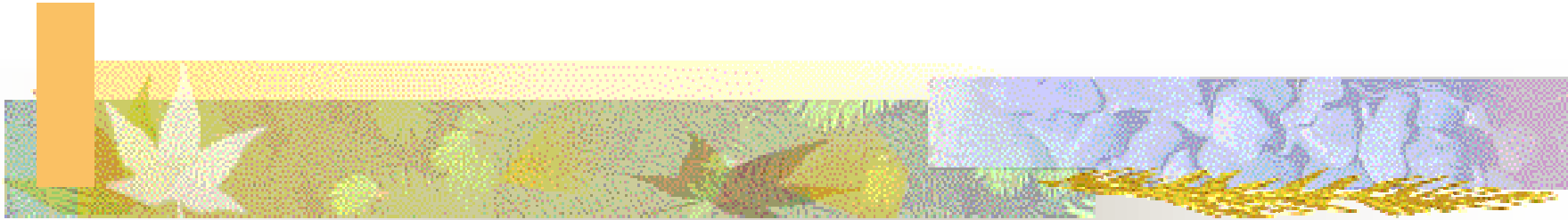


ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS CON UML

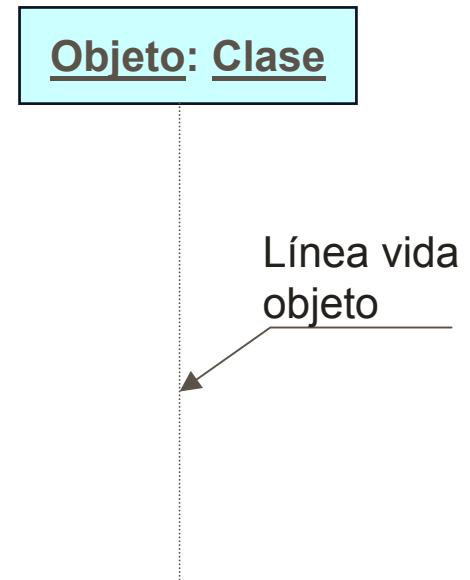
(Parte III)



Ing. Luis Zuloaga Rotta

Los Diagramas de Secuencia

- Muestran interacciones entre objetos según un punto de vista temporal.
- Representa una interacción entre objetos en base a la cronología de los envíos de mensajes.



Representación gráfica
de un objeto

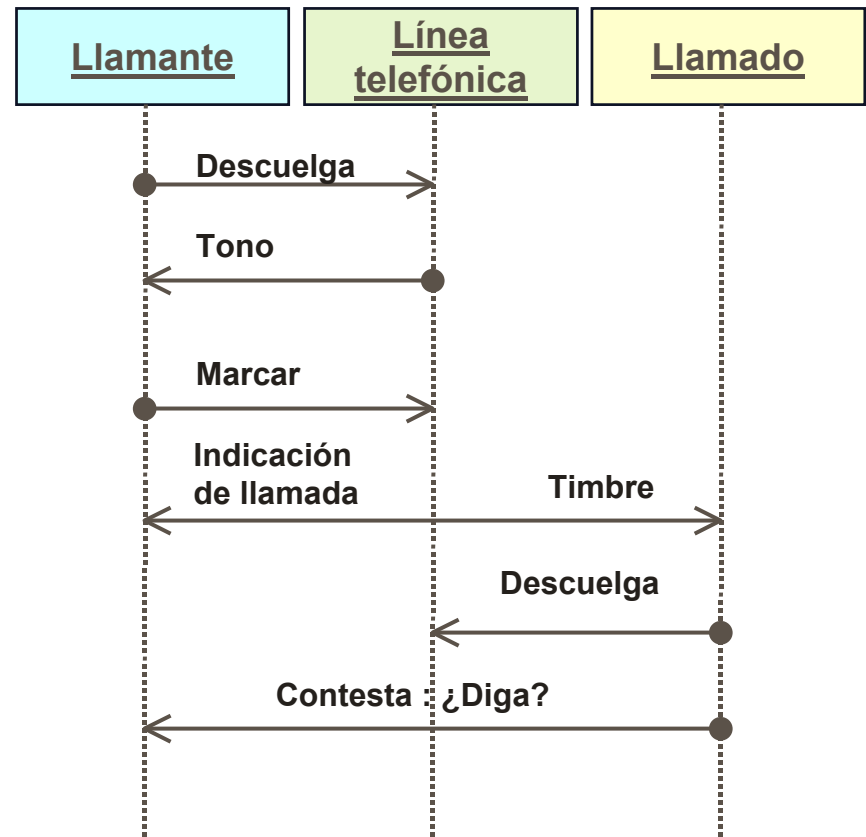


Mensajes entre objetos

- Los objetos se comunican intercambiando mensajes por medio de flechas horizontales, orientadas del emisor del mensaje hacia el destinatario.
- El orden de envío de los mensajes viene dado por la posición sobre el eje vertical.
- En modelado de objetos, los diagramas de secuencia se utilizan de dos maneras diferentes :
 - Para documentar los Casos de Uso
 - Para representar en forma mas precisa las interacciones entre objetos.

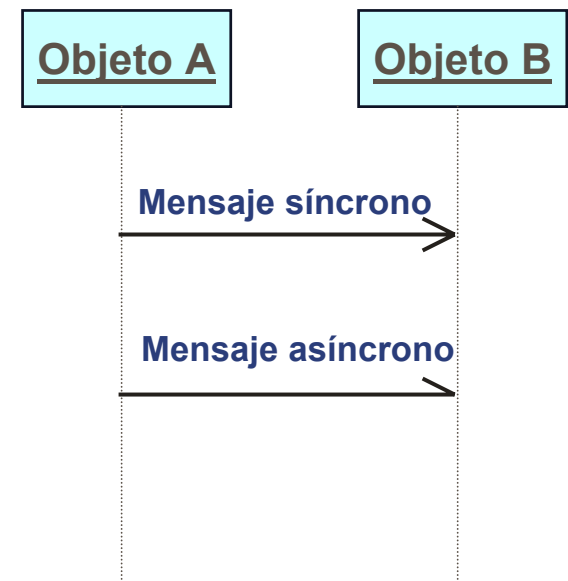
Documentando Casos de Uso

- Se concentra sobre la descripción de la interacción, a menudo en términos próximos al usuario y sin entrar en los detalles de la sincronización.
- La indicación que acompaña a las flechas corresponde entonces a eventos que ocurren en el ámbito de la aplicación. En este estadio del modelado, las flechas no corresponden aún a envíos de mensajes en el sentido de lenguajes de programación.

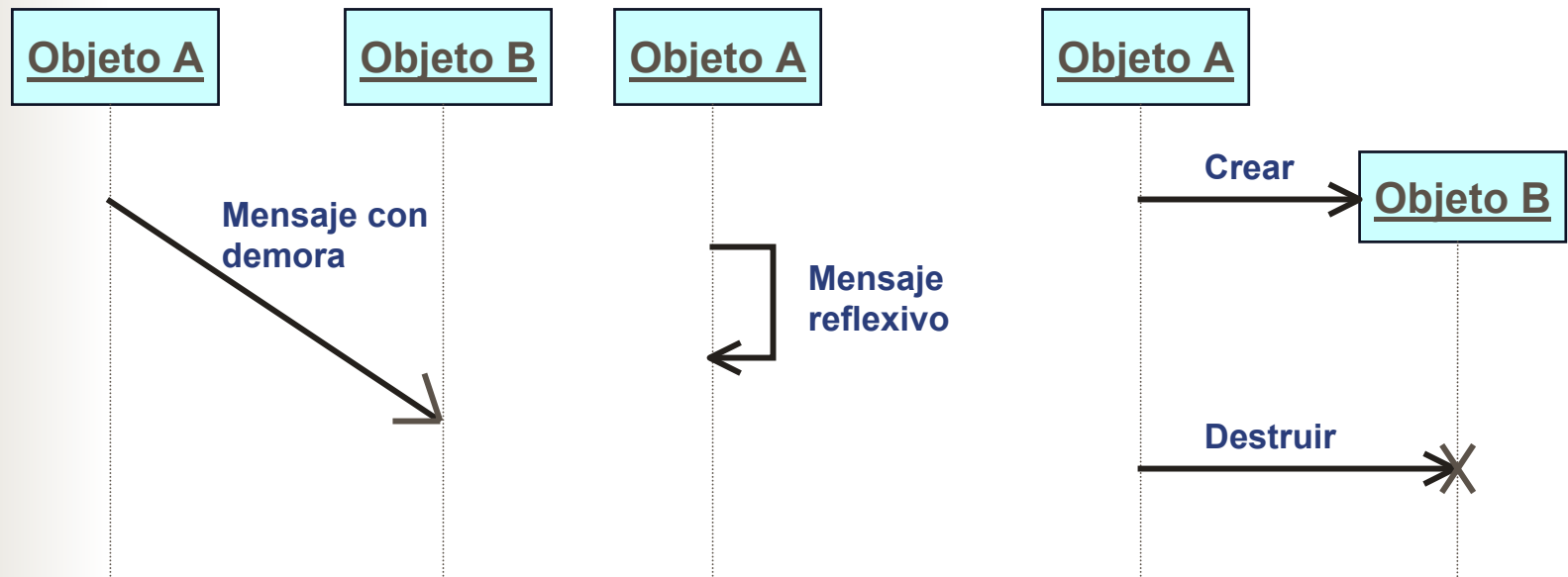


Los mensajes en la interacción entre objetos

- El concepto de mensaje unifica todas las formas de comunicación entre objetos, en particular la llamada de una operación, un evento discreto, la señal entre flujos de ejecución o la interrupción de hardware.
- Los diagramas de secuencia distinguen dos categorías de envíos de mensaje :
 - Síncronos, en los que el emisor está bloqueado y espera que el receptor haya terminado de tratar el mensaje.
 - Asíncronos, en los que el emisor no está bloqueado y puede continuar su ejecución.



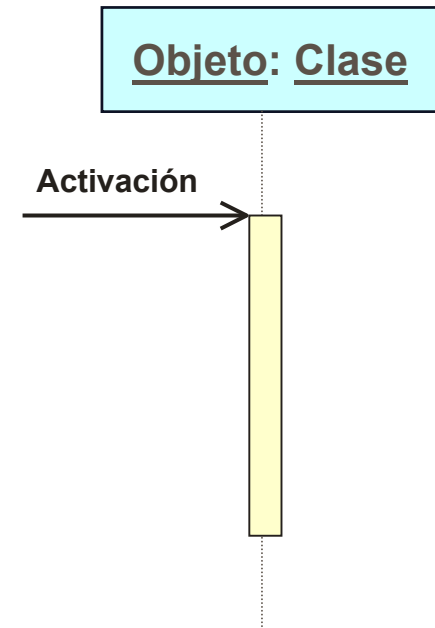
Representación de las interacciones



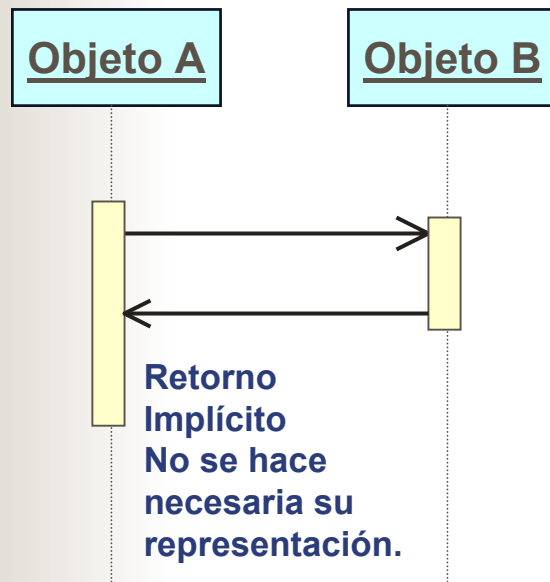
Representación de la creación y de la destrucción de objetos.

Periodos de actividad de los objetos

- Un periodo de actividad corresponde al tiempo durante el que un objeto efectúa una acción, bien directamente, o bien por medio de otro objeto que le sirve de subcontratado.
- Los periodos de actividad se representan por bandas rectangulares colocadas sobre las líneas de vida.
- El inicio y el fin de una banda corresponden, respectivamente, al inicio y fin de un periodo de actividad.



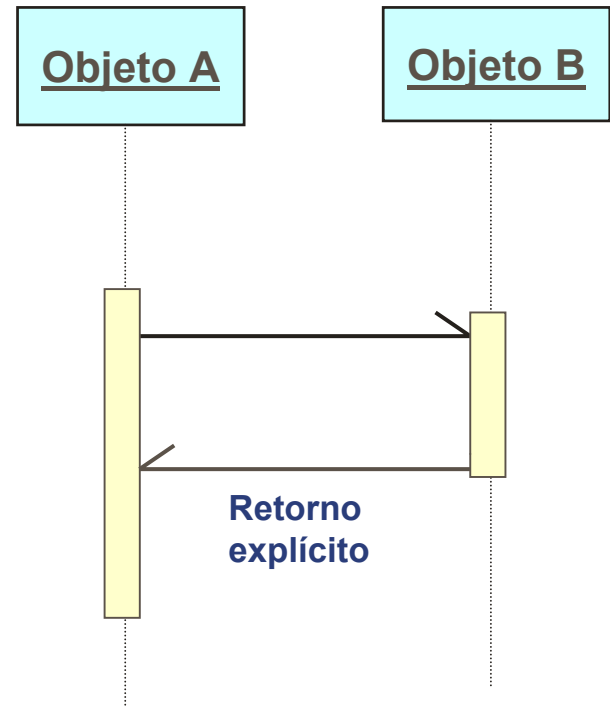
Activación de un objeto mediante un mensaje síncrono



- El diagrama siguiente muestra el caso de un objeto **A** que activa otro objeto **B**, tal que el período de actividad del objeto **A** recubre el periodo de actividad del objeto **B**.
- En caso de una llamada de operación, el flujo de ejecución es pasada por el objeto **A** al objeto **B**. El objeto **A** queda bloqueado hasta que el objeto **B** le devuelve una respuesta.
- *En el caso de una llamada de operación, y mas generalmente en el caso de envíos síncronos, el retorno al fin de la ejecución de la operación es implícito, no es necesario representarlo en los diagramas.*

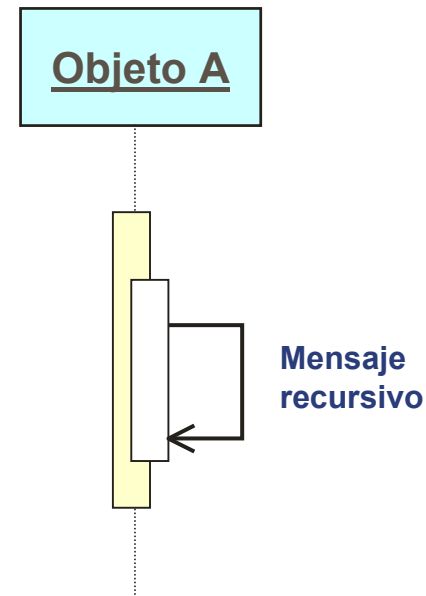
Activación de un objeto mediante un mensaje asíncrono

- En el caso de envío asíncronos, el retorno debe materializarse cuando existe.
- El diagrama de la derecha muestra un objeto **B** inicialmente activado por el objeto **A**, que devuelve un mensaje al objeto A antes de cesar su ejecución.

