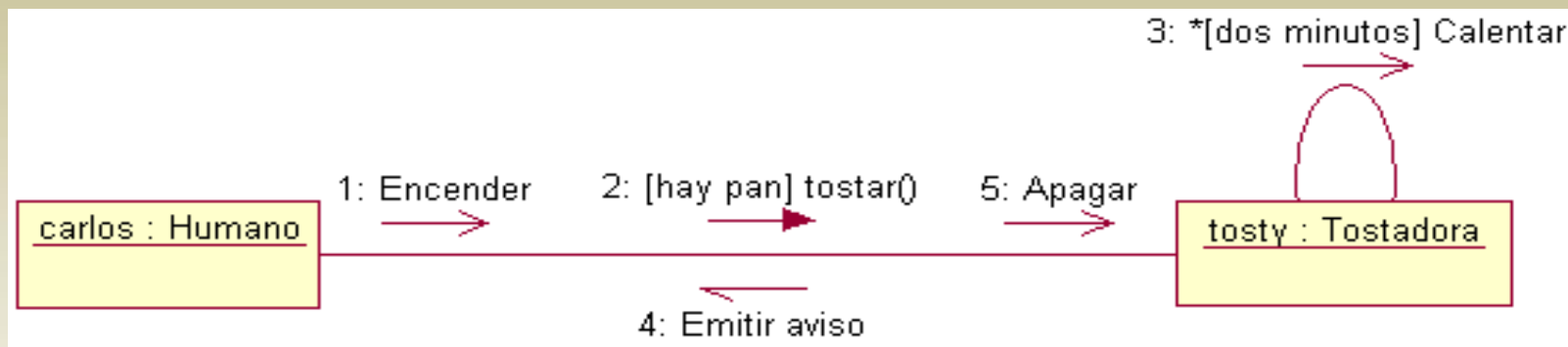
Diagramas de Colaboración

- Iteración





Diagramas de Colaboración





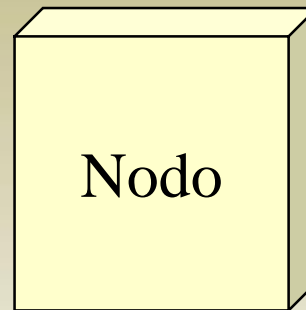
Diagramas de Despliegue

- Un diagrama de despliegue muestra las *relaciones físicas* entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).
- En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa el software en el hardware que lo contiene.

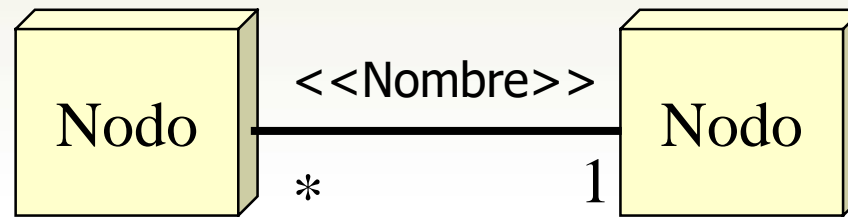


Diagramas de Despliegue

- **Nodo:** Elemento donde se ejecutan los componentes, representan el despliegue físico de estos componentes.