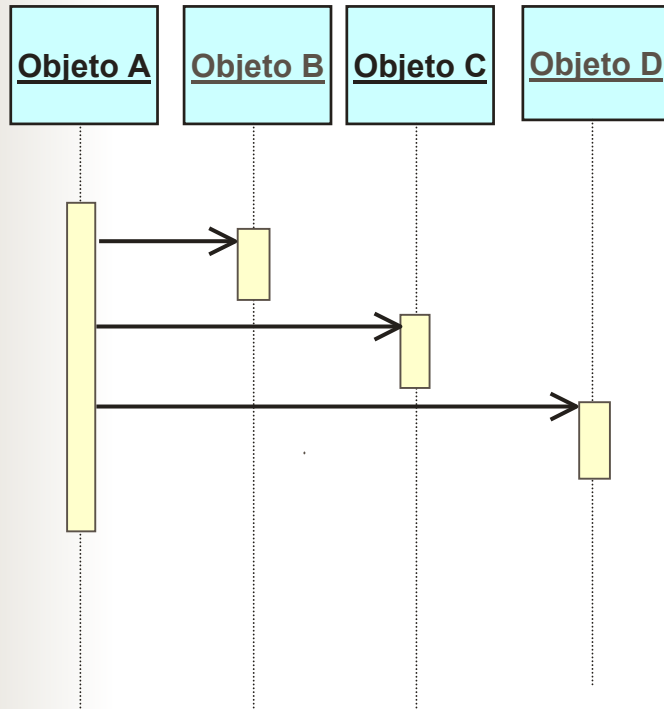


Los mensajes recursivos

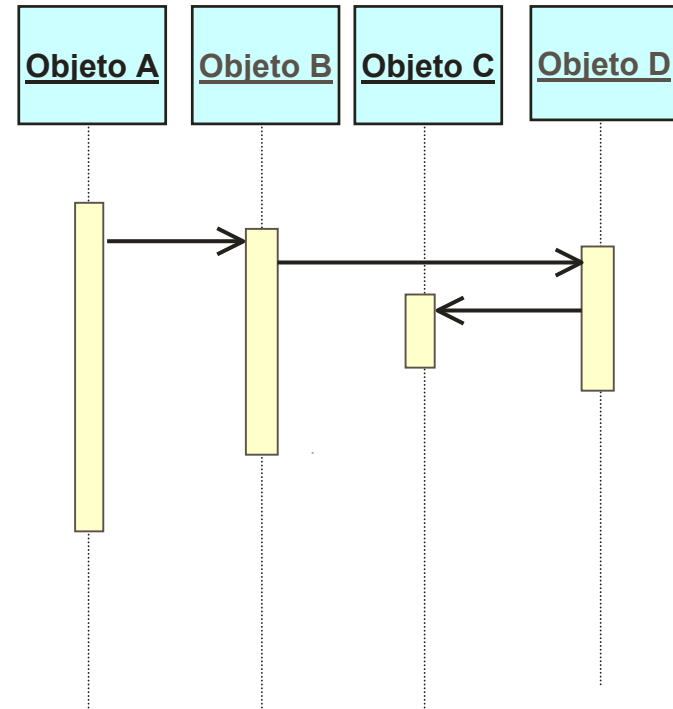
- El caso particular de envíos de mensajes recursivos se representa por un desdoblamiento de la banda rectangular.



Estructuras de control



Control centralizado

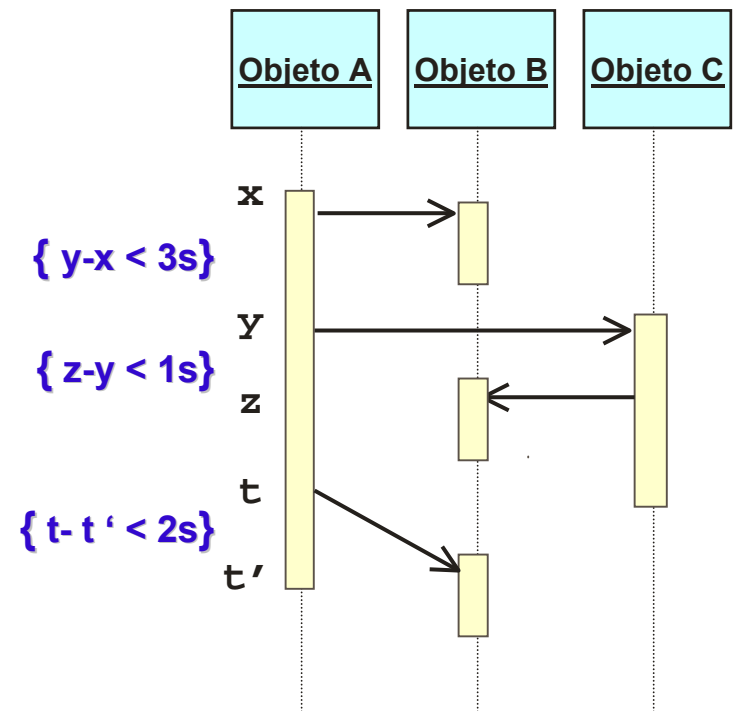


Control descentralizado

La forma de los diagramas de secuencia es el reflejo del modo de control de la interacción.

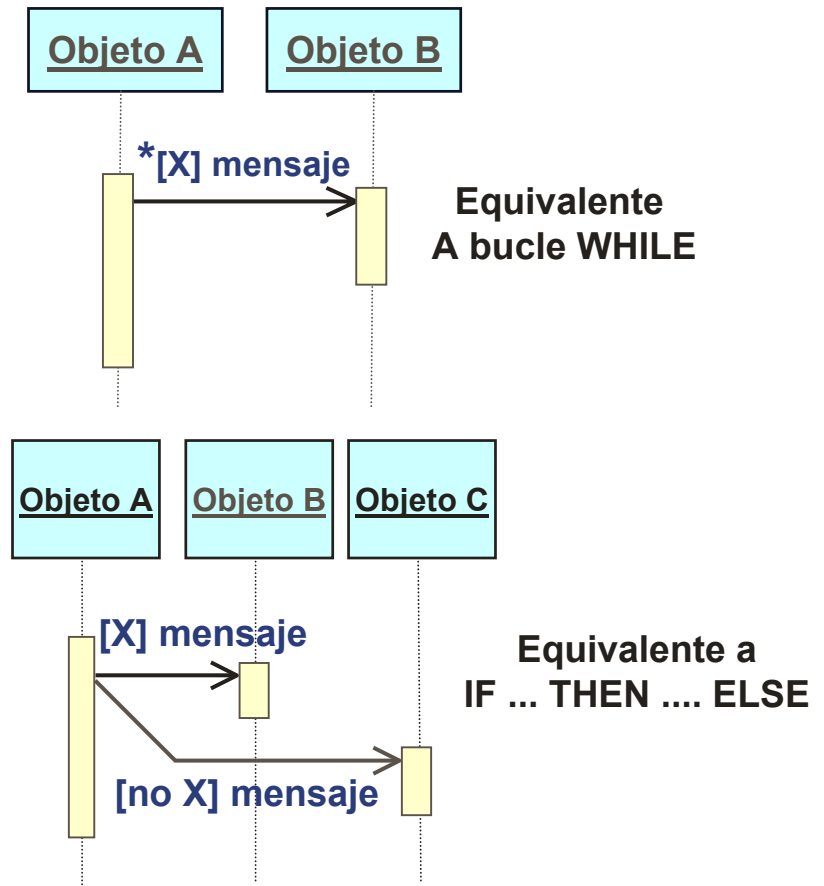
Restricciones temporales

- Los diagramas de secuencia pueden completarse por indicaciones textuales, expresadas en forma de texto libre o de pseudocódigo.
- El instante de emisión de un mensaje, llamado transición, puede señalarse con un nombre en el diagrama cerca del punto de partida de la flecha que simboliza el mensaje.



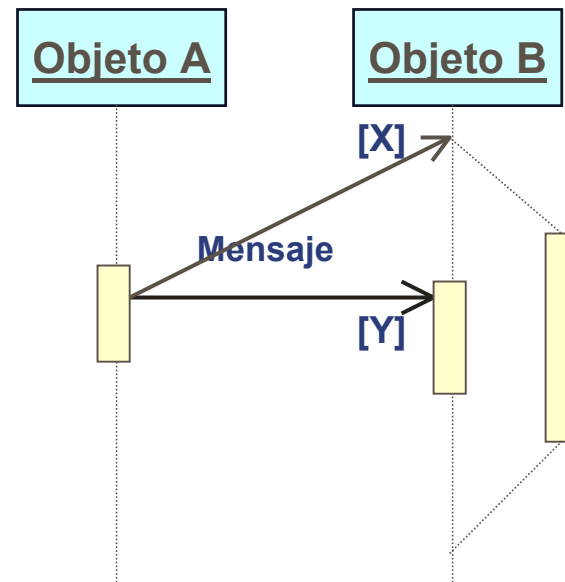
Mensajes condicionados

- La adición de pseudocódigo sobre la parte izquierda del diagrama permite la representación de bucles y bifurcaciones, de modo que los diagramas de secuencia pueden representar la forma general de una interacción, más allá de la simple consideración de un escenario particular.



Las alternativas

- Las alternativas, en el lado del destinatario del mensaje, se representan desdoblando la línea de vida del objeto destinatario.

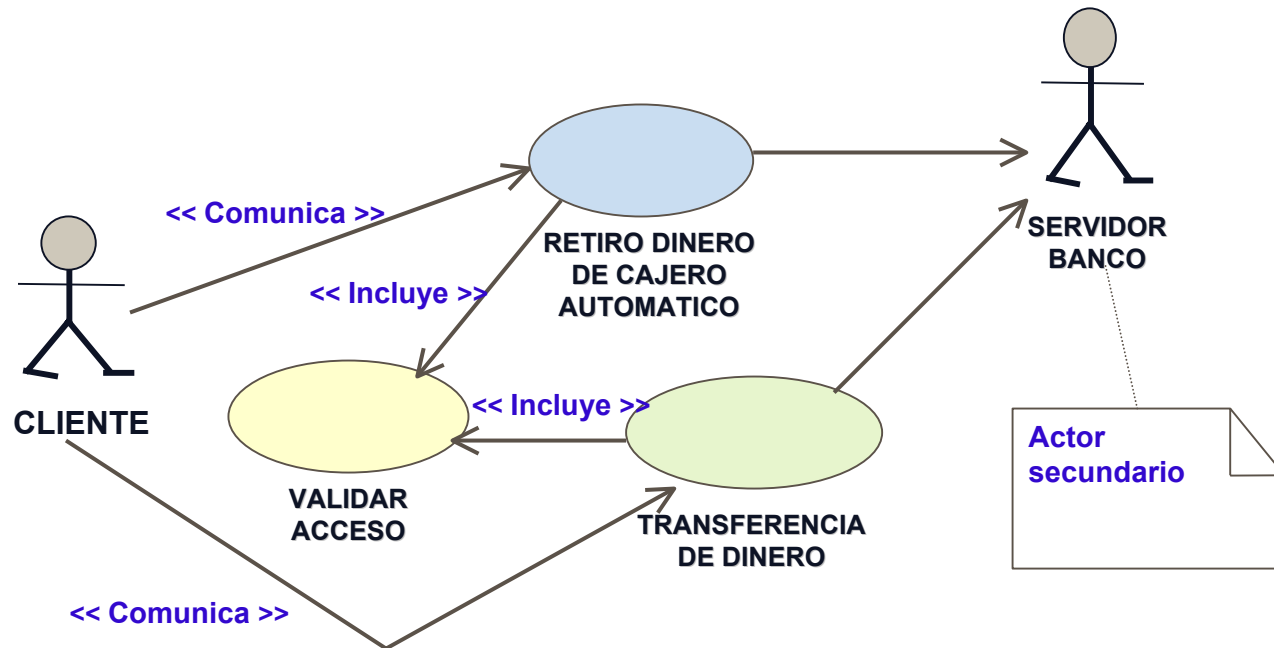




Ejemplo

- Supongamos que deseamos contruir el diagrama de secuencia del siguiente UC:
- *CLIENTE RETIRA DINERO DE CAJERO AUTOMATICO*
 - Pre-condiciones:
 - Usuario es cliente del Banco
 - Usuario conoce su password
 - El cajero esta comunicado con el servidor central de cuentas del Banco
 - El cliente tiene fondos suficientes en su cuenta como para efectuar retiro del monto requerido
 - El dispensador de dinero del cajero puede atender el retiro solicitado
 - Actores : Cliente y Servidor de Banco
 - Trigger : Arribo de un cliente al cajero para retirar dinero

Diagrama de Casos de Uso : Cajero Automatico

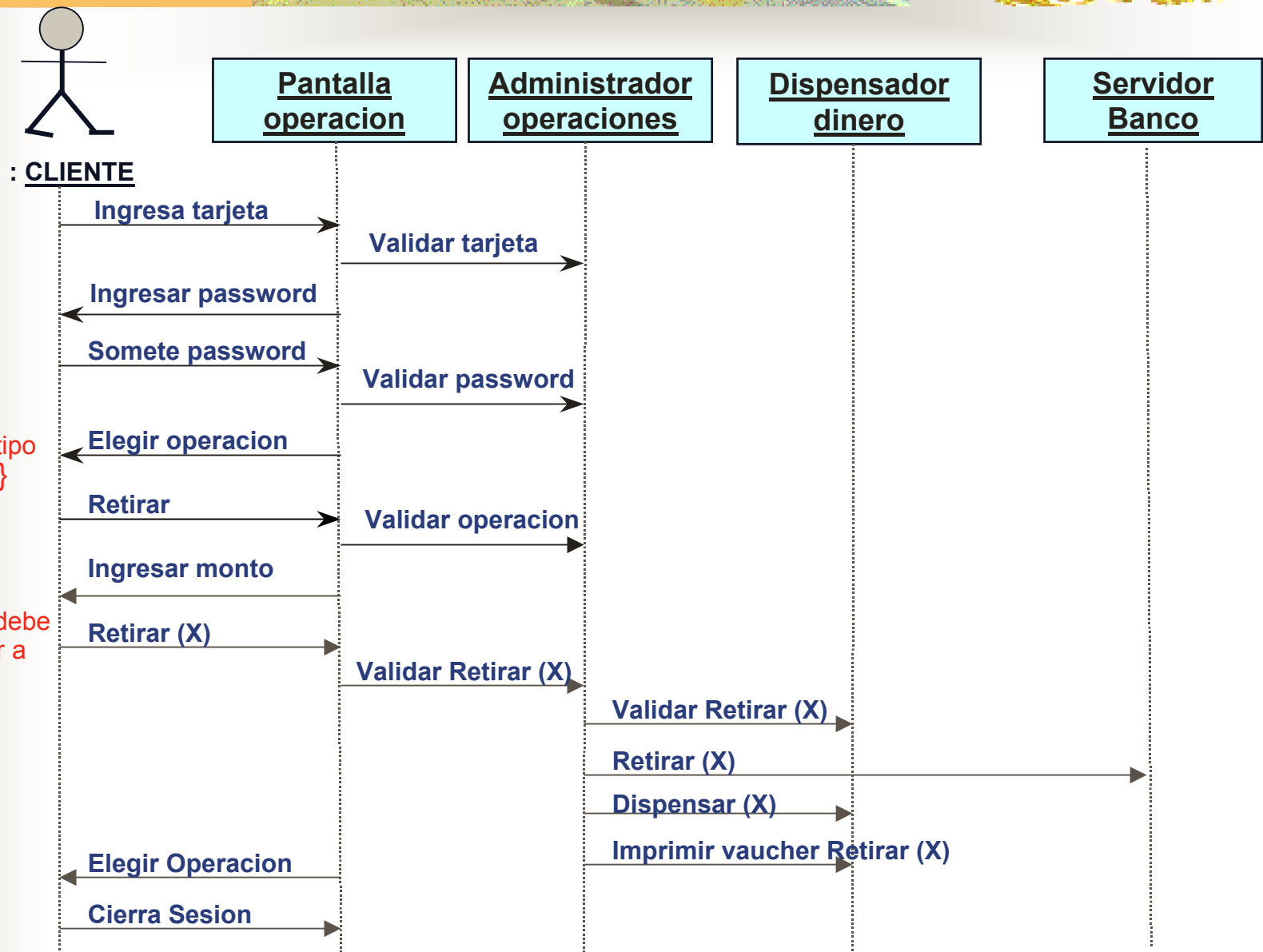


Especificación de Caso de Uso

1. Cliente ingresa tarjeta
2. Cajero valida tarjeta de cliente
3. Cliente somete password
4. Cajero valida password
5. Cajero muestra pantalla de operaciones
6. Cliente solicita operación de retiro
7. Cajero valida operación
8. Cajero muestra pantalla de retiro
9. Cliente requiere monto de retiro
10. Cajero valida posibilidad de atender retiro
11. Cajero solicita a servidor de Banco operación de retiro
12. Servidor verifica cuenta cliente
13. Servidor actualiza cuenta de cliente con monto de retiro
14. Cajero ordena a dispensador entregar monto de dinero de retiro
15. Cajero ordena impresión de vaucher de retiro
16. Cajero muestra pantalla de operaciones
17. Cliente cierra sesión (puede elegir continuar.....)

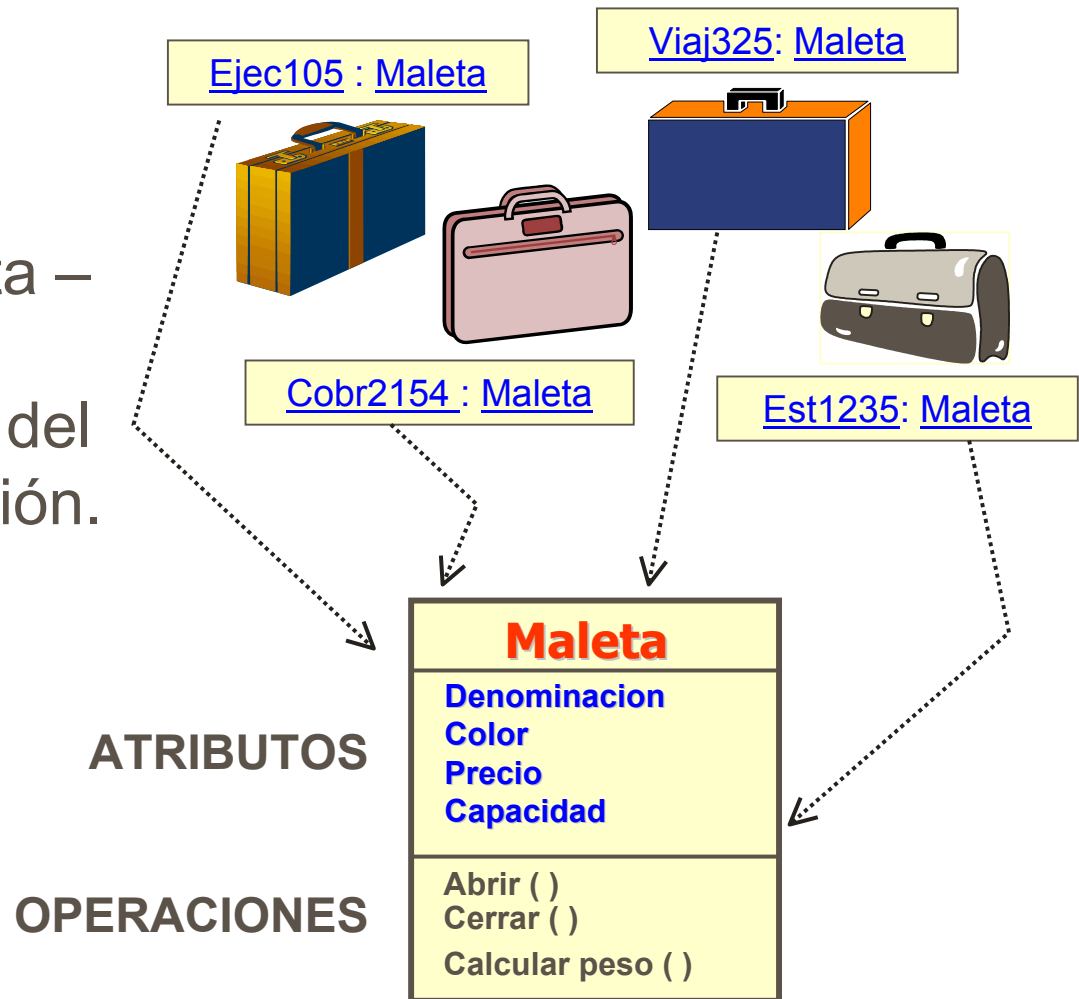
Las actividades
12 y 13
corresponden al
actor secundario
y no al caso de
uso.

La actividades 17
no corresponde al
caso de uso.



Las Clases

- Una clase es una descripción abstracta – condensada- de un conjunto de objetos del ámbito de la aplicación.
- Las clases se representan por rectángulos compartimentados.



Estereotipos de clases

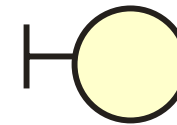
- El rectángulo que simboliza la clase puede contener también un estereotipo y propiedades.
- Un estereotipo es un mecanismo que se utiliza para categorizar las clases.
- UML define 3 estereotipos primarios :
<<Borde>> (boundary), <<Control>> y
<<Entidad>> (entity).
- Las propiedades designan los valores vinculados a un elemento de modelado, como atributos, asociaciones y etiquetas.

Nombre clase
<<Estereotipo>>
Propiedad

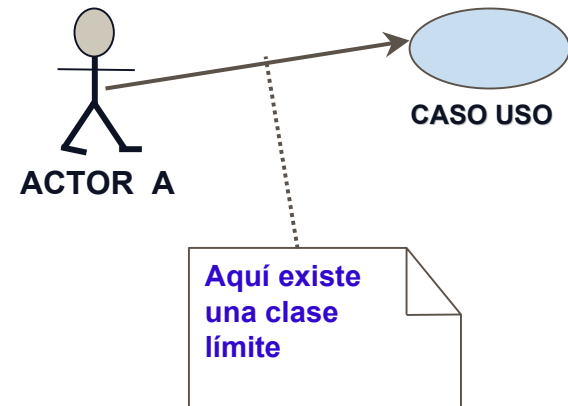
Formulario
Matricula
<<Borde>>
Estado = prueba
Autor = luis

Clases Límite o borde

- Son aquellas clases que establecen el límite entre su sistema y el resto del mundo.
- Estas incluyen todos los formularios, reportes, interfaces a hardware (printers o escaners) e interfaces a otros sistemas.
- Como mínimo debe existir una clase límite por cada interacción actor-caso uso.

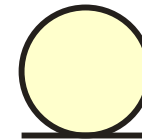


Simbolo UML para una clase limite



Clases Entidad

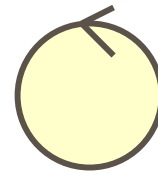
- Son las que almacenan información que es necesario grabar en almacenamiento persistente.
- Son las clases que tienen mayor significado para el usuario y son típicamente nombradas usando terminología del dominio del negocio.
- Frecuentemente se crea una Tabla en la BD por cada clase entidad. En lugar de definir primero la estructura de la BD, tenemos la opción de desarrollar la estructura de la BD desde información reunida en el modelo de objetos.



Simbolo UML para una clase entidad

Clase control

- Son las responsables de coordinar los esfuerzos de otras clases.
- Hay típicamente una clase control por caso de uso, la cual controla la secuenciación de eventos a través del caso de uso.
- La clase control no ejecuta alguna funcionalidad por sí misma y otras clases no envían muchos mensajes a ella. En lugar de ello envía un lote de mensajes a otras.
- La clase control simplemente delega responsabilidad a las otras clases, por esta razón son llamadas clases administrador.



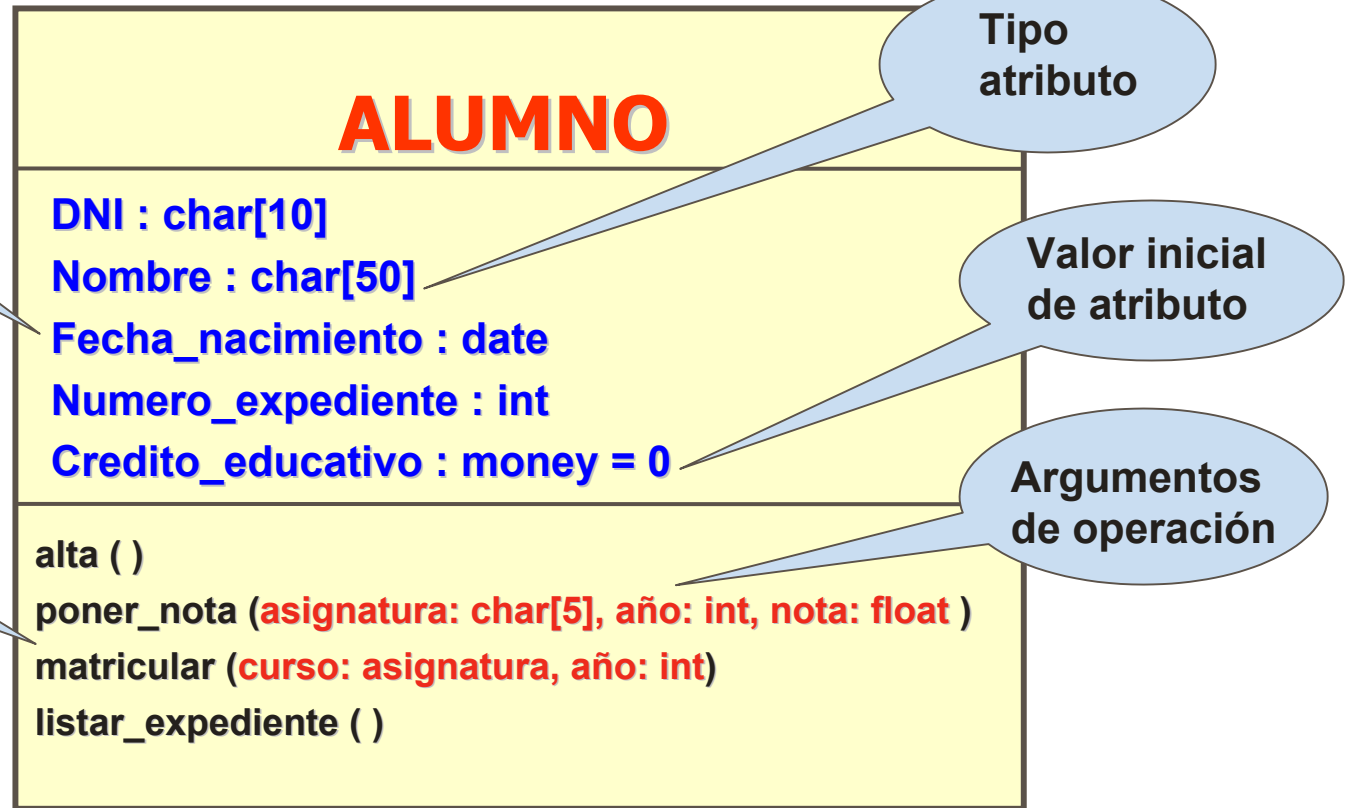
Simbolo UML para una clase control



Atributos y Operaciones

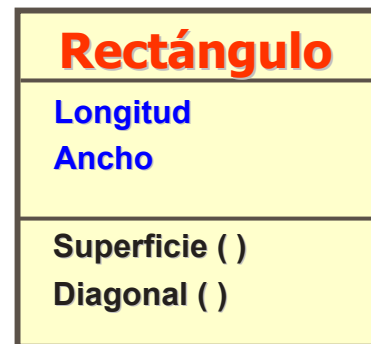
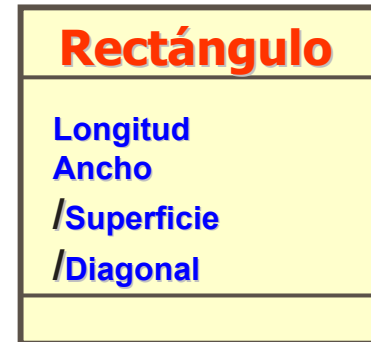
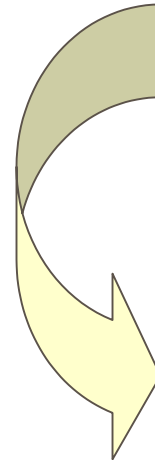
- Pueden ser mostrados de manera exhaustiva o no en los compartimientos de clases.
- Lo recomendable es completarlos progresivamente en la transición del análisis hacia el diseño.
- Sintaxis :
 - **Nombre_atributo**: tipo_atributo= valor_atributo
 - **Nombre_operación**(argumento :tipo_argumento=valor_predeterminado,): tipo_devuelto

Especificación detallada de una Clase



Atributos derivados

- Los atributos derivados ofrecen una solución para asignar a las clases propiedades que son derivadas de otras propiedades.
- En diseño el atributo derivado será transformado en una operación que encapsulará la propiedad.



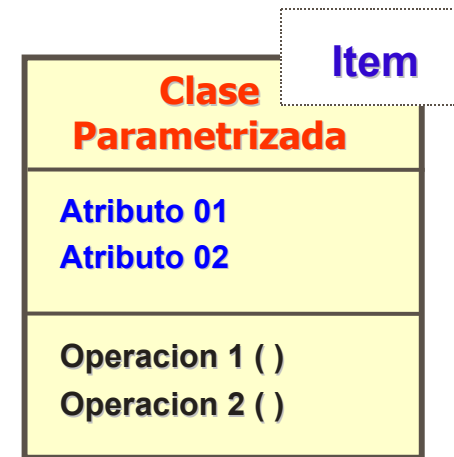
Visibilidad de los atributos y las operaciones

- Público : que hace el elemento visible a todos los clientes de la clase
- Protegido : que hace el elemento visible a las subclases de la clase.
- Privado : que hace el elemento visible sólo para la clase.



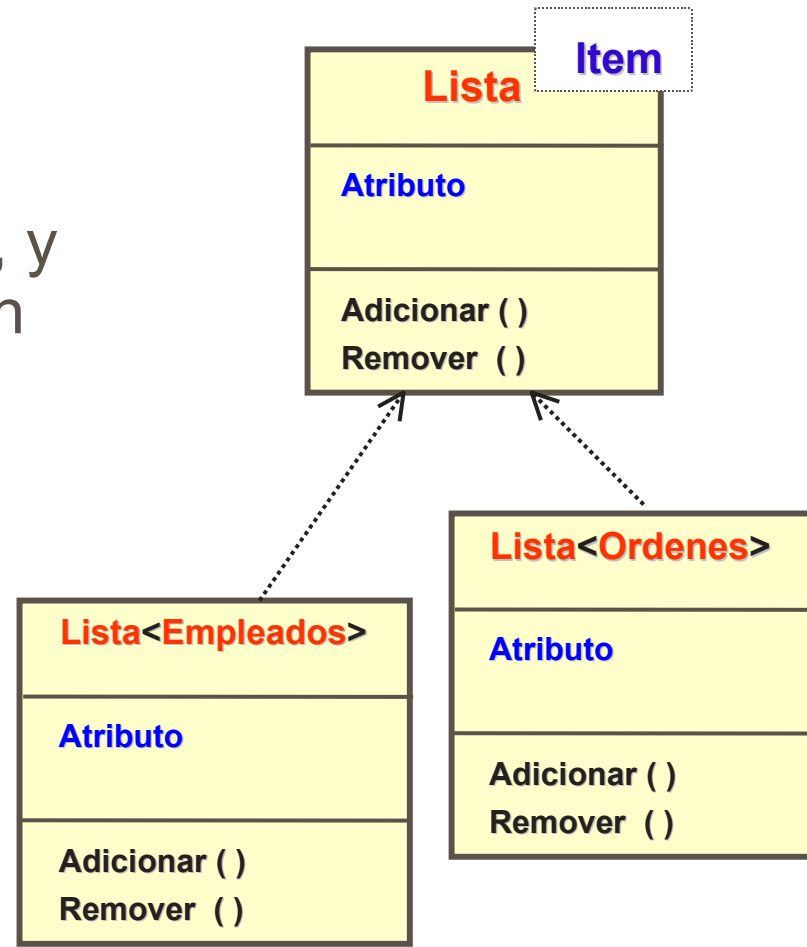
Clase parametrizada

- Son modelos de clases. Una clase parametrizable es una clase que es usada para crear una familia de otras clases.
- Una clase parametrizable no puede ser utilizada tal cual. Conviene primero instanciarla, a fin de obtener una clase concreta que podrá a su vez ser instanciada en objetos.



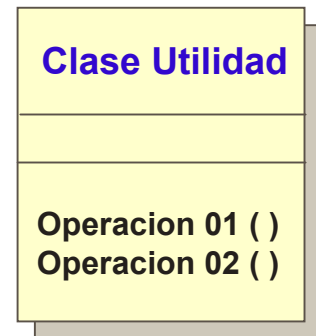
... Clase parametrizada

- Una clase parametrizable es alguna suerte de contenedor, y también es conocida como un template(plantilla), como los *templates de C++*.
- Este tipo de clases no aparece generalmente en el análisis.
- Las clases parametrizables se utilizan sobre todo en diseño detallado para incorporar, por ejemplo, componentes reutilizables.



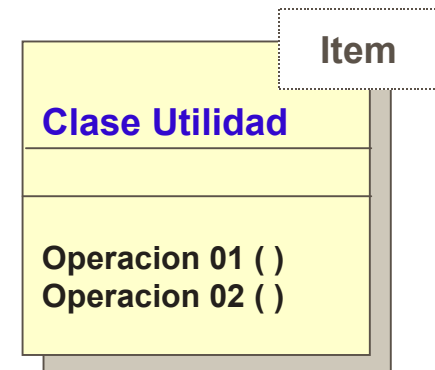
Clase Utilidad

- Es una colección de sólo operaciones.
- Aquellas operaciones de uso general que no pueden acomodarse dentro de una clase particular, se encapsulan dentro de una clase utilidad para uso por las otras clases del sistema.
- Son clases que no pueden ser instanciadas, y que son útiles para agrupar elementos dentro de un módulo, como por ejemplo las funciones de una biblioteca matemática.



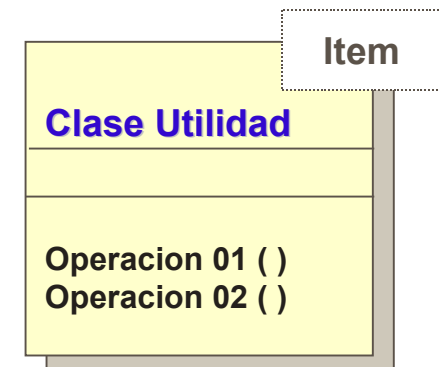
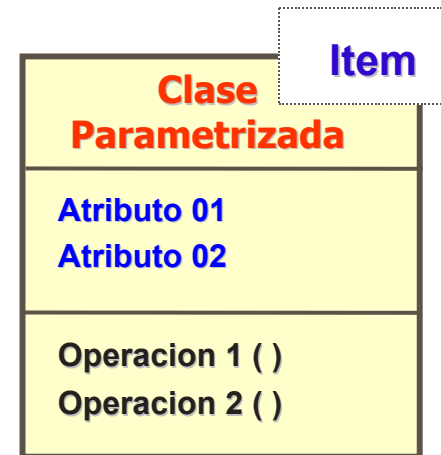
Clase parametrizada utilidad

- Es una clase parametrizada que contiene sólo un conjunto de operaciones.
- Representa la plantilla (template) que es utilizado para crear clases utilidad.



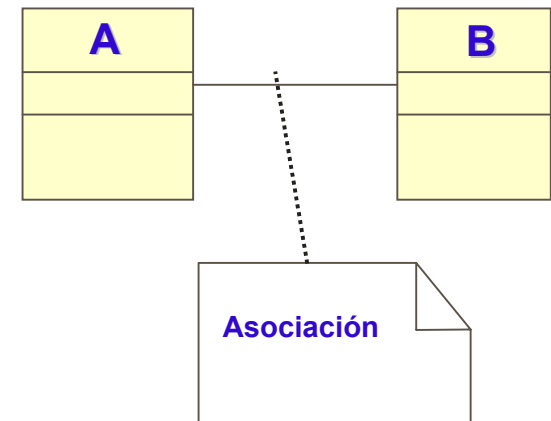
Metaclase

- Es una clase cuyas instancias son clases en lugar de objetos.
- Las clases parametrizadas y las clases parametrizadas utilidad son ejemplos de metaclases.



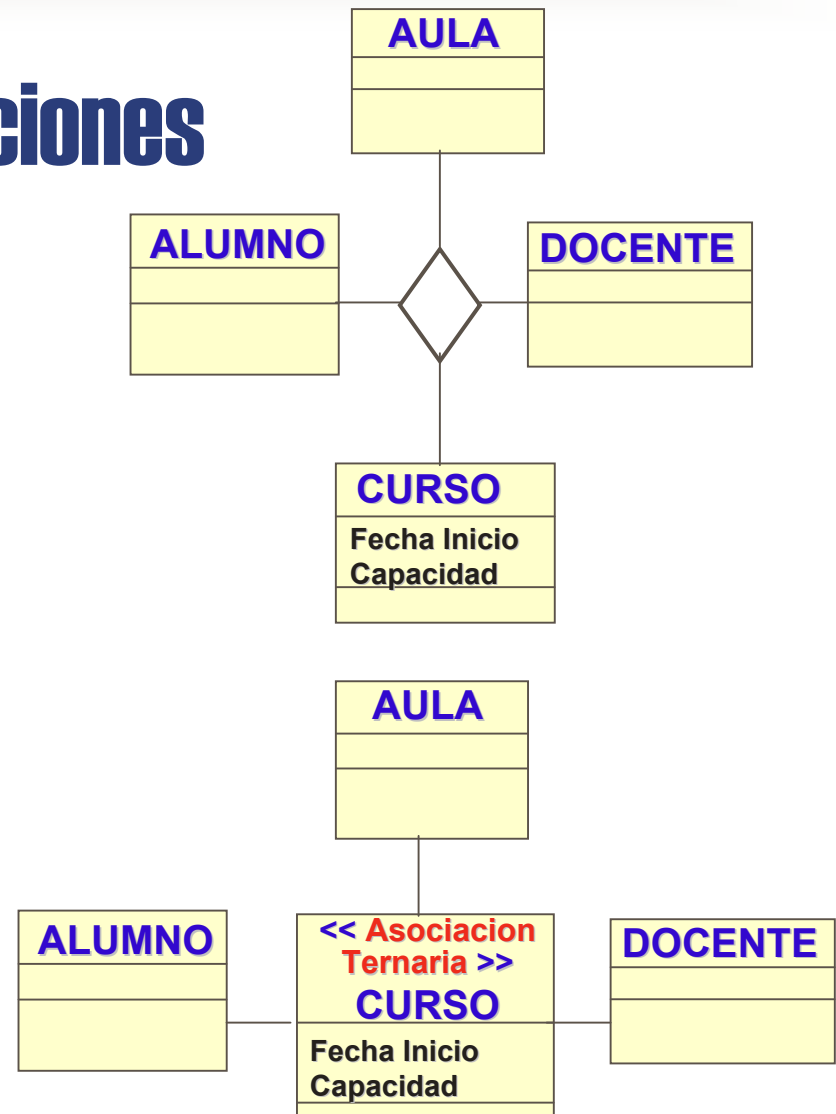
Asociaciones

- Representan relaciones estructurales entre clases de objetos.
- La mayor parte de las asociaciones son binarias, es decir, que conectan dos clases.



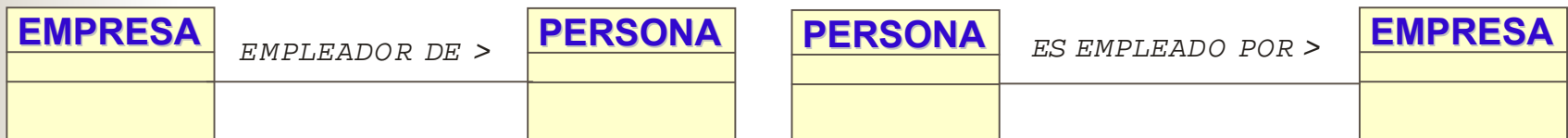
Pluralidad de las asociaciones

- Sin embargo existen relaciones superiores, y se representan por medio de un rombo al que llegan los diferentes componentes de la asociación.
- Las asociaciones n-arias pueden representarse generalmente promoviendo la asociación al rango de clase y añadiendo una restricción que expresa que las múltiples ramas de la asociación se instancian simultáneamente.



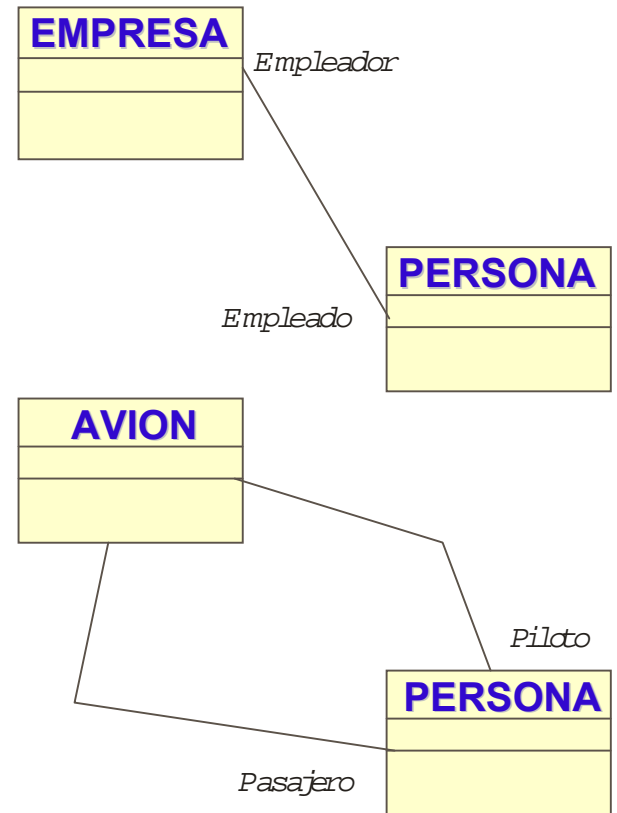
Nombrado de las asociaciones

- Los nombres de las asociaciones deben ser escritos en cursiva y ubicarse al centro y por encima de la línea de asociación.
- Se recomienda dar nombre a las asociaciones por una forma verbal, ya sea del tipo :
 - activa (***trabaja para***), o bien
 - pasiva (***es empleado por***).



Nombrado de roles o funciones

- Al extremo de una asociación se llama rol o función. Cada asociación binaria posee dos roles, uno en cada extremo.
- El rol describe como una clase ve a otra clase a través de la asociación.
- Cuando dos clases se relacionan por una sola asociación, el nombre de las clases basta normalmente para caracterizar el rol o función; el nombrado de los roles tiene mayor interés cuando existe mas de una asociación entre dos clases.

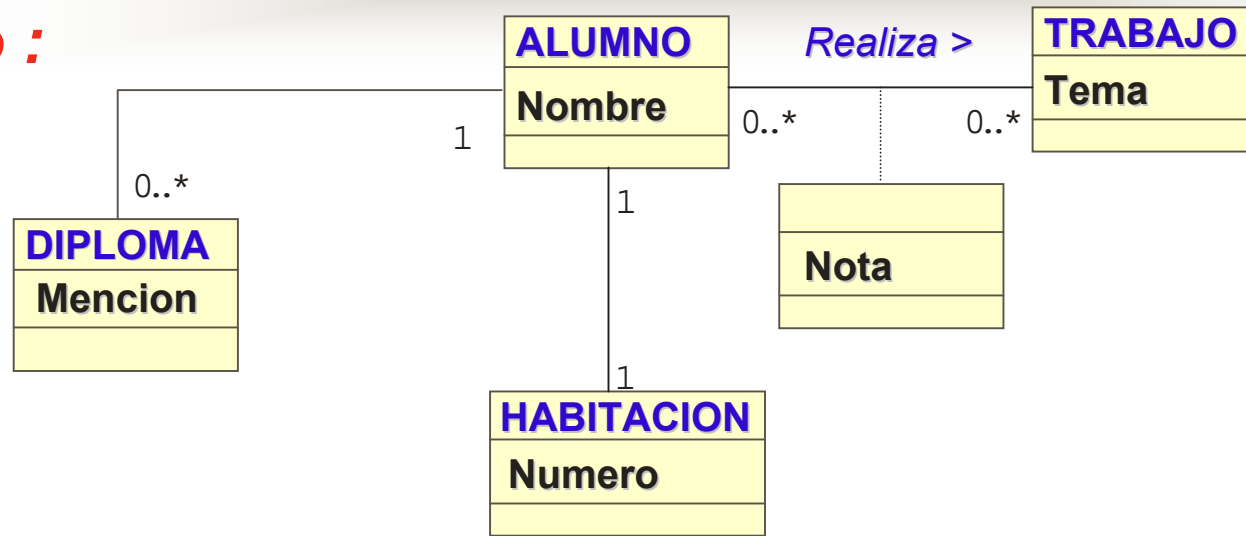




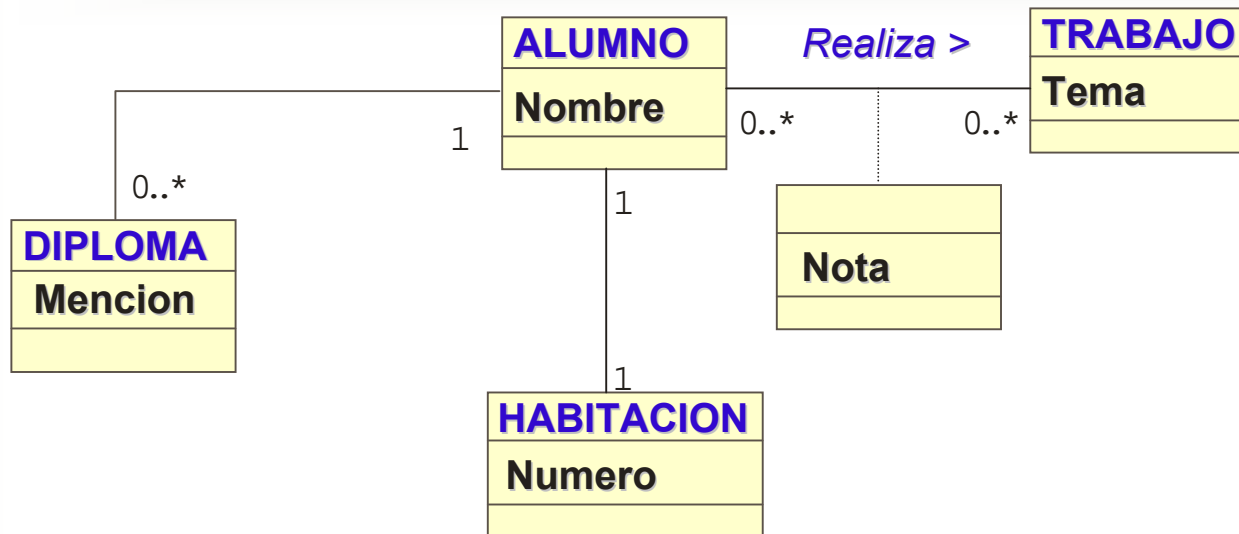
Colocación de atributos según la multiplicidad

- La materialización de las asociaciones toma todo su interés para las asociaciones ***n*** a ***n*** (*muchos a muchos*).
- Para las asociaciones **1** a **1**, los atributos de la asociación pueden ser desplazados a una de las clases que participan en la asociación.
- En las asociaciones **1** a ***n***, el desplazamiento es posible generalmente hacia la clase del lado *n*; sin embargo es frecuente promover la asociación al rango de clase para aumentar la legibilidad.

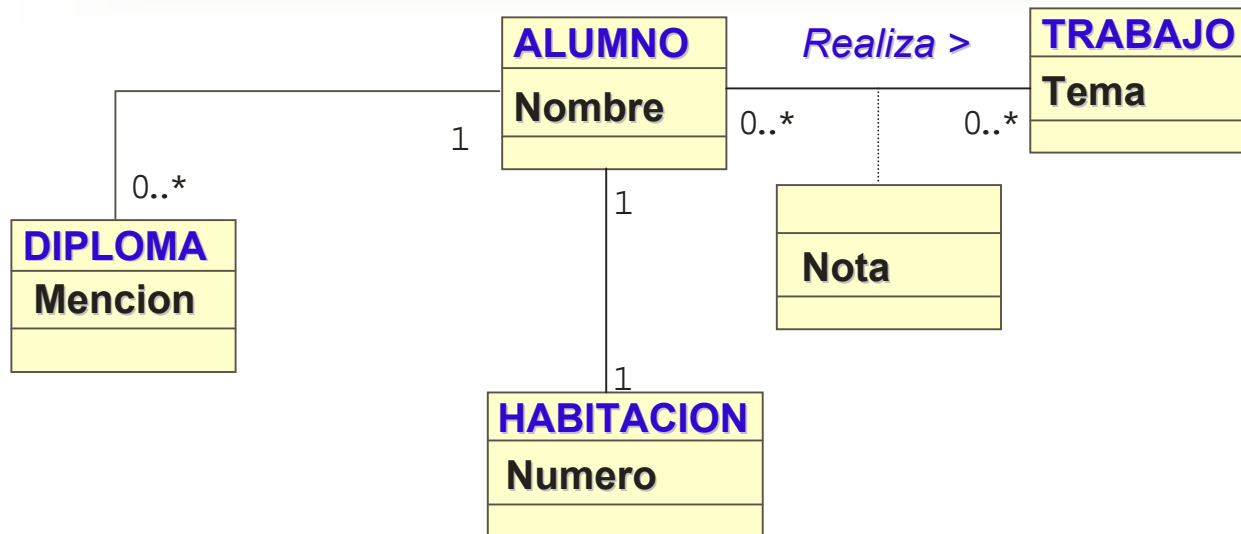
Ejemplo :



- La asociación entre la clase Alumno y la clase Trabajo es del tipo *n a n*. La clase Trabajo describe el tema, la solución aportada por el Alumno no se conserva.
- En el caso de los controles de conocimientos, cada Alumno escribe individualmente sobre un Trabajo dado y la nota obtenida no puede almacenarse en un Alumno en particular (*porque éste realiza numeros trabajos*) ni en un Trabajo (*porque hay tantas notas como alumnos*). La nota es un atributo de la relación entre la clase Alumno y la clase Trabajo.



- A fin de curso, cada Alumno recibe un Diploma con una mención que depende del rendimiento individual del Alumno.
- La relación entre el Diploma y el Alumno es personalizada porque un Diploma sólo afecta a un Alumno dado. La mención es un atributo del Diploma. La mención no se almacena en el Alumno por dos razones : no cualifica a un Alumno y un Alumno puede obtener varios Diplomas.



- Cada Alumno posee una Habitación la cual es identificada por un número y una Habitación no es compartida por varios Alumnos.
- La asociación entre los Alumnos y las Habitaciones es del tipo 1 a 1. El número es un atributo de la clase Habitación porque un número caracteriza al objeto Habitación.

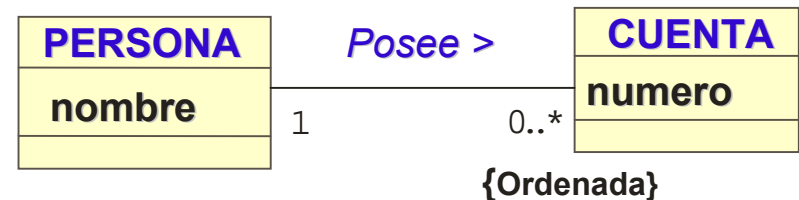


Restricciones de asociación

- Las restricciones pueden definirse sobre una relación o sobre un grupo de relaciones.
- La multiplicidad presentada anteriormente es una restricción sobre el número de enlaces que pueden existir entre dos objetos.
- Las restricciones se representan en los diagramas por expresiones encerradas entre llaves.
- Las restricciones pueden ser del tipo :
 - Ordenada
 - Subconjunto
 - O-exclusivo

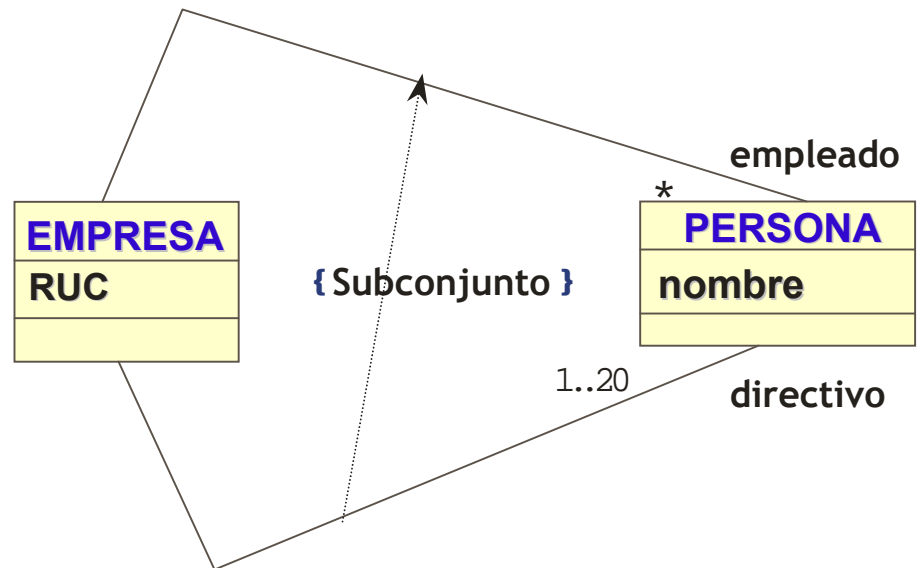
La restricción { ordenada }

- Puede colocarse sobre el rol para especificar que una relación de orden describe los objetos colocados en la colección; en este caso, el modelo no especifica cómo se ordenan los elementos, sino solo que el orden debe mantenerse durante la edición, eliminación u otra operación de objetos.



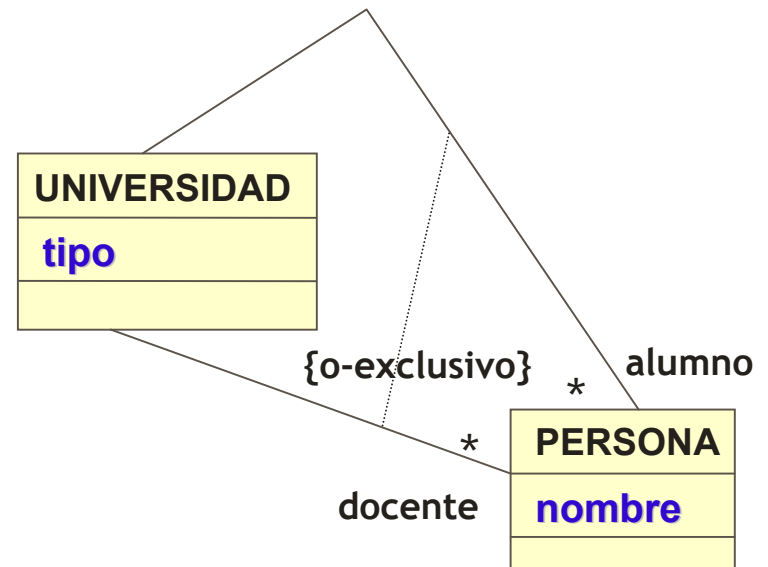
La restricción { Subconjunto }

- Esta restricción indica que una colección está incluída en otra colección.
- Establece que los objetos que participan en la asociación restringida debe también participar en la otra.



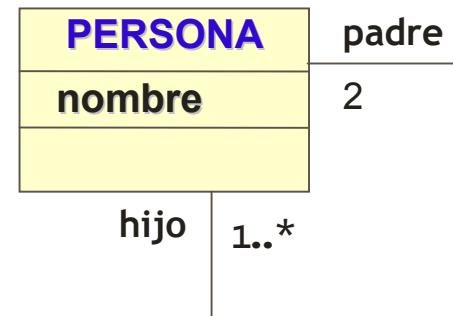
La restricción { o-exclusivo }

- Precisa que para un objeto dado, es válida una sola asociación entre un grupo de asociaciones.
- Esta restricción evita la introducción de subclases artificiales para materializar la exclusividad.



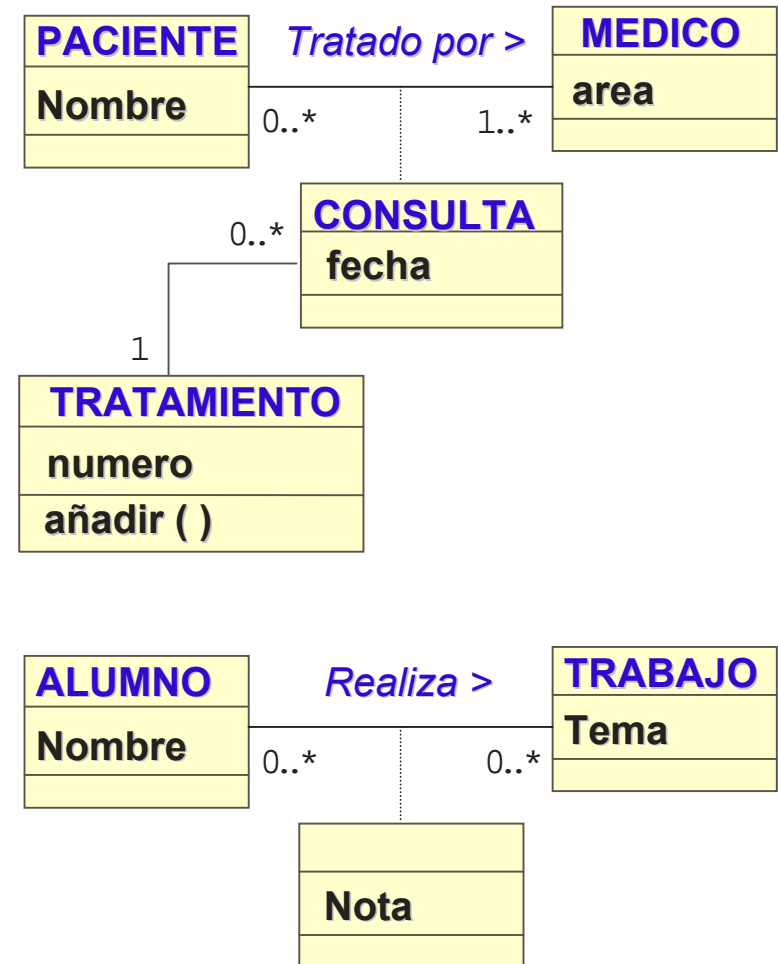
Asociación recursiva

- Se representa por la asociación de una clase con si misma.
- El nombre del rol de la asociación toma de nuevo toda su importancia para distinguir las instancias que participan en la relación.



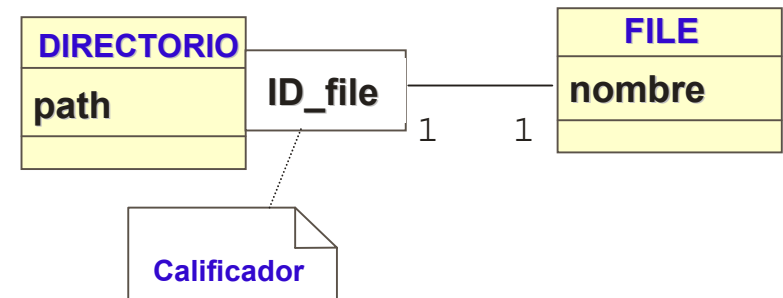
La Clase Asociación

- Una asociación puede representarse por una clase, para añadir, por ejemplo, atributos y operaciones a la asociación.
- Es una clase como otras y puede participar en otras relaciones en el modelo.
- Una asociación que contiene atributos sin participar en relaciones con otras clases se llama asociación atribuida.



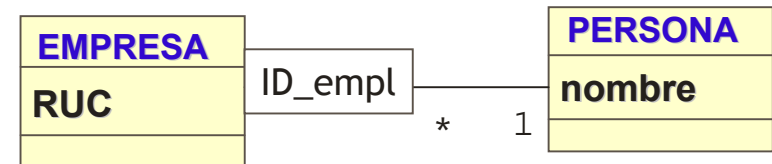
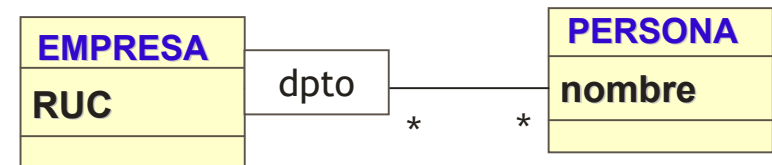
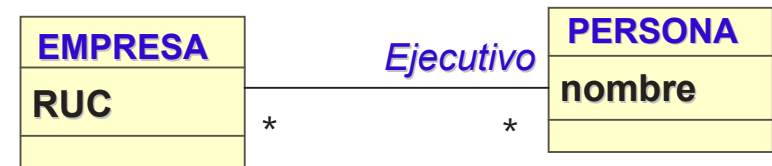
Restricción de una asociación

- Consiste en seleccionar un subconjunto de objetos entre el conjunto de objetos que participan en una asociación.
- Se realiza por medio de una tupla de atributos particulares (llamada clave) y se utiliza conjuntamente con un objeto de la clase de partida.
- La clave se representa sobre el rol de la clase de partida bajo el nombre de **cualificador** o **calificador**. La clave pertenece a la asociación y no a las clases asociadas.
- El **cualificador** es un atributo especial que reduce la multiplicidad efectiva de una asociación. Las asociaciones uno a muchos y muchos a muchos pueden ser cualificadas.



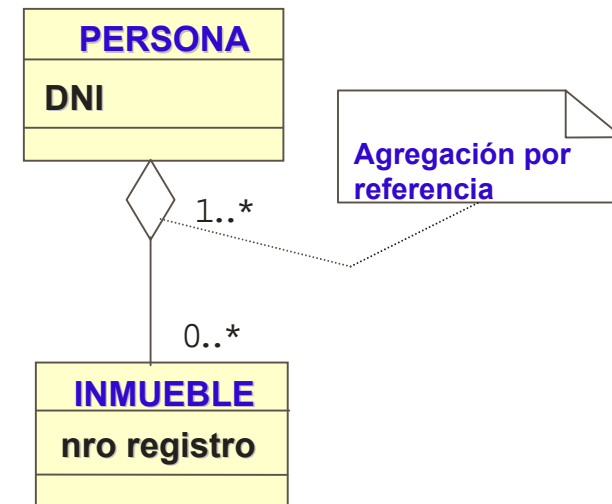
Cualificador

- Cada instancia de la clase A acompañada del valor de la clave, identifica un subconjunto de instancias de B que participan en la asociación. La restricción reduce la multiplicidad de la asociación.
- El par (instancia de clase de partida, valor de clave) identifica un subconjunto de instancias de la otra clase. A menudo la multiplicidad se reduce a 1, aunque no siempre.



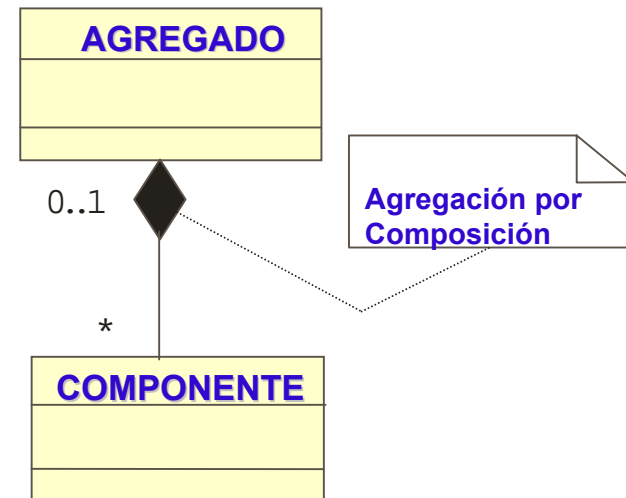
La Agregación de Clases

- Expresa un fuerte acoplamiento entre los objetos agregados.
- Los siguientes criterios implican una agregación:
 - Una clase forma parte de otra clase.
 - Los valores de atributos de una clase se propagan en los valores de atributos de otra clase.
 - Una acción sobre una clase implica una acción sobre otra clase.
 - Los objetos de una clase son subordinados a los objetos de otra clase.

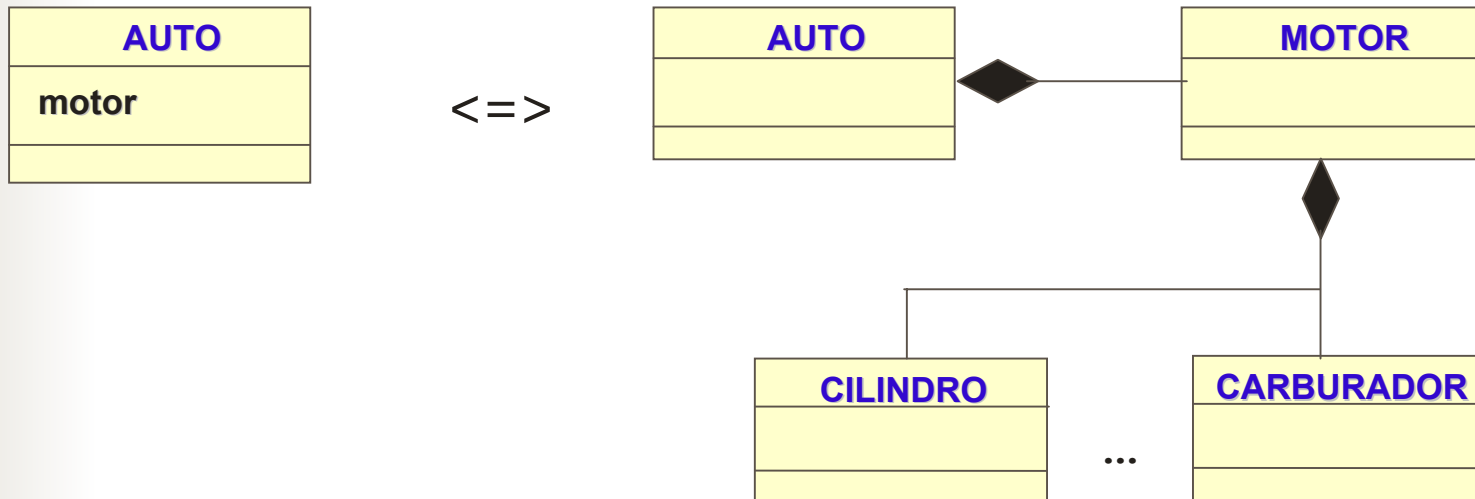


La Composición

- La contención física es un caso particular de agregación, llamado COMPOSICIÓN.
- Los atributos constituyen un caso particular de agregación realizada por valor: están físicamente contenidas por el agregado.
- La composición implica una restricción sobre el valor de multiplicidad en el lado del agregado: sólo puede tomar los valores 0 o 1. El valor 0 en el lado del componente corresponde a un atributo no explicitado.



Equivalencia entre notación de contención por valor de atributo y composición



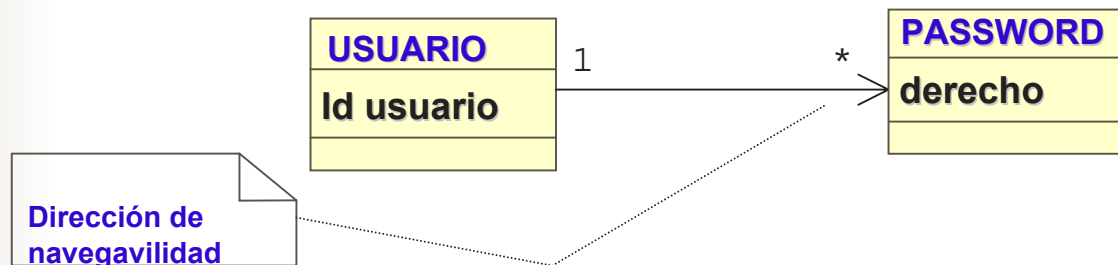


Atributos de una agregación

- En una gregación la *clase todo* tendrá atributos adicionales por cada una de las *clases parte*. Si estos atributos están contenidos por valor o por referencia aquí es donde se definen.
- Un atributo de contención por valor sugiere que el todo y la parte son creadas y destruidas al mismo tiempo.
- Una agregación por referencia sugiere que el todo y la parte son creadas y destruidas en diferentes tiempos.

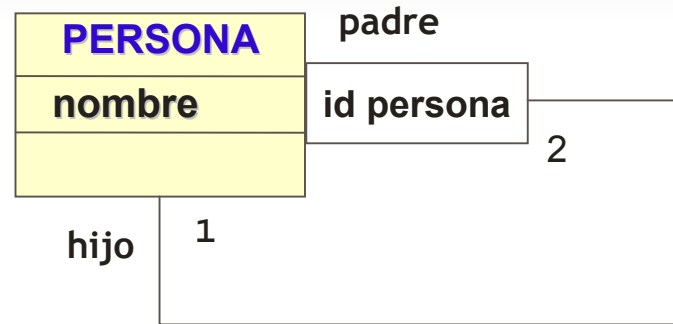
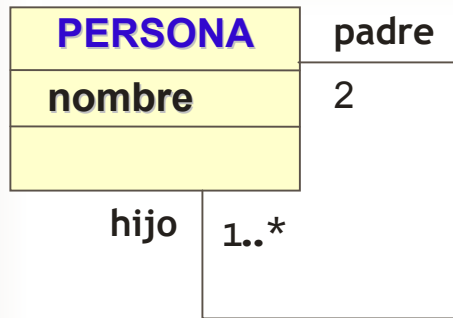
Navegabilidad

- Las asociaciones describen la red de relaciones estructurales que existen entre las clases y que dan lugar a los enlaces entre los objetos.
- Los enlaces pueden ser vistos como canales de navegación entre los objetos.
- En principio, las asociaciones son navegables en ambas direcciones. En ciertos casos sólo es útil una dirección de navegación.
- La navegabilidad se representa por una flecha orientada hacia la clase sobre la que es posible la navegación.



Expresiones de Navegación

- La sintaxis de las expresiones de navegación viene dada por las cuatro reglas siguientes :
 - destino::=conjunto '.' selector
 - destino::=conjunto '.' '~' selector
 - destino::=conjunto '[' expresion_booleana ']'
 - destino::=conjunto '.' selector '[' valor_de_clave ']'
- El selector corresponde bien a un nombre de atributo de los objetos del conjunto, bien a un nombre de asociación de la clase de objetos, o bien a un nombre de rol opuesto sobre un enlace que concierne a los objetos del conjunto. El destino es un conjunto de valores o de objetos cuyo número depende de la multiplicidad del conjunto y de la asociación.



- **UnaPersona.Hijos** designa todos los hijos de una persona dada.
- **UnaPersona. ~Hijos** designa los padres de una persona dada.
- **UnaPersona.Hijos[edad>=18años]** designa todos los hijos mayores de edad de una persona dada.
- **UnaPersona.Hijo[UnNombre]** identifica un hijo dado de manera no ambigua.



La Generalización

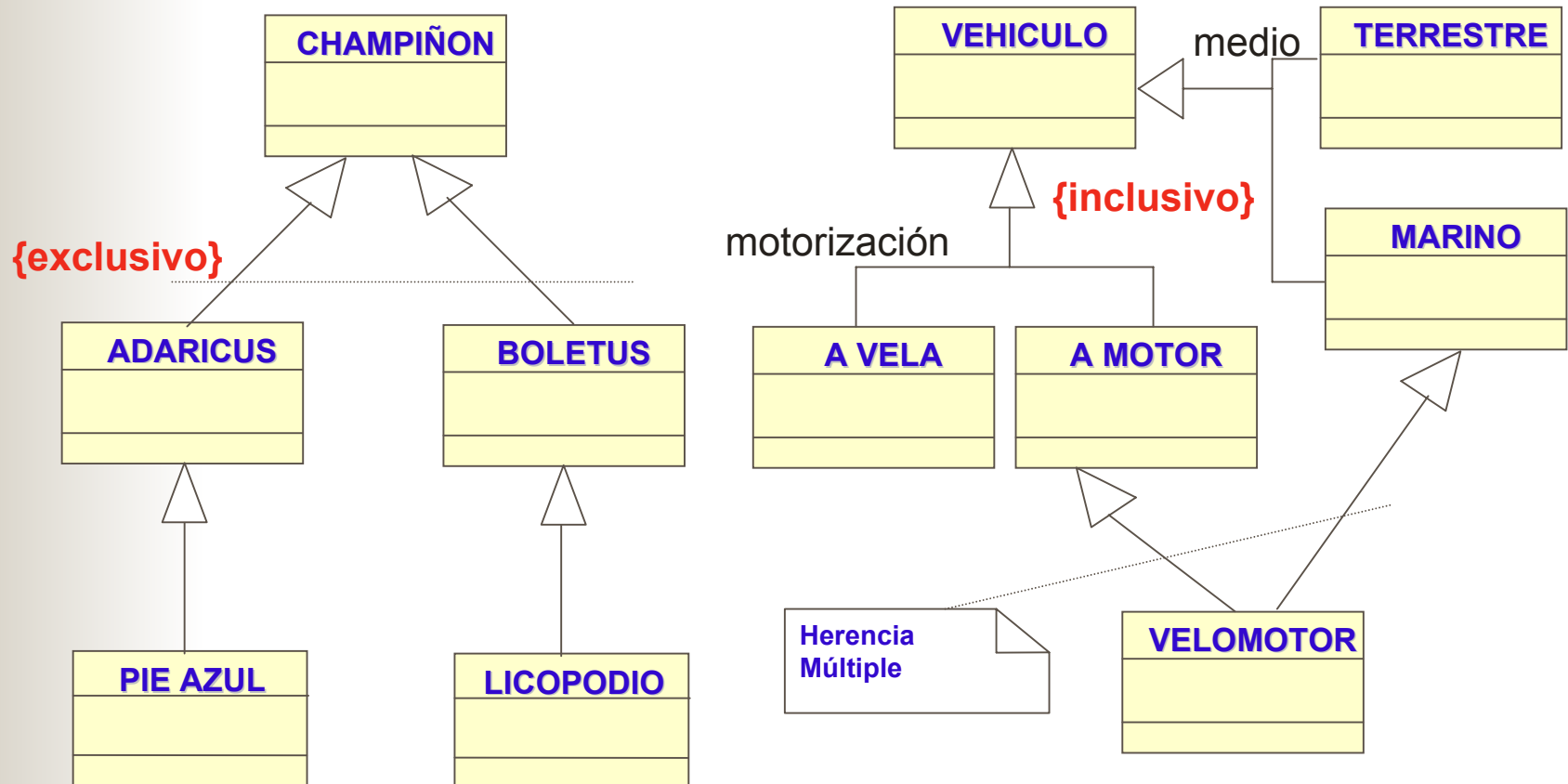
- UML emplea este término para designar la relación de clasificación entre un elemento más general y un elemento más específico.
- La generalización se aplica principalmente a las clases, paquetes y casos de uso.
- Los atributos, las operaciones, las relaciones y las restricciones definidas en las superclases se heredan íntegramente en las subclases.
- Las clases pueden tener varias superclases; en este caso, la generalización se denomina múltiple.
- Una clase puede ser especializada según varios criterios diferentes (discriminates).



Restricciones de la Generalización

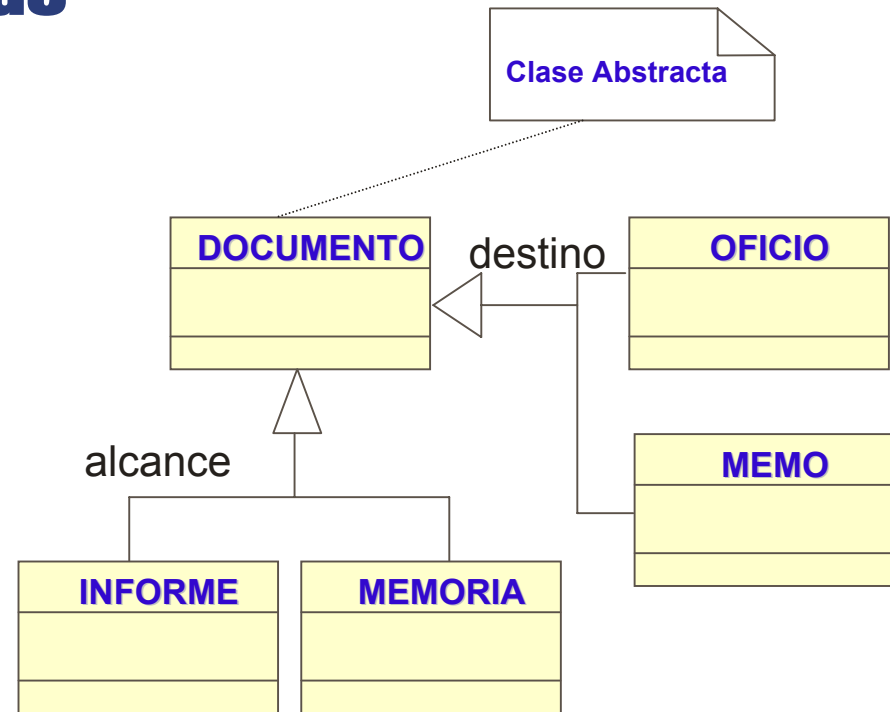
- En principio, la generalización simboliza una descomposición exclusiva, es decir, que un objeto es como máximo instancia de una de las subclases.
- La restricción {Disyuntivo} o {Exclusivo} indica que cualquier clase descendiente de una clase A sólo puede ser descendiente de una subclase de A.
- La restricción {Encabalgamiento} o {Inclusivo} indica que una clase descendiente de una clase A pertenece al producto cartesiano de las subclases de A. Un objeto concreto se construye a partir de una clase obtenida por mezcla de varias superclases.

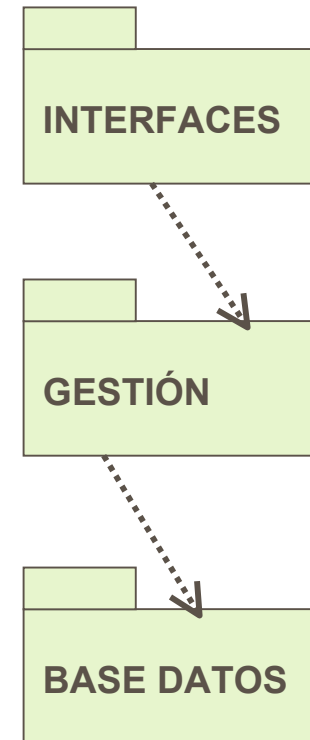
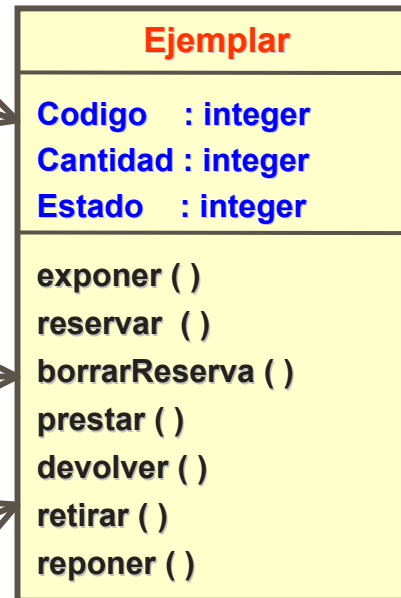
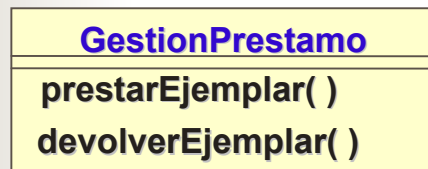
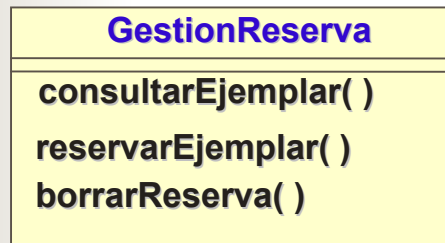
Restricciones de la Generalización



Las Clases Abstractas

- No son instanciables directamente; no dan lugar a objetos, sino que sirven de especificación más general –de tipo- para lograr manipular los objetos instancias de una (o varias) de sus subclases.





Comportamiento de los Objetos



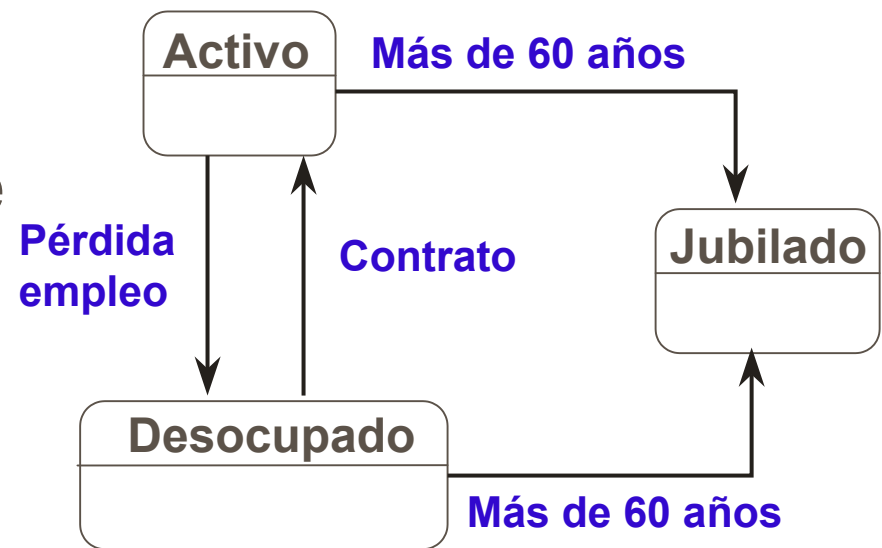


Qué es un Diagrama de Transición de Estado ?

- Es usado para mostrar la historia de vida de los objetos de una clase dada, los eventos que causan una transición desde un estado a otro, y las acciones que resultan desde un cambio de estado.
- El espacio de estados de una clase dada es la enumeración de todos los posibles estados de un objeto.

Los Diagramas de Estados-Transiciones

- Visualizan autómatas de estados finitos desde el punto de vista de los estados y las transiciones.



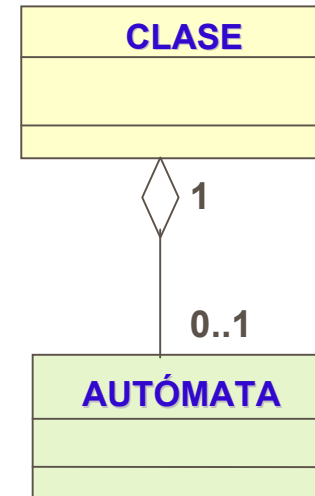


Los Automatas

- El comportamiento de los objetos de una clase puede describirse de manera formal en términos de estados y eventos, por medio de un autómata vinculado a la clase considerada.
- Los objetos que no presentan un comportamiento reactivo muy marcado pueden considerarse como objetos que están siempre en el mismo estado: en este caso, sus clases no poseen ningún autómata.
- Un autómata es una abstracción de los comportamientos posibles, a imagen de los diagramas de clases que son abstracciones de la estructura estática.

Autómatas y escenarios

- Los autómatas y los escenarios son complementarios. Los escenarios se representan por una colaboración entre objetos.
- La forma de la interacción entre los objetos que colaboran dentro de un escenario se determina por los estados respectivos de los diferentes objetos.



Abstracción de los comportamientos posibles.

Los Estados

- Cada objeto está en un momento determinado en un estado particular.
- Los estados se caracterizan por la noción de duración y de estabilidad. Un objeto está siempre en un estado dado por un cierto tiempo y un objeto no puede estar en un estado desconocido o no definido.

CURSO SECCION
seccion numAlumnos
Adicionar()

Matemáticas I :CURSO SECCION
numAlumnos=7

El máximo nro de alumnos por sección es 10

numAlumnos <10



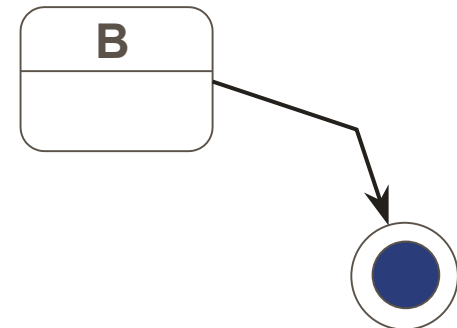
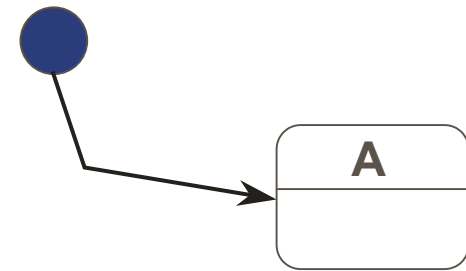
numAlumnos =10



Estado inicial y final

- Los autómatas usados por UML son deterministas.
- Ello significa que siempre hay que describir el estado inicial del objeto, el cual es único y aparece cuando el objeto es creado.
- El estado final indica el fin de la vida de un objeto. Es posible tener varios estados finales que corresponden cada uno a una condición de fin distinta.

estado inicial



estado final

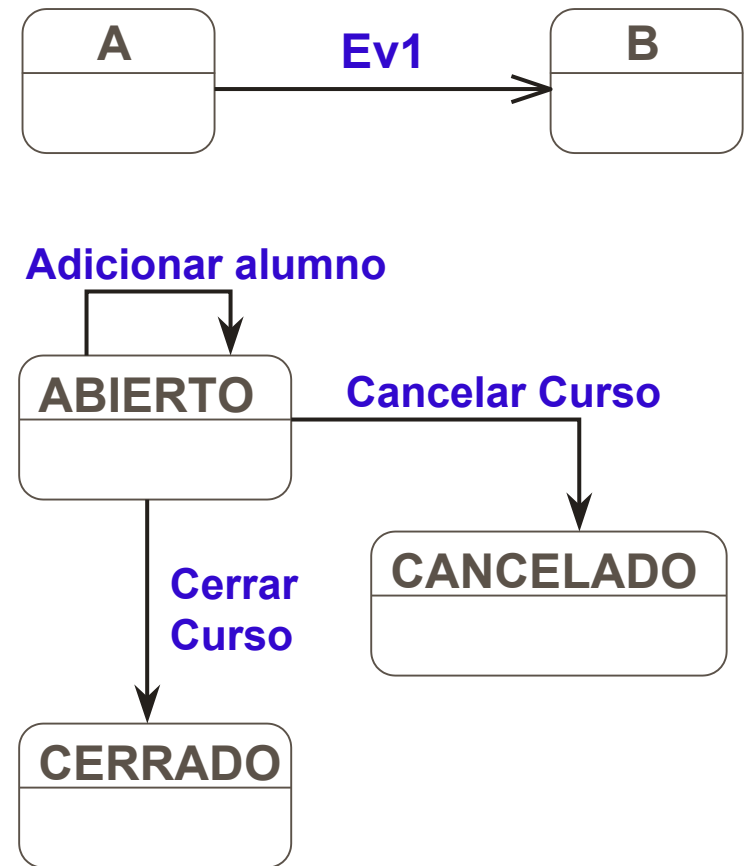
Las Transiciones

- Cuando las condiciones dinámicas evolucionan, los objetos cambian de estado siguiendo las reglas descritas en el autómata asociado a sus clases.
- Los estados están vinculados por conexiones unidireccionales llamadas transiciones.
- El paso de un estado a otro se efectúa cuando se desencadena una transición por un evento que aparece en el ámbito del problema.
- Las transiciones no vinculan necesariamente estados distintos.



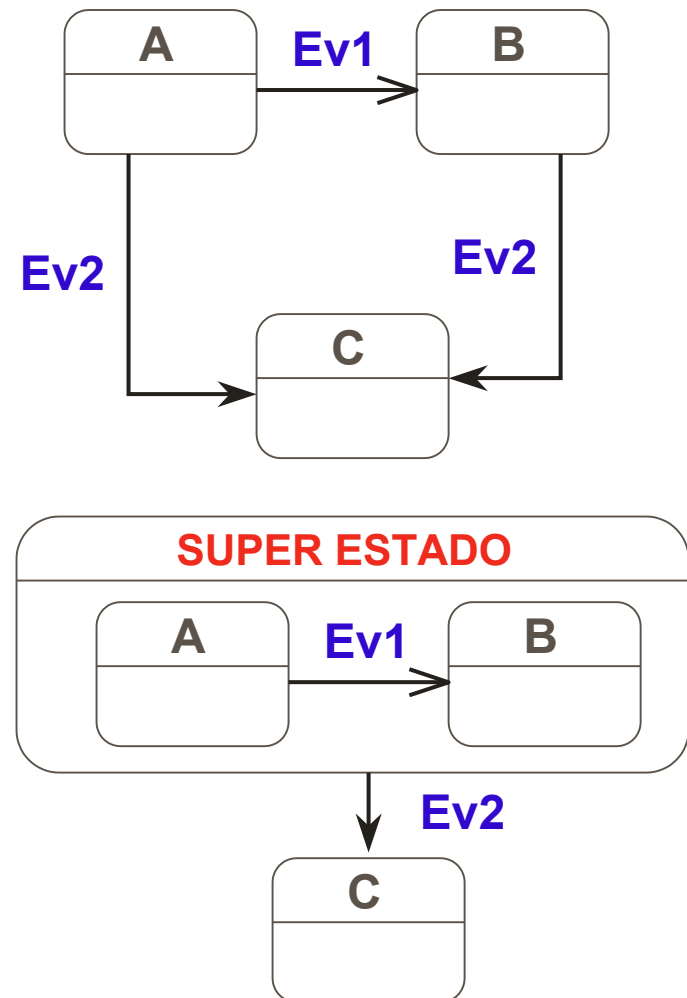
Los Eventos

- Un evento sirve de desencadenante para pasar de un estado a otro.
- Los eventos determinan que caminos deben seguirse. Los eventos, las transiciones y los estados son indisolubles en la descripción del comportamiento dinámico.



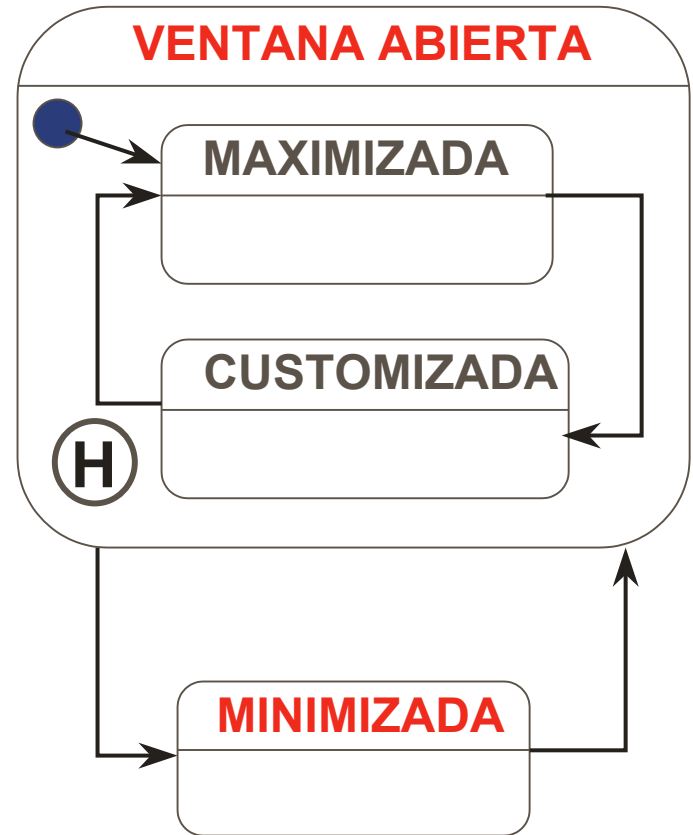
Super y sub estados

- Super estados son los estados mas generales, los estados mas específicos se llaman sub estados.
- Un estado puede descomponerse en varios sub estados disyuntivos (estados anidados).
- Los sub estados heredan características de su super estado, en particular las variables de estado y las transiciones externas.