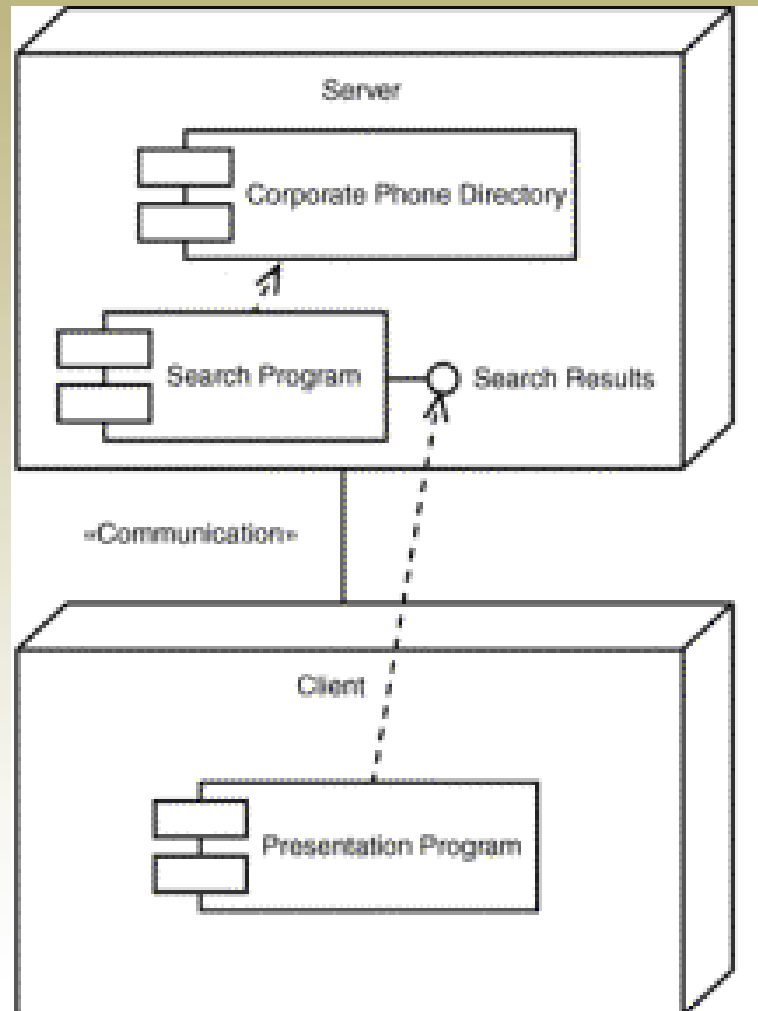


- **Asociación:** Representa el tipo relación que soporta la comunicación entre nodos





Diagramas de Despliegue





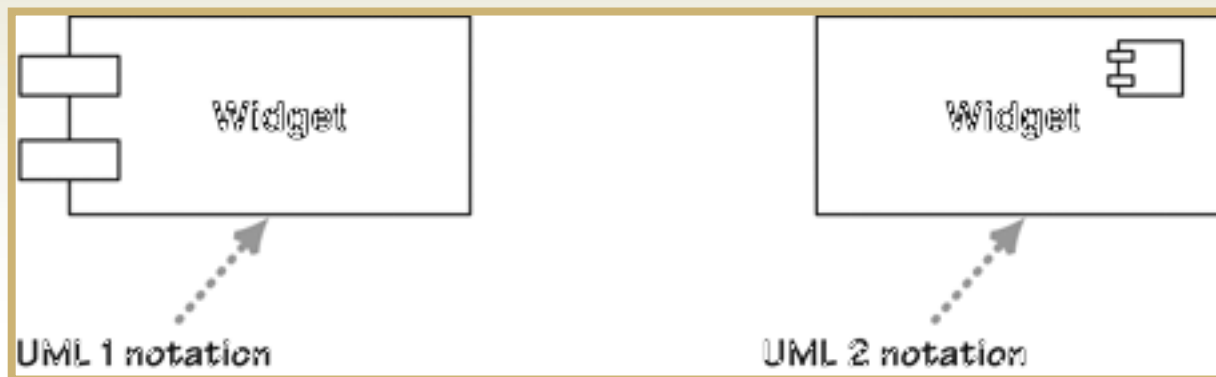
Diagramas de Componentes

- El diagrama de componentes muestra la relación entre componentes de *software*, sus dependencias, su comunicación, su ubicación y otras condiciones.
- Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes *software*, sean éstos componentes de código fuente, binarios o ejecutables.
- Define los módulos físicos del software y sus relaciones.



Diagramas de Componentes

- **Componente**
 - Es un tipo de contenedor.
 - Provee una vista encapsulada de las funcionalidades definidas en las clases.
 - Por ejemplo, un **paquete** en un diagrama de componentes representa una división física del sistema.

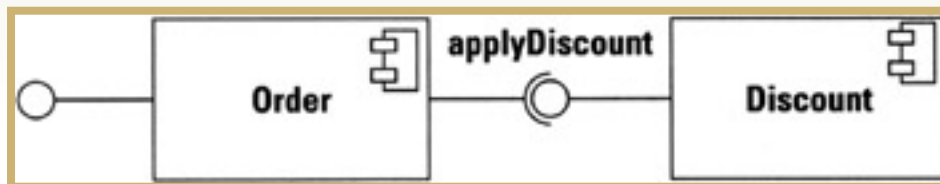
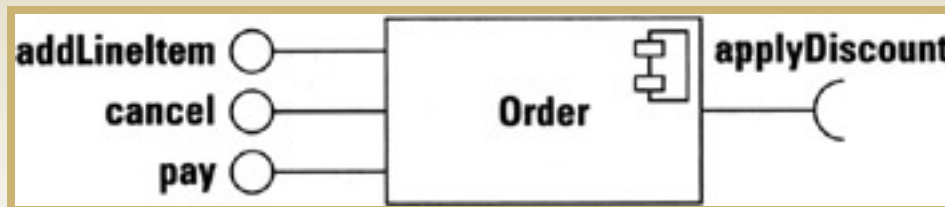




Diagramas de Componentes

- **Interfaces**

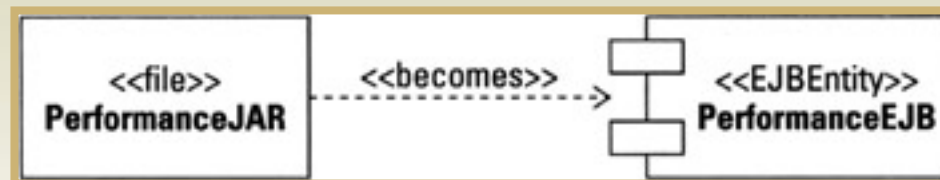
- Las interfaces son los **puntos visibles** de entrada o los servicios que un componente está ofreciendo y dejando disponibles a otros componentes de software y clases.
- Exponen funcionalidades para otros componentes y las requeridas de otros.