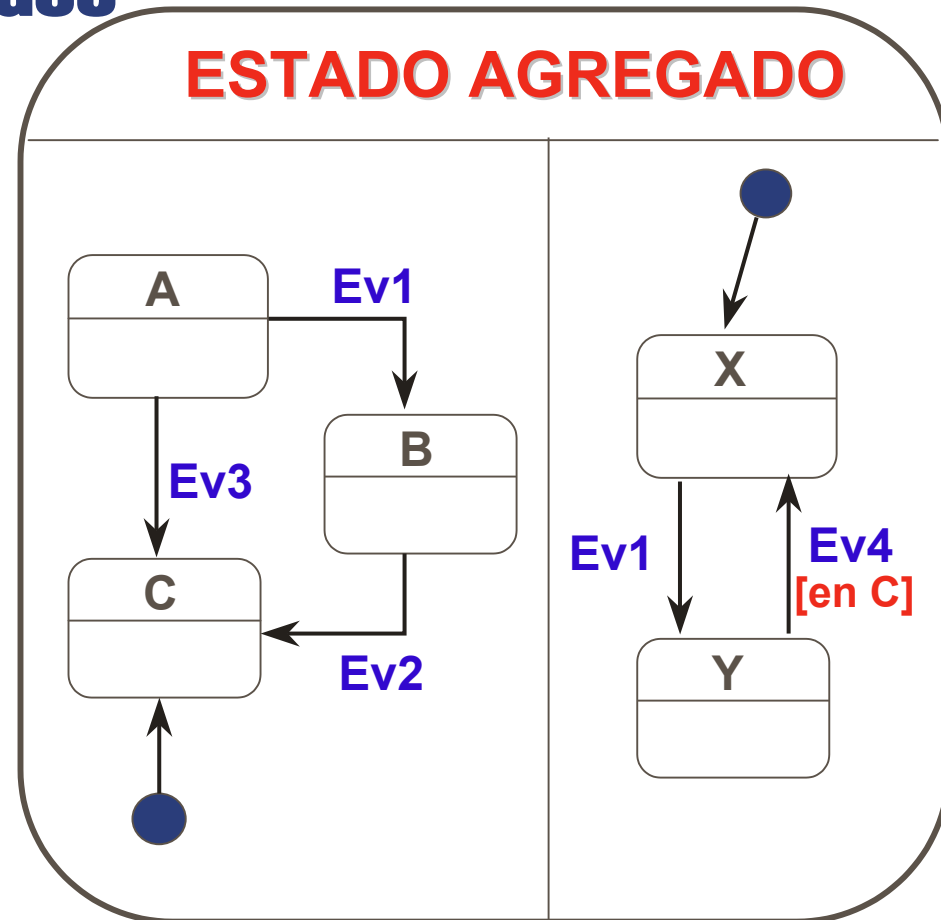
Estados anidados con Historia

- El uso de la característica de historia **(H)** señala que al retornar un objeto a un super estado, este ingresará al último estado en el que estuvo dentro del super estado.
- Si la característica de historia no es utilizada siempre el sub estado inicial del super estado será asignado al objeto retornante.



Agregación de Estados

- La agregación de estados es la composición de un estado a partir de otros varios estados independientes.
- La composición es de tipo conjuntiva lo que implica que el objeto debe estar simultáneamente en todos los estados que componen la agregación de estados



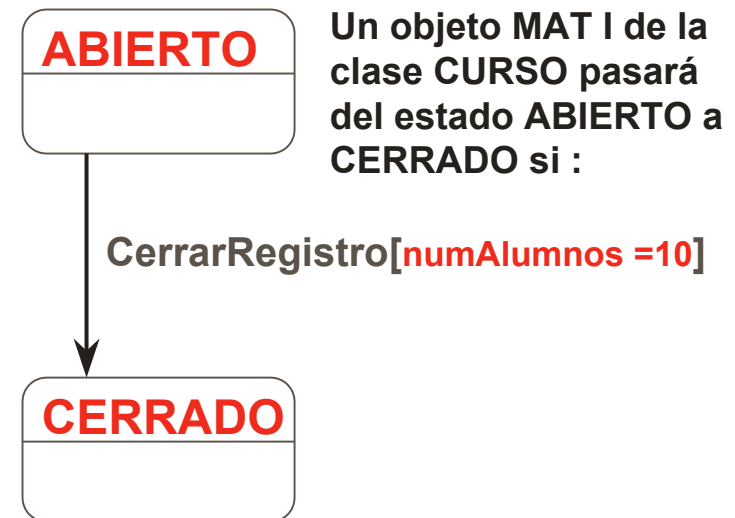
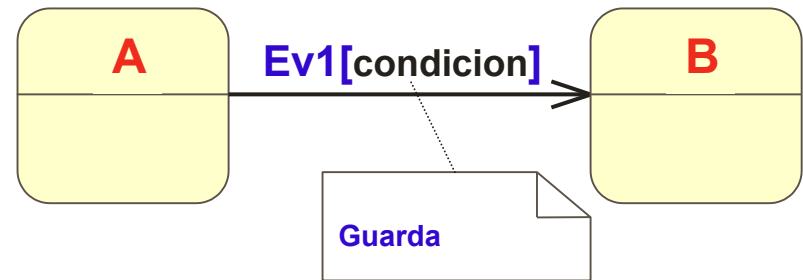


Especificación de un Evento

- Nombre del evento
- Lista de parámetros
- Objeto emisor
- Objeto destinatario
- La descripción del significado del evento

Los Guardas

- Un guarda es una condición booleana que valida o no el desencadenamiento de una transición en la ocurrencia de un evento.
- Los guardas permiten mantener el aspecto determinista de un autómata de estados finitos, incluso cuando varias transiciones pueden ser desencadenadas por el mismo evento.





Las Operaciones, las acciones y las actividades

- El enlace entre las operaciones definidas en la especificación de clase y los eventos que aparecen en los diagramas de estados-transiciones se efectúa por medio de acciones y actividades.
- Cada transición puede ir acompañada del nombre de una acción a ejecutar cuando la transición sea desencadenada por un evento.
- La acción corresponde a una de las operaciones declaradas en la clase del objeto destinatario del evento.
- Los estados pueden contener también acciones que se ejecutan al entrar o salir del estado o al ocurrir un evento mientras el objeto está en ese estado.

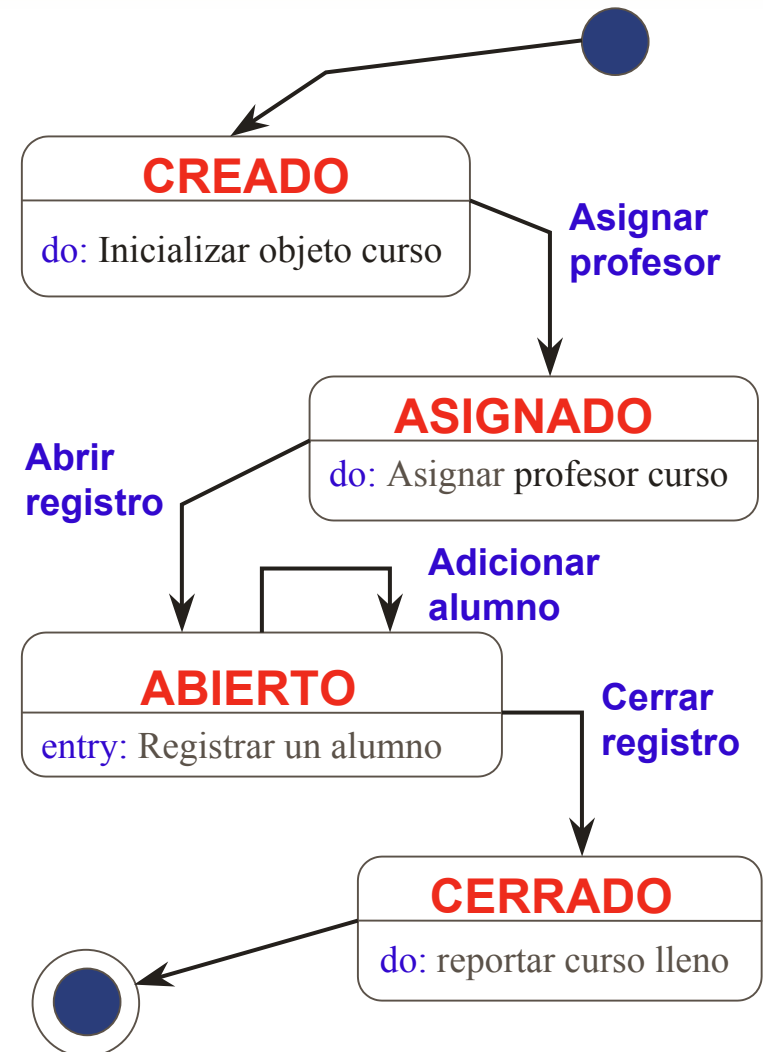
Acciones de la Transición

- Estas acciones son operaciones asociadas a la transición de un estado a otro:
 - Toma una cantidad insignificante de tiempo completarla.
 - Considerada ininterrumpible
 - El evento que produce la transición puede generar el envío de otro evento.



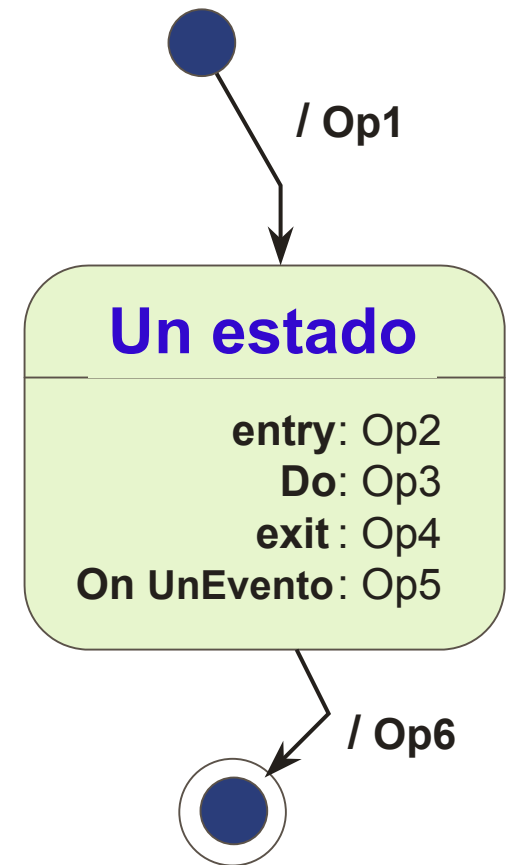
Acciones de Estado

- La acción de entrada (*entry:*) se ejecuta de manera instantánea y atómica al entrar en el estado.
- La acción de salida (*exit:*) se ejecuta al salir del estado.
- La acción sobre el evento interno (*on:*) se ejecuta al ocurrir un evento que no conduce a otro estado. Un evento interno no entraña la ejecución de las acciones de entrada/salida.



Puntos de ejecución de las Operaciones

- Existen seis puntos, cuyo orden es :
 - La acción asociada a la transición de entrada (Op1)
 - La acción de entrada de estado (Op2)
 - La actividad en el estado (Op3)
 - La acción de salida del estado (Op4)
 - La acción asociada a los eventos internos (Op5)
 - La acción asociada a la transición de salida del estado (Op6)



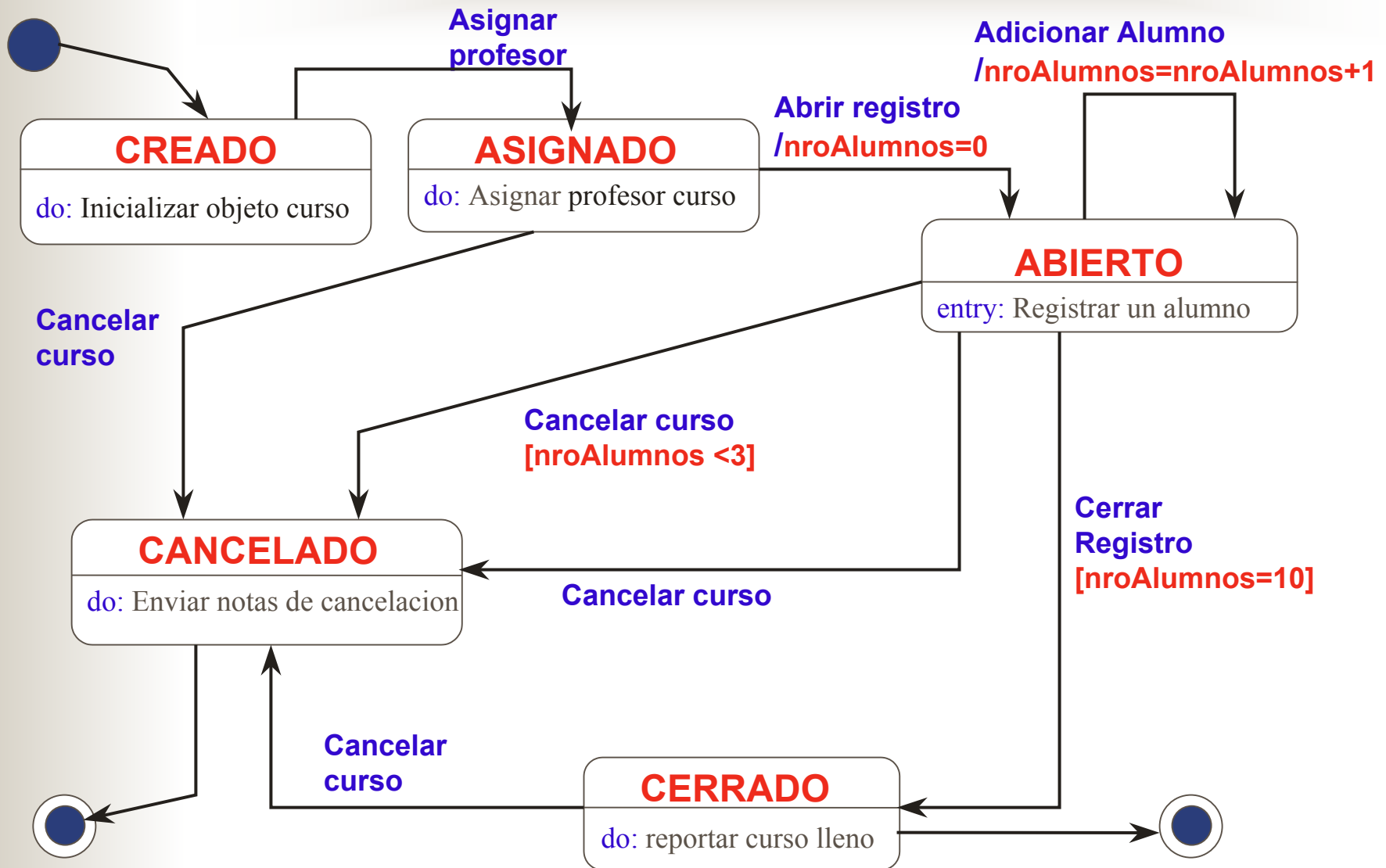


Diagrama de Estados

Ejemplar
Codigo : integer
Cantidad : integer
Estado : integer
exponer ()
reservar ()
borrarReserva ()
prestar ()
devolver ()
retirar ()
reponer ()