Diagramas de Componentes

- Dependencia de módulos
 - Abstrae la implementación de la interfaz e indica la dependencia entre módulo





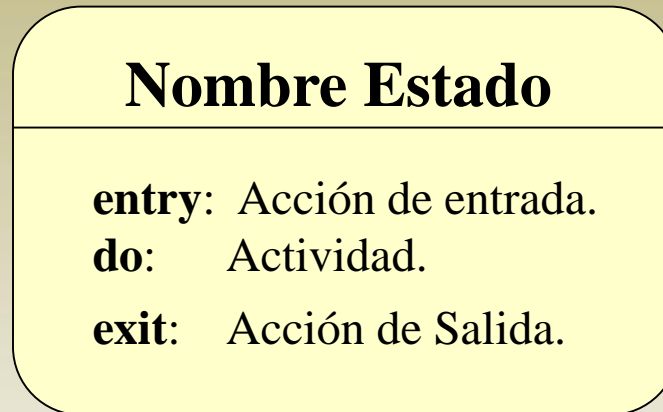
Diagramas de Estado

- Los diagramas de estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto.
- Los diagramas de estado ven a los objetos como *máquinas de estado* o autómatas finitos que pueden estar en un conjunto de estados finitos y que pueden cambiar su estado a través de un estímulo perteneciente a un conjunto finito.



Diagramas de Estados

- Estado:

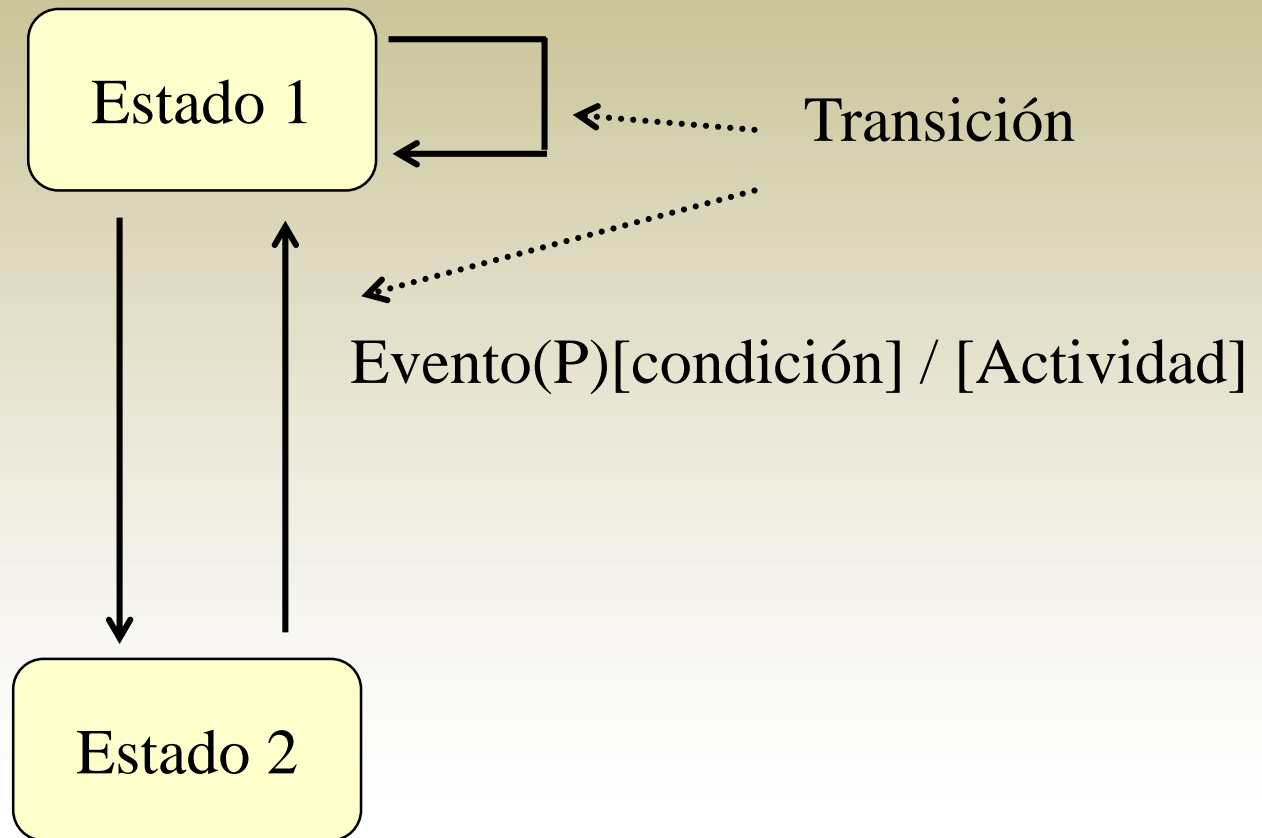


- Estado Inicial ●
- Estado Final ●



Diagramas de Estados

- Transición:





Diagramas de Estados

- Diagrama de Estados de un Libro:

