

OOAD Usando UML

Código del Curso: CY450

Versión 5.0

Objetivos del Curso

Al finalizar este curso ud. debería ser capaz de:

- Describir la necesidad del estudio del Análisis y Diseño Orientado a Objetos.
- Conocer los conceptos básicos del paradigma orientado a objetos.
- Describir la importancia del modelado UML.
- Describir el uso y funcionamiento de los diagramas UML en el aspecto estructural.
- Describir el uso y funcionamiento de los diagramas UML en el aspecto de comportamiento.
- Conocer las diferencias entre la especificación UML 1.x y la especificación UML 2.0
- Describir el uso y funcionamiento de los diagramas de la especificación UML 2.0 en el aspecto estructural y de comportamiento.
- Aprender a manejar herramientas de modelado.

Modelado Básico

Introducción a UML

Objetivos de Aprendizaje

Al finalizar esta unidad ud. debería ser capaz de:

- Describir la necesidad del Análisis y Diseño Orientado a Objetos (Object Oriented Analysis and Design – **OOAD**).
- Explicar los principios de la orientación a objetos.
- Describir el trabajo del Lenguaje Unificado de Modelado (Unified Modeling Language – **UML**).
- Describir la importancia del modelado en UML.

Metodologías de Desarrollo de Software

- Orientado a Aspectos.
- Literal.
- Estructurado.
- Orientado a Objetos.
- Patrones.

Tipos de Datos Abstractos

- Un tipo de dato abstracto es un modelo matemático.
- En orientación a objetos, el modelado de la vida real resulta en tipos de datos abstractos.
- Algunos se refieren a la programación orientada a objetos como programación de tipos de datos abstractos.
- En orientación a objetos, tratamos principalmente con datos; las operaciones son entonces definidas sobre esos datos.

Abstracción

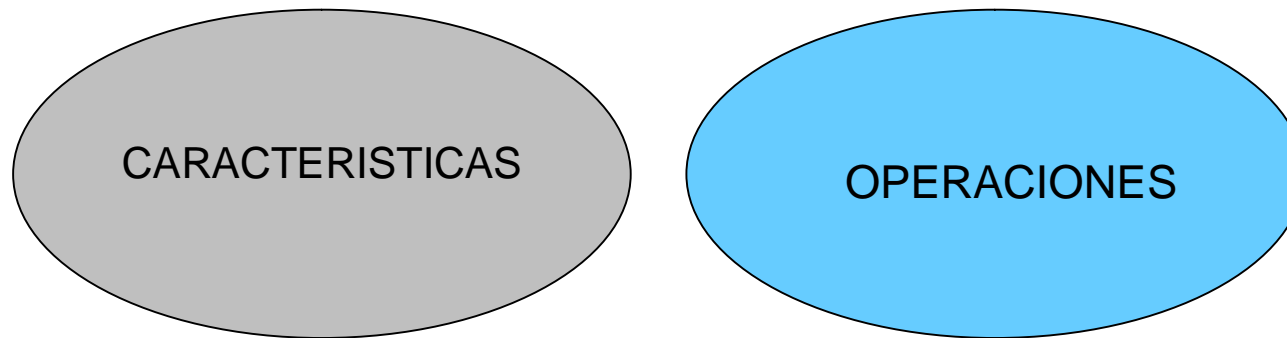
- Aspectos esenciales del problema.
- Extrae la información esencial y relevante.
- En forma simple, la abstracción corta lo innecesario.

Encapsulamiento

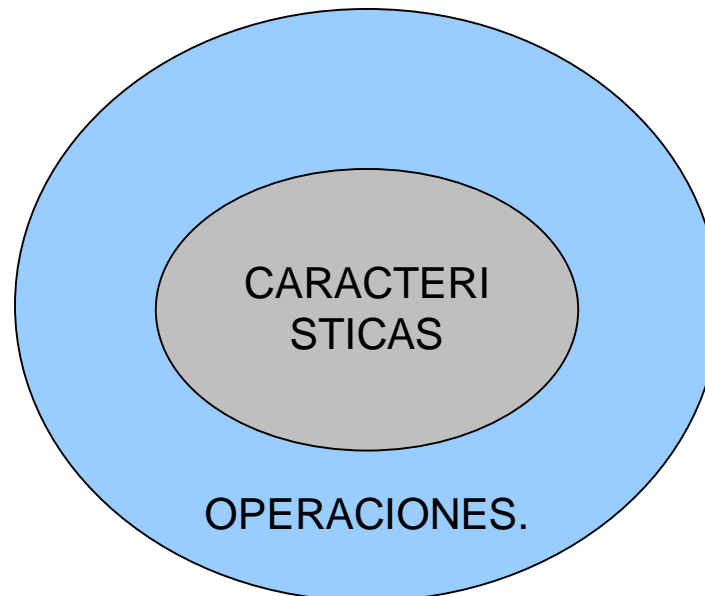
- Los Atributos de un objeto y los métodos que trabajan con ellos son ambos mantenidos en una cápsula.
- El encapsulamiento es lograda a través del ocultamiento de la información.
- Se refiere a construir una cápsula, en el caso de una barrera conceptual, alrededor de una colección de cosas.

Ejemplos de Encapsulamiento

Paradigma Procedimental



Paradigma Orientado a objeto

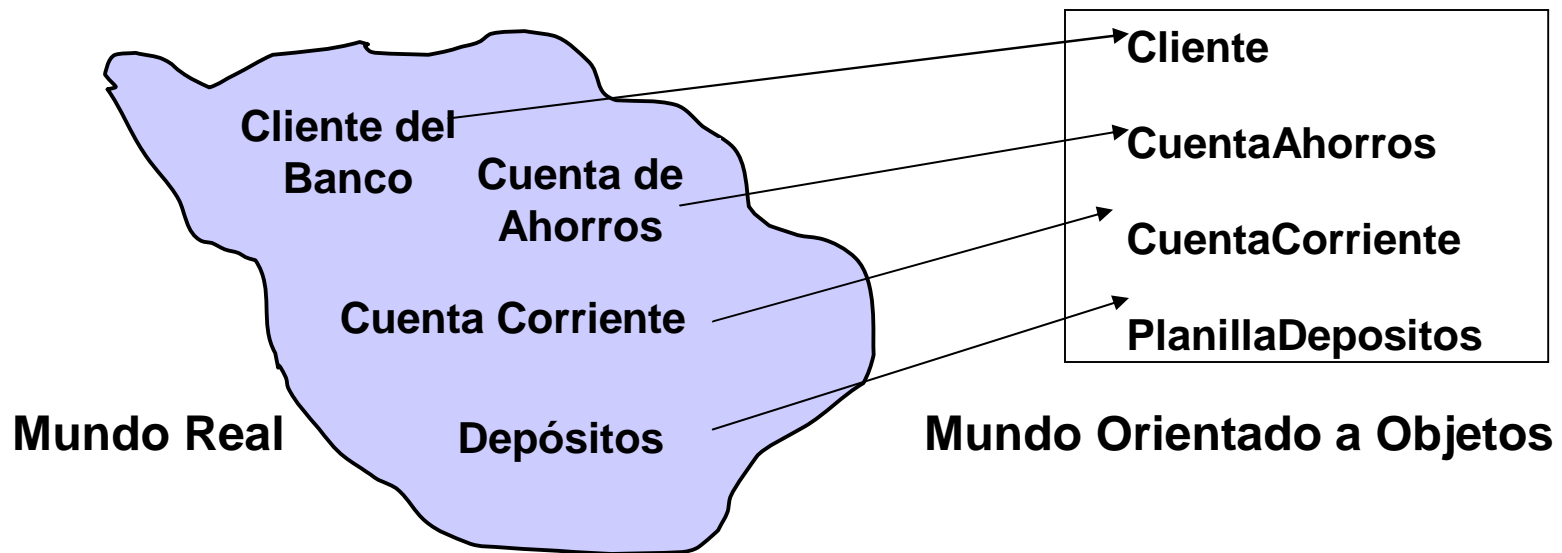


Ocultamiento de la Información

- Detalles esenciales de cómo inicializar y acceder el componente, sin conocer los detalles de la implementación.
- Teniendo encapsulados los miembros de datos y las operaciones, el usuario sólo necesita saber cómo acceder a las operaciones.
- Tanto la información y la implementación de las operaciones se ocultan al usuario.

Clases

- Las Clases son tipos de datos agregados definidos por el usuario, representan las entidades del mundo real.



Objetos

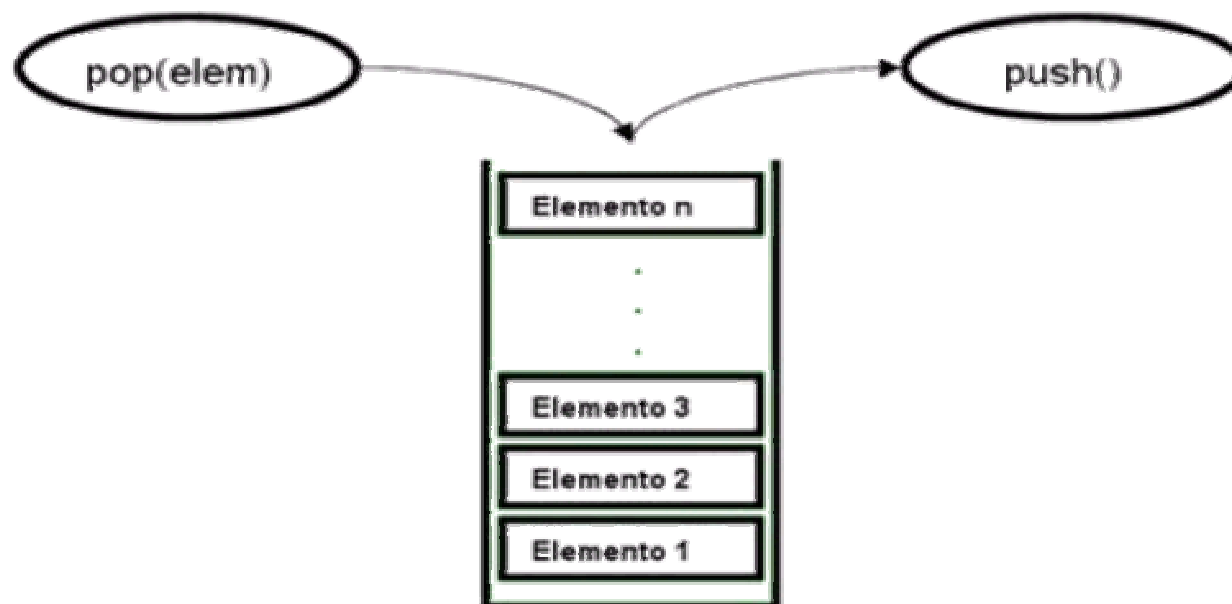
- Un objeto es una instancia de una clase.
- Un objeto tiene lo siguiente:
 - Estado, dado en un instante de tiempo.
 - Identidad, la cual es única entre los objetos de la misma clase.
 - Comportamiento, el cual es usualmente la información y las funciones de esa clase.

Interfaz e Implementación

- Una interfaz es la manera por la cual la entidad es presentada al mundo real.
- Un punto importante a notar es que como usuario de un servicio, estamos interesados en lo que la entidad provee y no en cómo la entidad satisface nuestra demanda.
- El “que” describe la interfaz de la entidad y el “como” provee la implementación de la entidad.

Métodos

- Los métodos son una parte de la declaración de la clase. Los métodos son algoritmos usados por la clase para implementar la tarea prometida por la interfaz.
- Así, en el ejemplo de la pila, la tarea o responsabilidad de añadir (push) un elemento a la pila o sacar (pop) es llamada método.

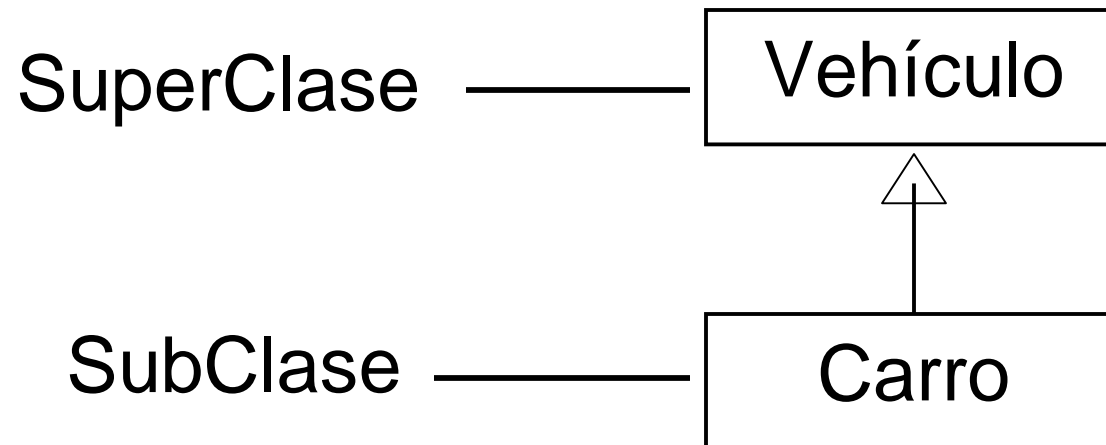


Mensajes

- Un mensaje es el mecanismo usado por un objeto para hacer que una tarea sea llevada a cabo por otro objeto.
- Típicamente un mensaje contiene lo siguiente:
 - El nombre del objeto que recibe el mensaje.
 - El nombre del mensaje.
 - Los argumentos pasados al objeto receptor (esto es opcional y también podemos pasar objetos mismos como argumentos).

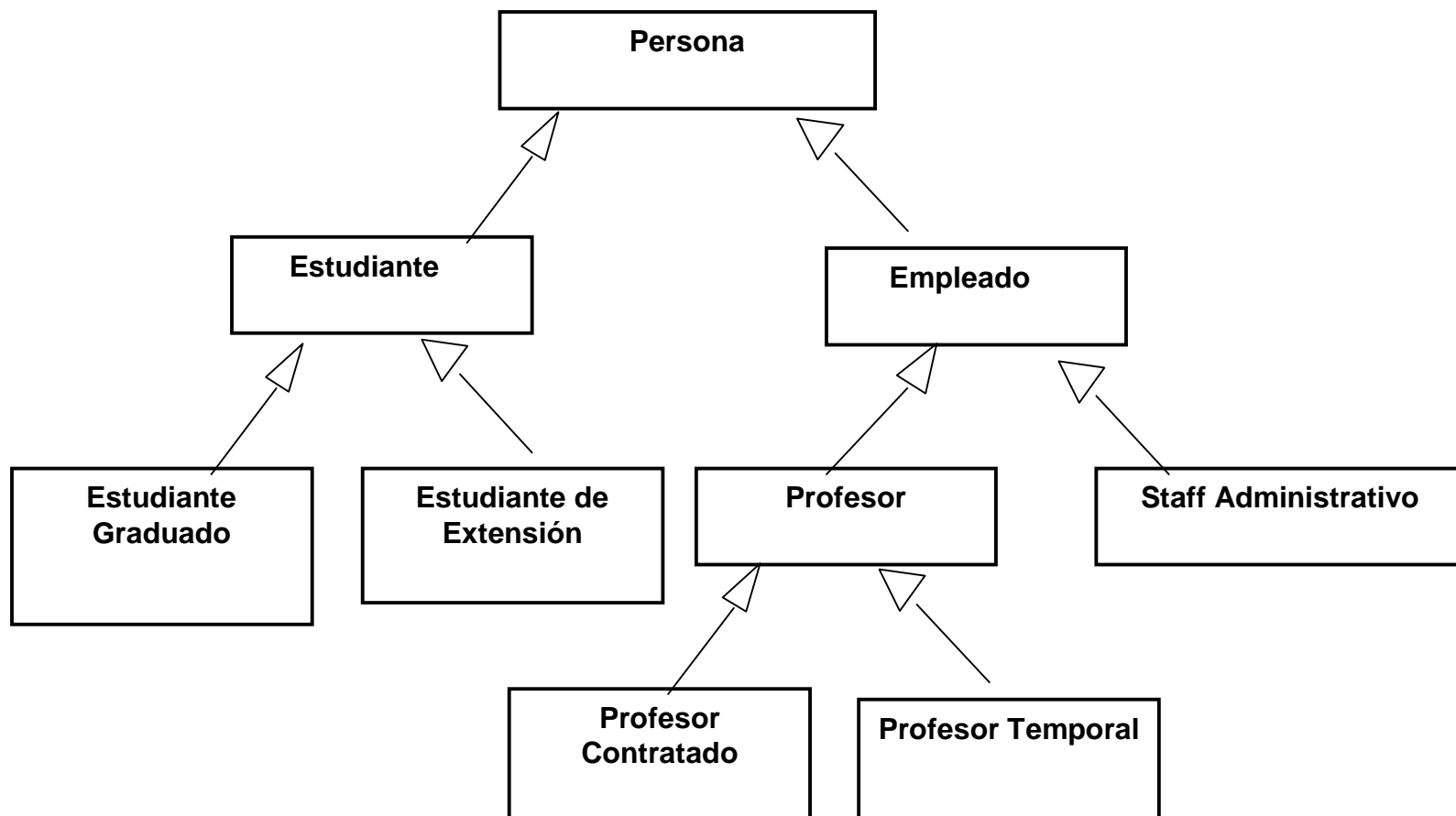
Herencia

- Por herencia, queremos decir que alguna entidad X comparte la estructura y comportamiento de alguna otra entidad Y.
- En este contexto, la entidad Y es llamada la “super clase” y la entidad X es llamada “subclase”.



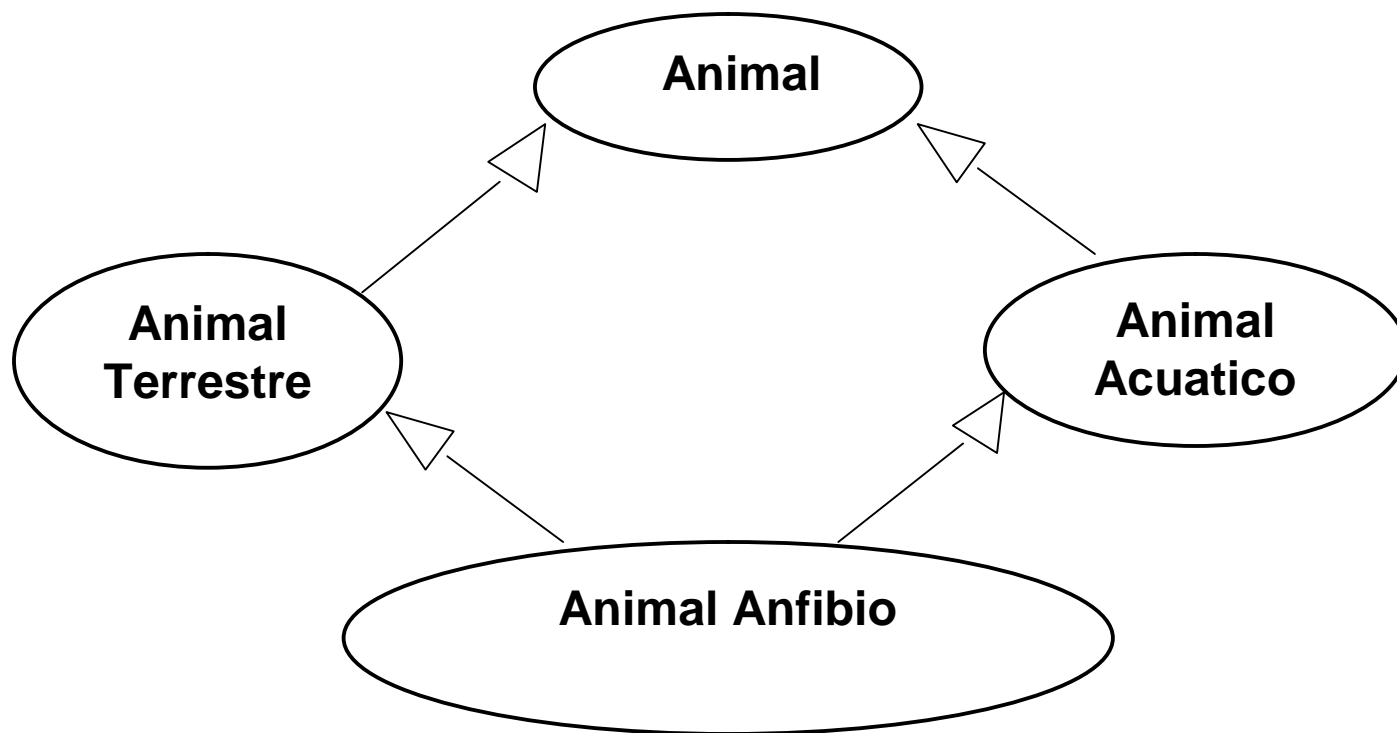
Herencia...1

- La jerarquía de herencia especifica la relación “es un” entre la subclase y la super clase.



Herencia...2

- También es posible para una subclase heredar de dos súper clases.



Agregación

- Una entidad contiene otra entidad.
- La relación “tiene un” también es referida como la relación de contención.



Polimorfismo

- Polimorfismo significa la posibilidad para hacer que una operación tenga un comportamiento diferente en diferentes situaciones.
- El comportamiento depende de los tipos de datos usados en la operación.
- Por ejemplo, las fechas se pueden crear de algunas de las siguientes maneras:
 - Dar tres diferentes enteros, día, mes y año (15, 8, 1947).
 - Dar una cadena (“15/08/1947”), donde las tres partes pueden ser extraídas.
 - Dar un entero largo (“19471508”), donde las tres partes pueden ser extraídas.

Tipo, Rol y Paquetes

- Un tipo (type) es un estereotipo de una clase que es usado para especificar un dominio de objetos junto con sus operaciones.
- Un rol describe el comportamiento de una entidad en un contexto particular.
- Un paquete es un mecanismo para agrupar elementos.

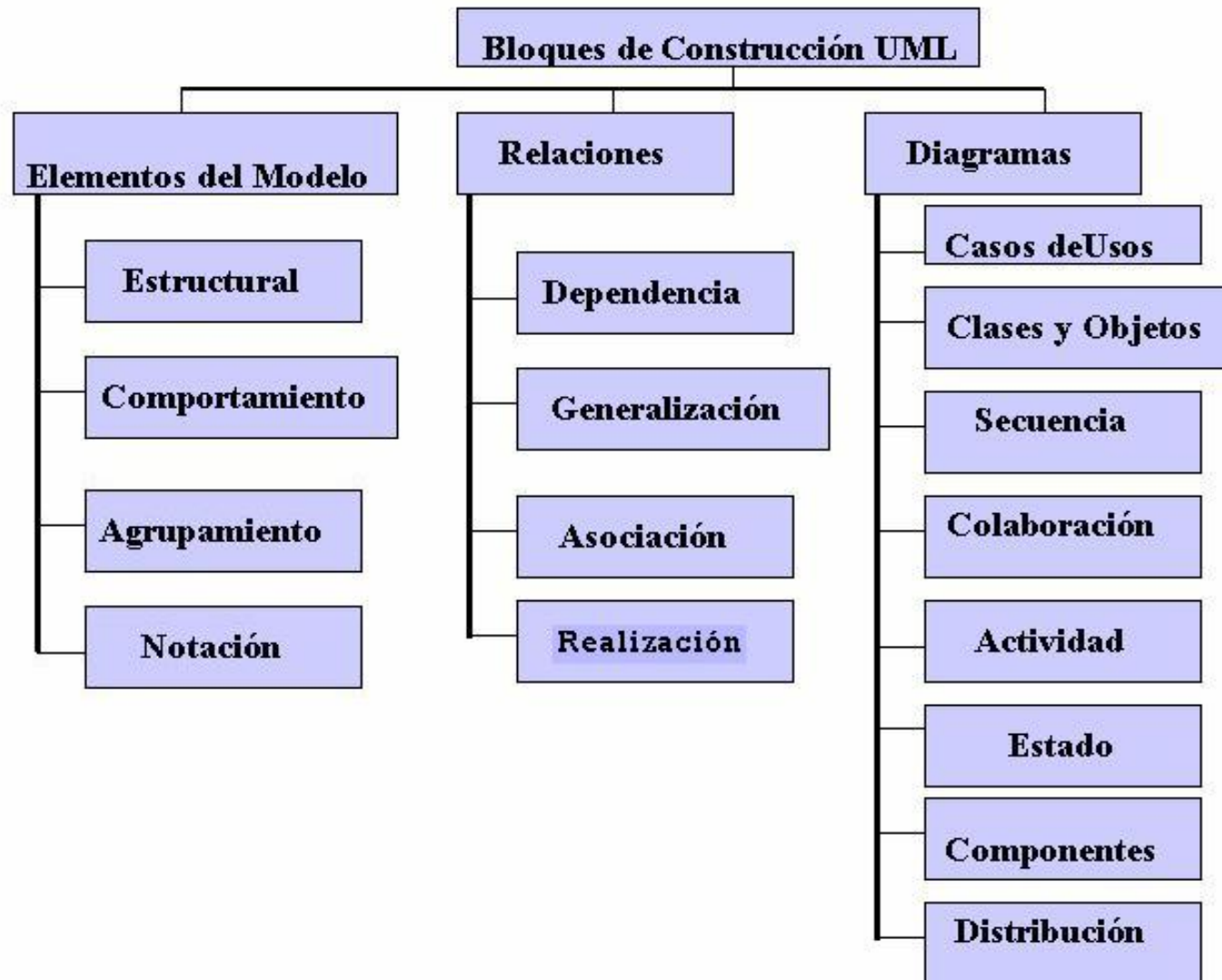
Lenguaje Unificado de Modelado

- El Lenguaje Unificado de Modelado (The Unified Modeling Language - UML) es un lenguaje para especificar, visualizar, construir y documentar artefactos de sistemas de software, como también para modelar negocios y otros sistemas de no-software.
- El UML provee lenguaje de modelado de aplicaciones para:
 - Modelado del proceso de negocio con casos de uso.
 - Modelado de clases y objetos.
 - Modelado de Componentes.
 - Modelado de distribución e implementación (deployment).

Modelo UML

- UML está conformado por tres importantes elementos, llamados: Los ***bloques de construcción*** del UML, las ***reglas*** que nos permiten juntar los bloques de construcción y algunos ***mecanismos comunes*** que se aplican a través del UML.
- Bloques de Construcción:
 - **Elementos del modelo.**
 - **Relaciones.**
 - **Diagramas.**

Modelo UML...1



Elementos Estructurales del Modelo UML

- **Elementos Estructurales:**
 - Clase.
 - Interfaz.
 - Colaboración.
 - Caso de Uso.
 - Clase Activa.
 - Componente Nod.

Elementos de Comportamiento, Agrupamiento y Anotación

- **Elementos de Comportamiento**
 - Tenemos dos elementos de comportamiento, la interacción y máquina de estado.
- **Elementos de Agrupamiento**
 - El único elemento de agrupamiento es el paquete.
- **Elementos de Anotación**
 - Estos son la parte explicativa del UML.

Relaciones

- **Dependencia**
 - Esta es una relación semántica entre dos elementos.
- **Asociación**
 - La asociación describe la relación estructural.
- **Generalización**
 - El hijo hereda los atributos, operaciones, relaciones y semánticas de una clase generalizada, el padre.
- **Realización**
 - Esta es una relación semántica entre clasificadores.

Diagramas

- **Diagrama de Clases** – Incluye un conjunto de clases, interfaces, colaboraciones y sus relaciones.
- **Diagrama de Objetos** – Muestra un conjunto de objetos y sus relaciones.
- **Diagrama de Casos de Uso** – Consiste de los casos de usos, actores y sus relaciones.
- **Diagrama de Secuencia** – Es un tipo de diagrama de interacción. Muestra el tiempo de ordenamiento de los mensajes.
- **Diagrama de Colaboración** – Este es otro tipo de diagrama de interacción. Revela la organización estructural de objetos que envían y reciben mensajes.
- **Diagrama de Estados** – Se usa para representar máquinas de estado. Consiste de estados, transiciones, eventos y acciones.
- **Diagrama de Actividad** – Es un tipo especial de diagrama de estados. Muestra el flujo de una actividad a otra en el sistema.
- **Diagrama de Componentes** – Es usado para mostrar la organización y dependencias entre el conjunto de componentes del sistema.
- **Diagrama de Despliegue e Implementación (Deployment)** – Muestra los nodos y los componentes que residen en ellos.

Reglas

- Las reglas ayudan a los bloques de construcción de UML a especificar un modelo bien-formado.
- Un modelo es bien formado si es auto consistente y sincroniza con todos los otros modelos relacionados.
- UML incluye reglas semánticas para:
 - **Nombres:** Los nombres son usados para identificar elementos del modelo, relaciones y diagramas.
 - **Visibilidad:** La visibilidad especifica la manera en la cual los nombres pueden ser vistos y usados por otros.
 - **Alcance:** El alcance es el contexto en el cual los nombres residen.
 - **Ejecución:** La ejecución es la manera en la cual el modelo dinámico es simulado.
 - **Integridad:** La integridad especifica como los elementos del modelo se relacionan unos con otros.

Mecanismos Comunes

- **Especificación.**
- **Adornos.**
- **División Común.**
- **Mecanismos de Extensibilidad:**
 - Estereotipos.
 - Valores etiquetados.
 - Restricción.

Resumen

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

- Describir la necesidad del Análisis y Diseño Orientado a Objetos (Object Oriented Analysis and Design – **OOAD**)
- Explicar los principios de la orientación a objetos.
- Describir el trabajo del Lenguaje Unificado de Modelado (Unified Modeling Language – **UML**).
- Describir la importancia del modelado en UML.

Modelado del Comportamiento con Casos de Uso

Objetivos

Al finalizar esta unidad ud. debería ser capaz de:

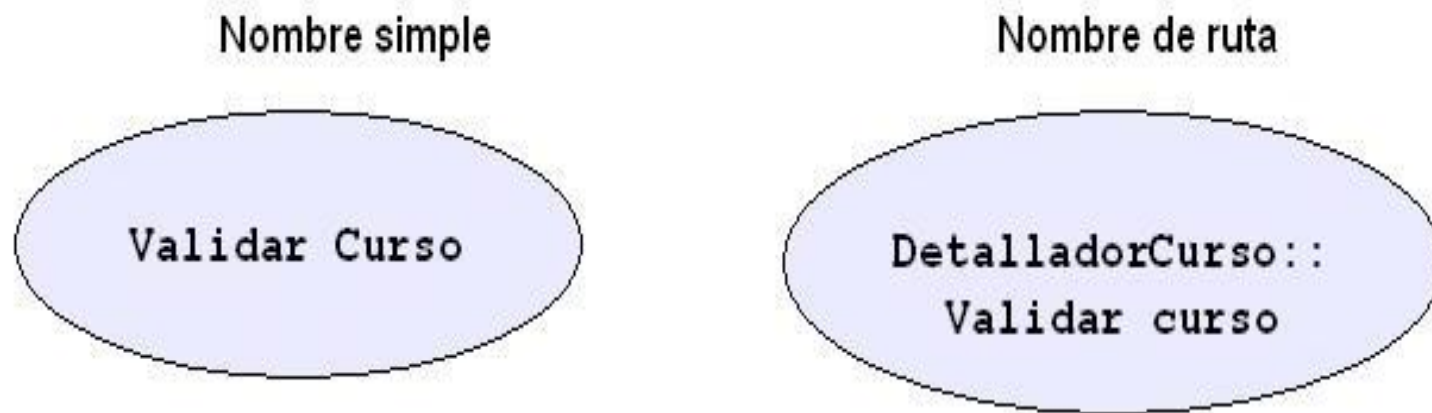
- Definir la importancia de los casos de uso.
- Definir cada uno de los elementos involucrados en un caso de uso.
- Describir el análisis para un caso de uso.
- Describir la importancia y los elementos de los diagramas de actividad.
- Elaborar un diagrama de casos de uso.
- Elaborar diagramas de actividad.

Casos de Uso

- Un caso de uso es una descripción de un conjunto de acciones que un sistema realiza con respecto a un actor particular interesado en el sistema.
- Un caso de uso es una interacción que un usuario u otro sistema tiene con el sistema que se está diseñando.
- Un caso de uso implica interacciones entre actores y el sistema.
- Formado por una combinación de un verbo y un sustantivo.

Casos de Uso...1

- En UML, un caso de uso se representa como una elipse con el nombre dentro de la elipse.



- Los casos de usos son útiles por que ayudan a:
 - Descubrir requerimientos.
 - Capturar la necesidad del usuario al enfocarse en la tarea que el usuario necesita realizar con el sistema.
 - Formular planes de prueba del sistema.
 - Controlar el desarrollo iterativo.

Elementos Típicos de los Casos de Uso

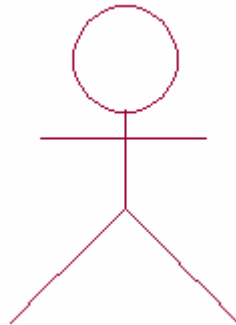
- Generar número Identificación.
- Procesar préstamo.
- Imprimir reporte.
- Ver estadísticas.
- Asignar curso.
- Crear documento.
- Publicar libro.

Actores

- Es el usuario o el sistema que tiene interés en el sistema que se diseña.
- Un **actor** es una entidad que interactúa con el sistema, no la abstracción del sistema.
- Representan los usuarios externos quienes usarán el sistema que se está diseñando.
- El actor puede ser otro sistema.
- El actor representa el rol que los usuarios del sistema juegan mientras interactúan con el sistema.

Actores...1

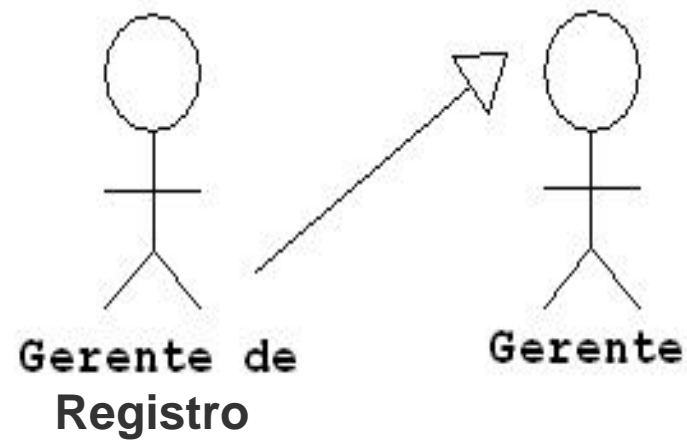
- Representación de un actor en UML:



- Para los casos de uso anteriores, se pueden identificar los siguientes actores:
 - Gerente Registro.
 - Oficinista del Banco.
 - Oficinista.
 - Responsable de actualizar Inventario.
 - Adjudicador de Cursos.
 - Autor.
 - Editor Libro.

Actores...2

- Ejemplo de Herencia de Actores



Casos de Uso en UML

- Un caso de uso ilustra lo que hace un sistema, pero no especifica el modo en que el sistema implementa el caso de uso.
- El comportamiento de un caso de uso se describe con la descripción de un flujo de eventos.
- El flujo de eventos puede ser mostrado usando :
 - Texto estructurado informal.
 - Seudocódigo.
 - Texto estructurado con precondiciones y postcondiciones.

Formato Estructurado

Formato para Flujo de Eventos Estructurado

Nombre			
Caso de Uso:		Id Caso de Uso:	
Actores:			
Descripción:			
Casos de Uso Relacionados:			
Entradas:		Salidas:	
Curso Típico			
Acción del Actor		Respuesta del Sistema	
Curso Alternativo			
Acción del Actor		Respuesta del Sistema	
Pre-condiciones:			
Post-condiciones:			

Colaboraciones

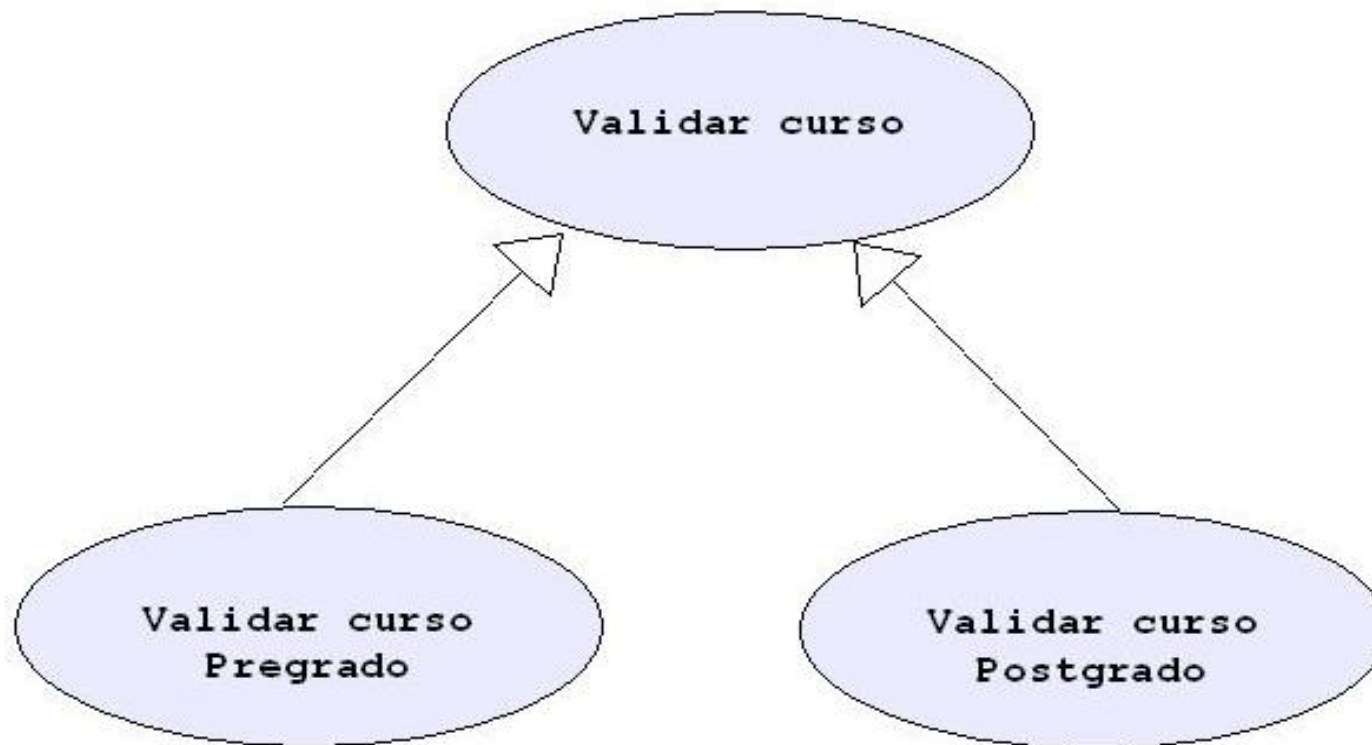
- La **colaboración** es usada en UML para modelar la sociedad de elementos, tanto estática como dinámica, que ayuden a implementar el comportamiento de un caso de uso.
- Un caso de uso es realizado por colaboración.
- La colaboración de un caso de uso se representa como una elipse punteada, y la realización del caso de uso se describe como una línea punteada con la flecha apuntando hacia el caso de uso que realiza.



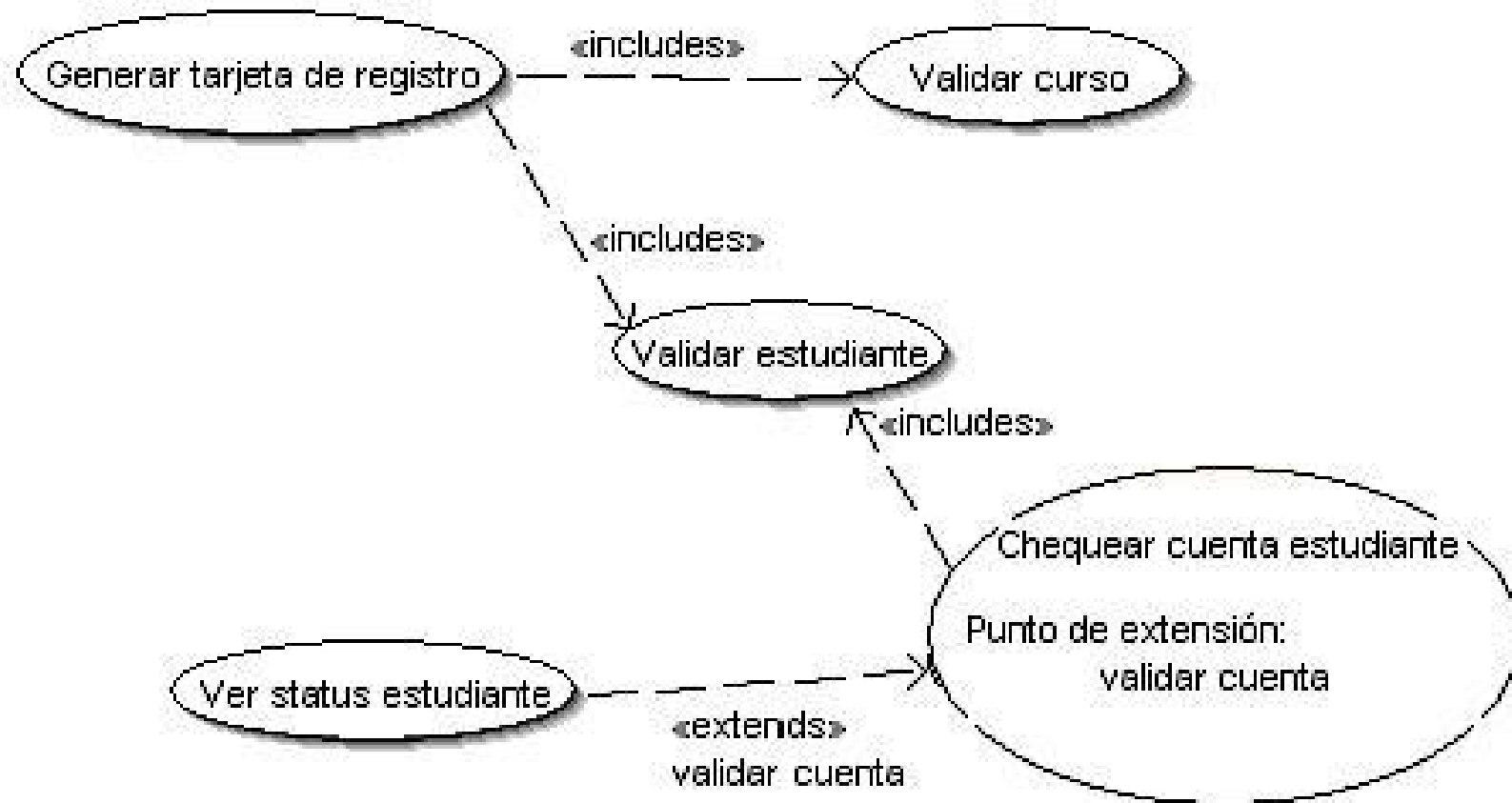
Relaciones entre Casos de Uso

- Se pueden usar tres tipos de relaciones con casos de uso. Estas son:
 - Generalización.
 - Include (inclusión).
 - Extend (extensión).
- En la relación de **Generalización** los casos de uso pueden heredar de otros casos de uso, del mismo modo que las clases. El caso de uso hijo hereda el comportamiento y significado del caso de uso padre.

Casos de Uso de Generalización

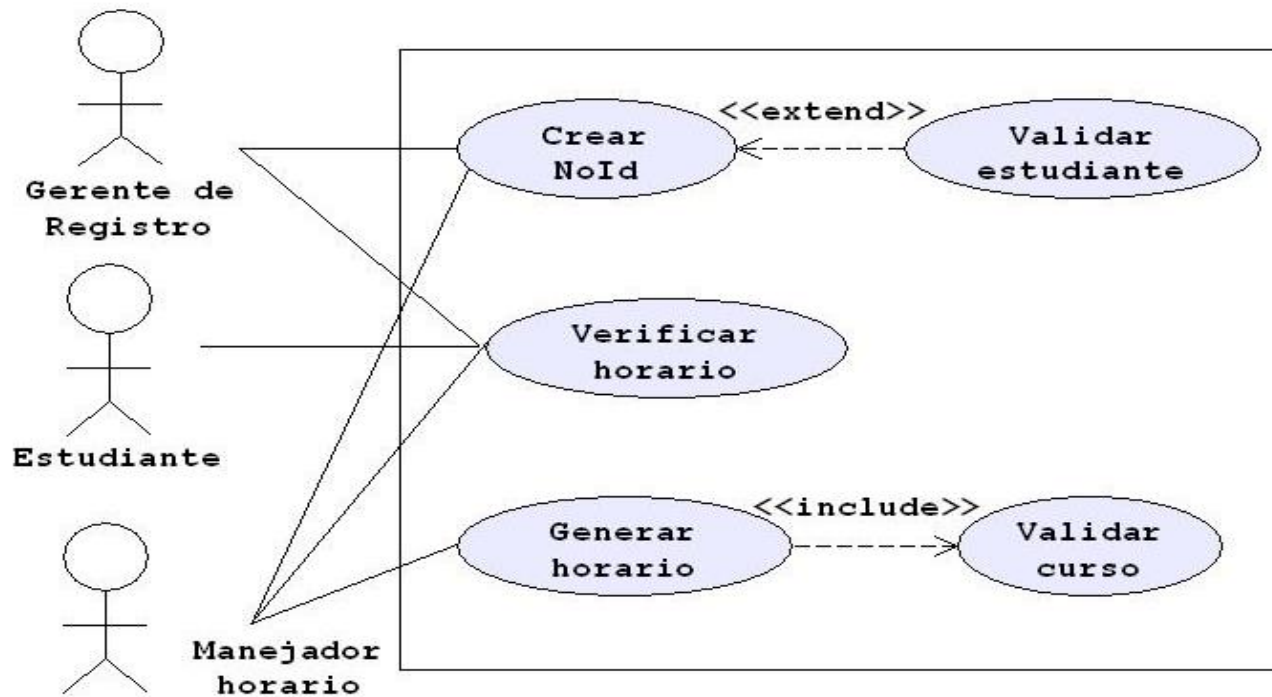


Relaciones de Include y Extend en Casos de Uso



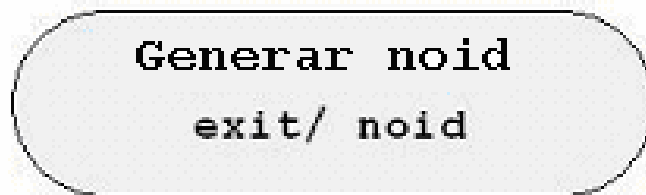
Diagramas de Casos de Uso

- Un diagrama de caso de uso describe un conjunto de caso de uso, actores y sus relaciones.
- Un diagrama de caso de uso consiste en:
 - Un conjunto de casos de uso.
 - Actores.
 - Asociaciones de comunicación entre actores y casos de uso.
 - Relaciones entre casos de uso.



Estado de Actividad y Acción

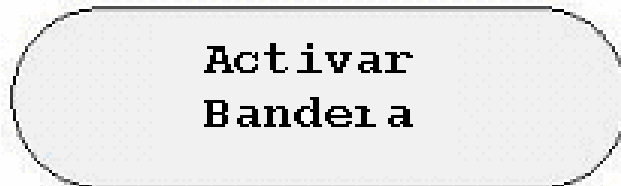
- Los estados de actividad denotan algunas actividades a ser realizadas. Pueden o no ser atómicos.
- Se observa que los estados de actividad son representados como operaciones, las que a su vez pueden tener otros estados de actividad o estados de acción.
- Los estados de actividad y acción son denotados gráficamente como una forma de pastilla.



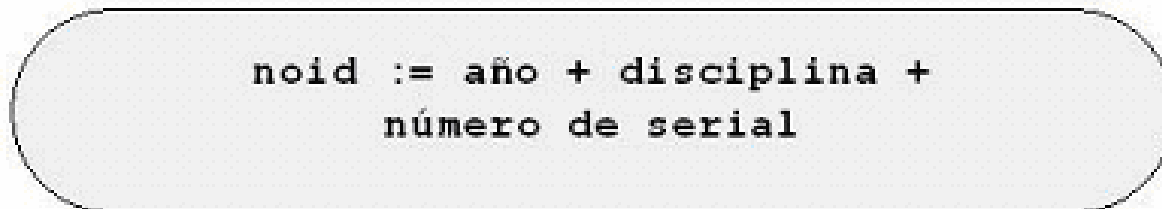
Estado de Actividad y Acción...1

- Los estados de acción son atómicos (no es posible una descomposición posterior) por naturaleza.
- Un estado de acción puede ser una acción simple o una expresión.

Acción simple

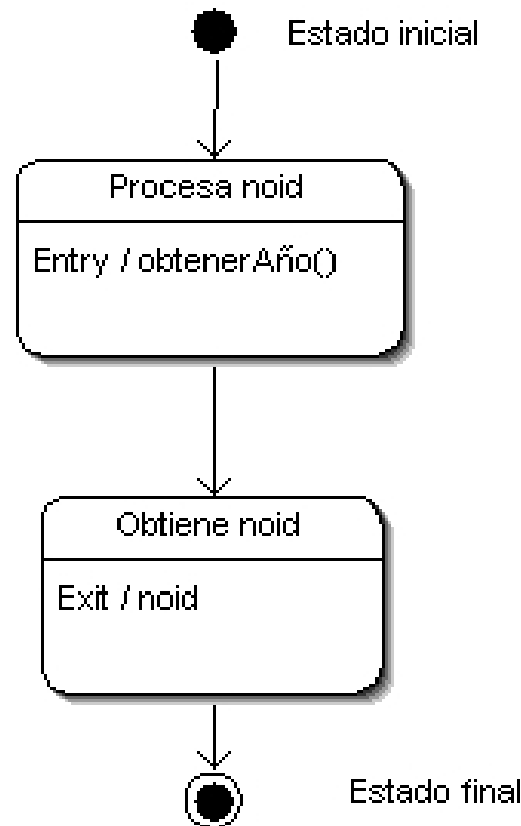


expresión



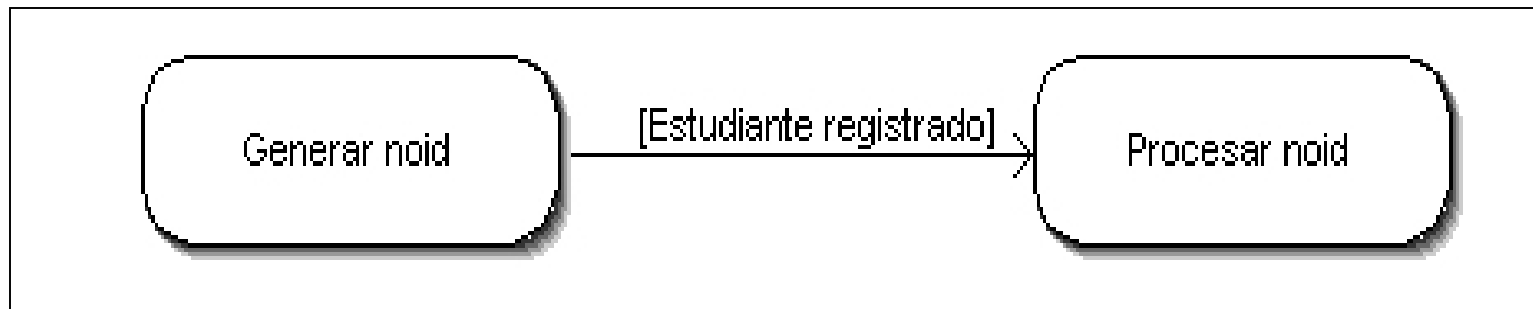
Transiciones

- Las transiciones se usan para mostrar el cambio de una actividad o acción a otra en el sistema y se muestra usualmente con una línea dirigida con una flecha apuntando hacia el siguiente estado.



Condición de Guardia

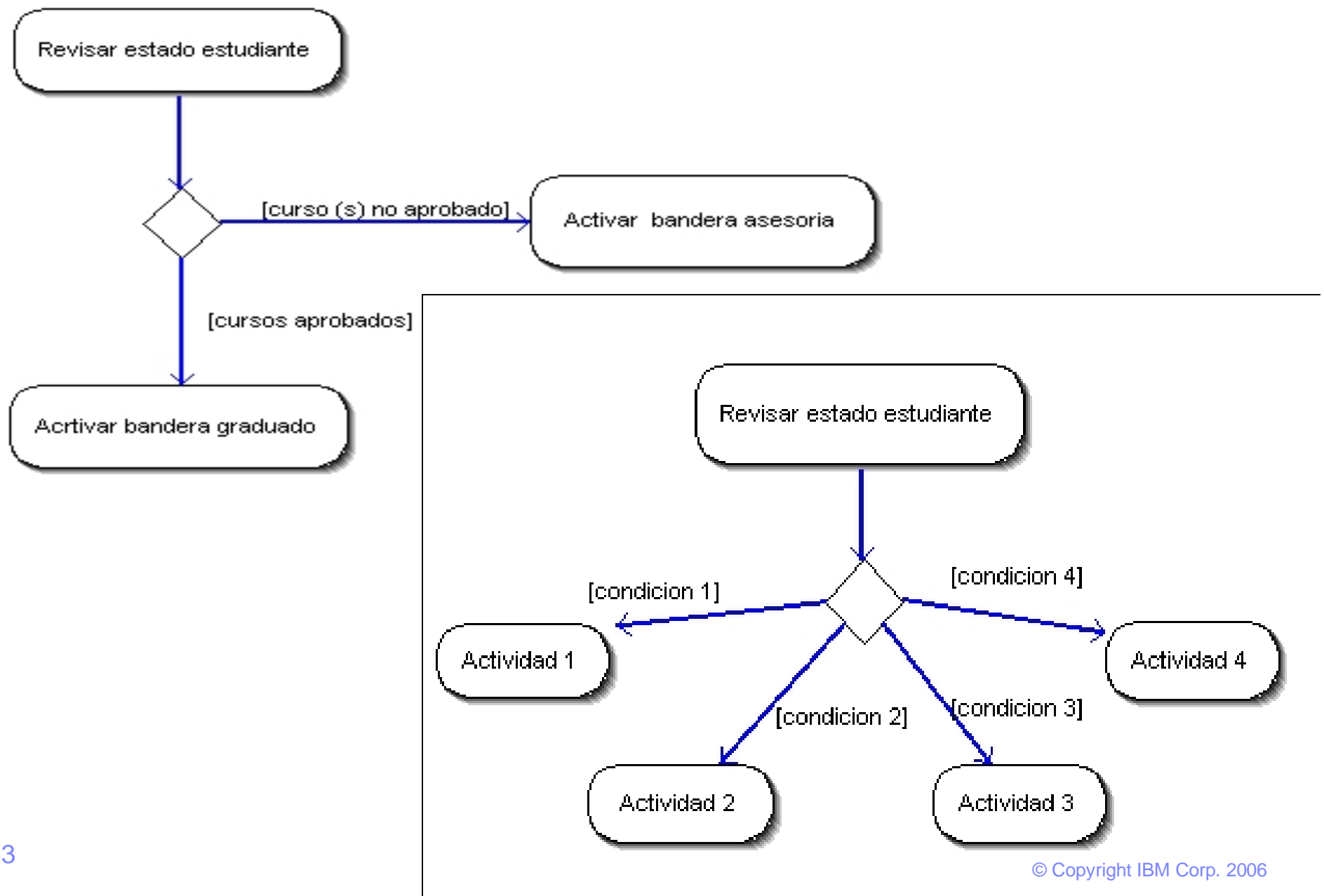
- Una condición de guardia puede ser asignada a una transición para restringir su ejecución.



Decisión o Bifurcación

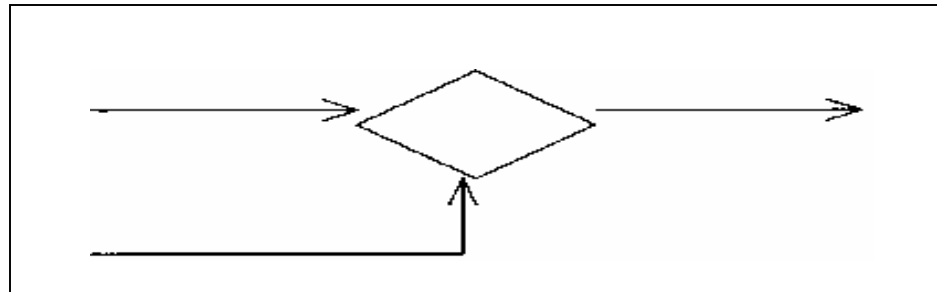
- La bifurcación es similar a una construcción `if ... then ... else`.
- Cuando se muestra la bifurcación en diagramas de actividad, se indica la condición bajo la cual tuvo lugar la transición. La bifurcación se denota con un diamante.
- Cada rama de la bifurcación tiene una condición de guardia.

Decisión o Bifurcación...1



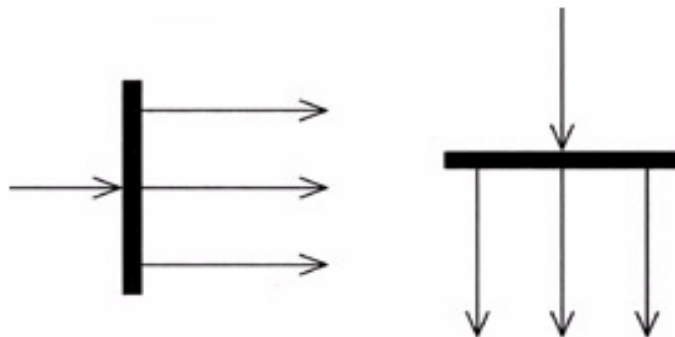
Punto de Fusión y Concurrency

- La fusión es el lugar donde dos rutas alternativas se unen y continúan juntas

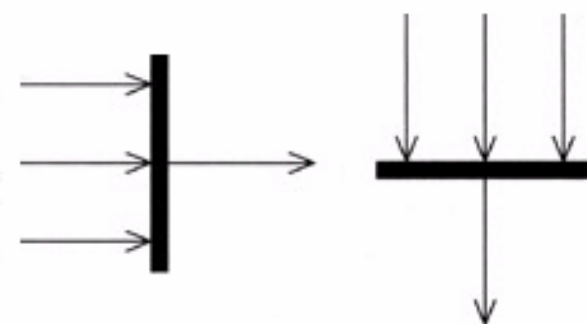


- UML usa la ramificación (forking) y unión (joining) para manejar un flujo concurrente.

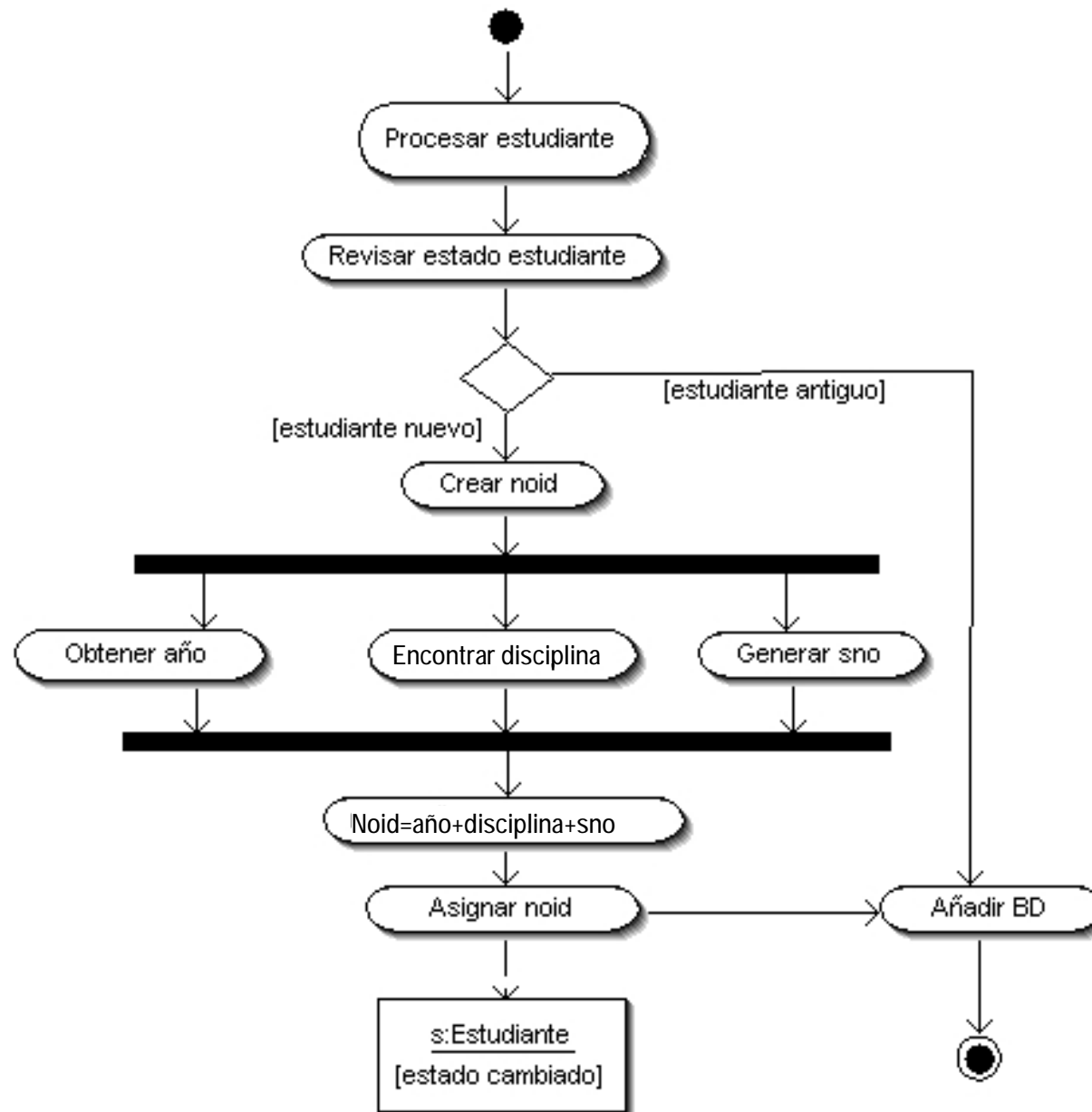
Forking



Joining



Ejemplo de Diagrama de Actividad



Indicaciones

Se utilizan para enviar y recibir señales traducidas como eventos.

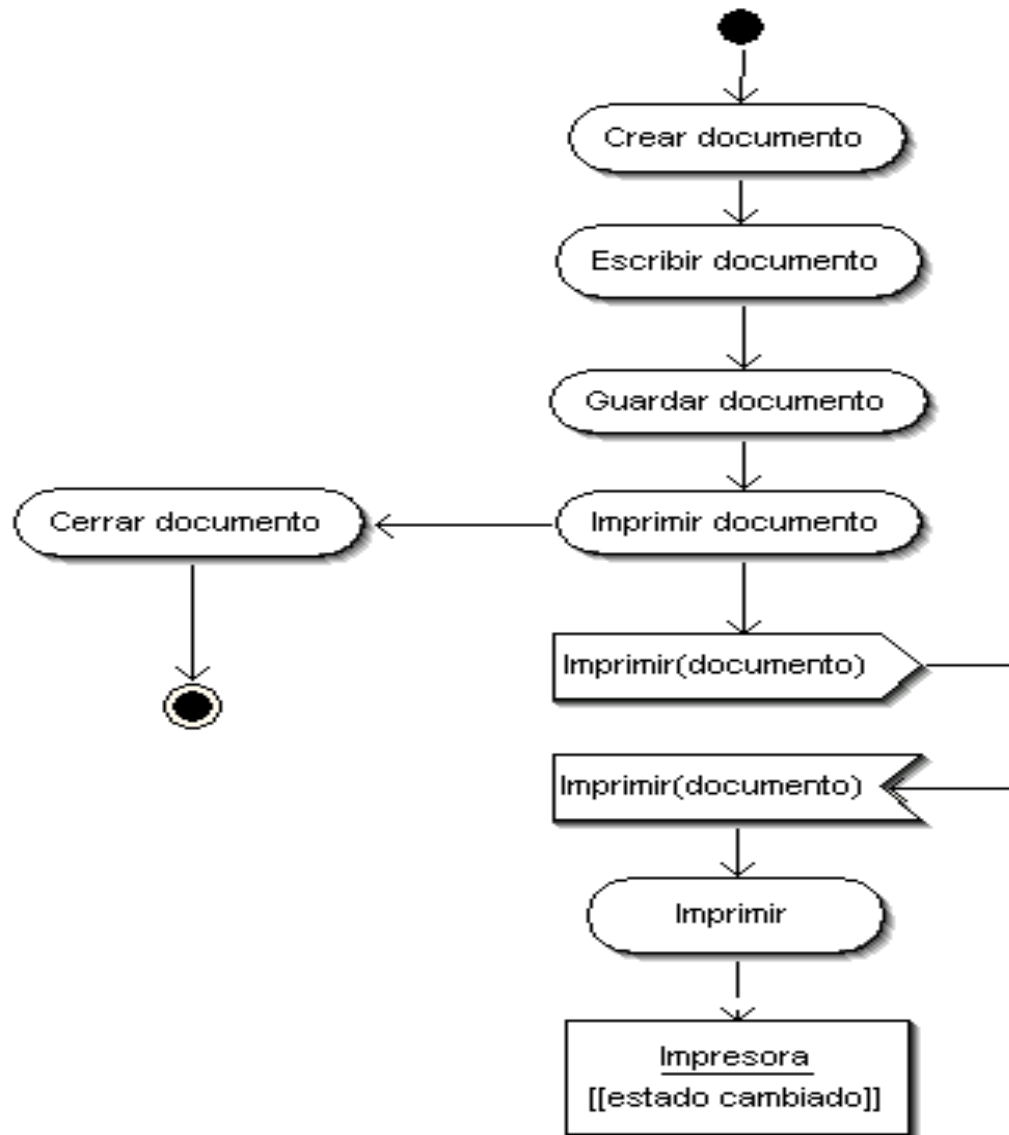


Diagrama de Actividad con Carriles

- Se puede categorizar el diagrama de actividad en grupos.
- Los carriles son descritos con una línea vertical, la cual separa cada grupo.

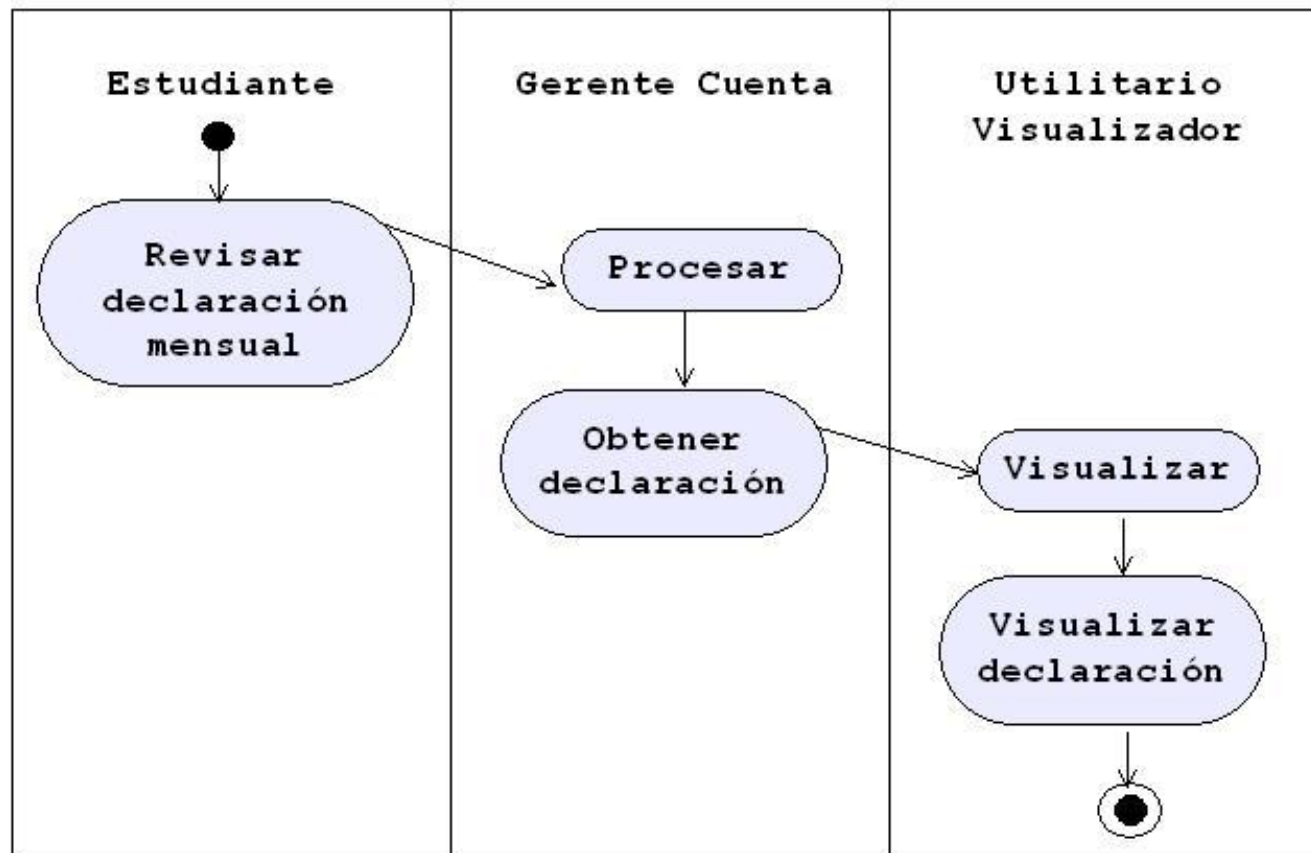
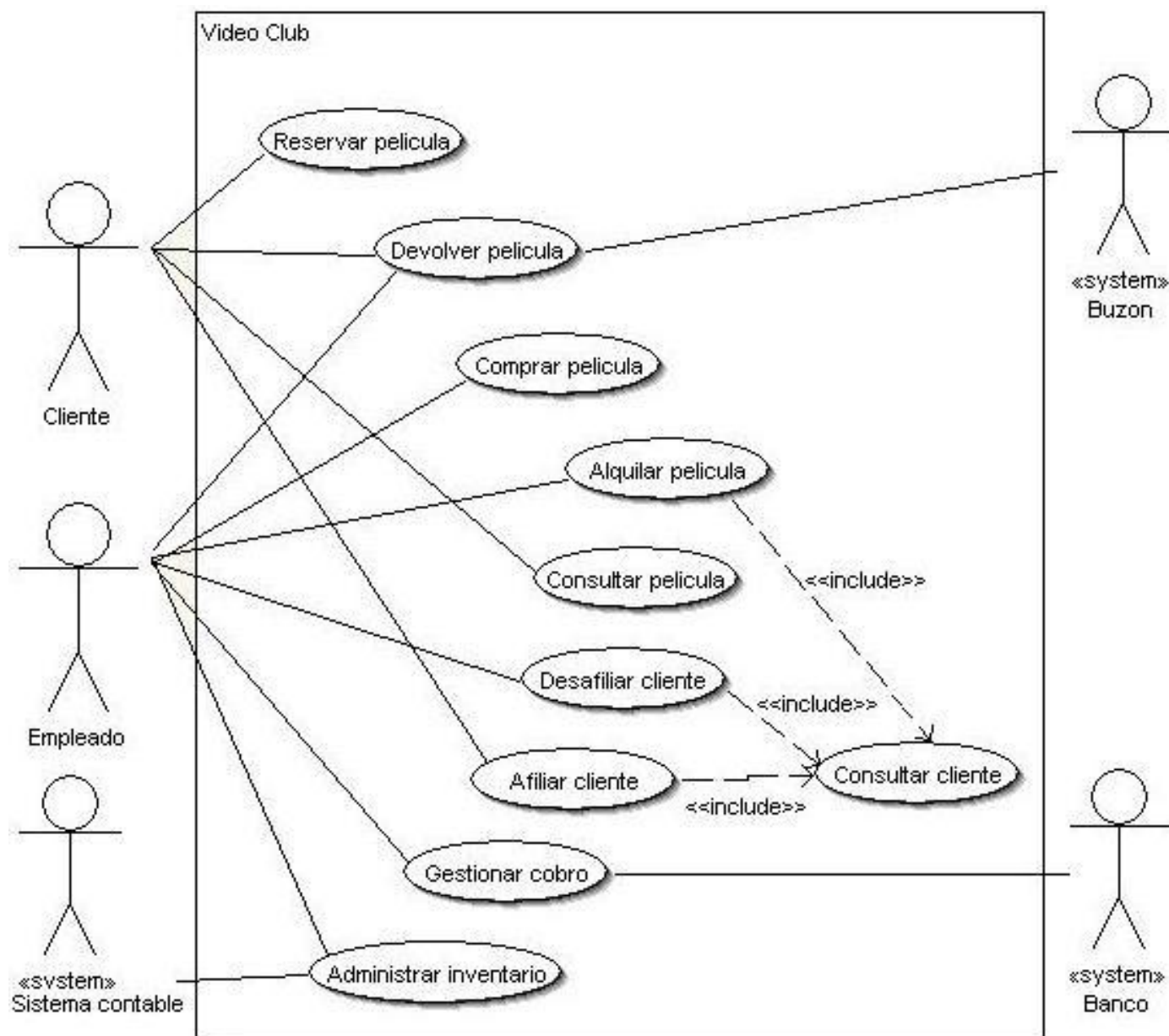


Diagrama de Casos de Uso para la Tienda de Video



Flujo de Eventos

Nombre C.U:	Alquilar película	Id C.U	CU-001
Actores:	Empleado		
Descripción:	Permite que el cliente alquile una película		
Casos de Uso Relacionados:	Consultar película, Consultar cliente		
Entradas:	Datos de la película y del cliente	Salidas:	Registro de la película alquilada, la existencia de la película, la ficha del cliente y el sistema contable
Curso Típico:			
Acción del Actor		Respuesta del sistema	
1.- El empleado introduce en el sistema la cedula del cliente.			
		2.- El sistema verifica que el cliente este afiliado.	
3.- El empleado introduce el código de la película que el cliente escogió.			
		4.- El sistema verifica que la película exista o que este disponible.	
		5.- El sistema muestra los datos de la película.	
		6.- El sistema solicita la verificación de la película	
7.-El empleado realiza la confirmación.			
		8.- El sistema muestra el monto a cancelar.	

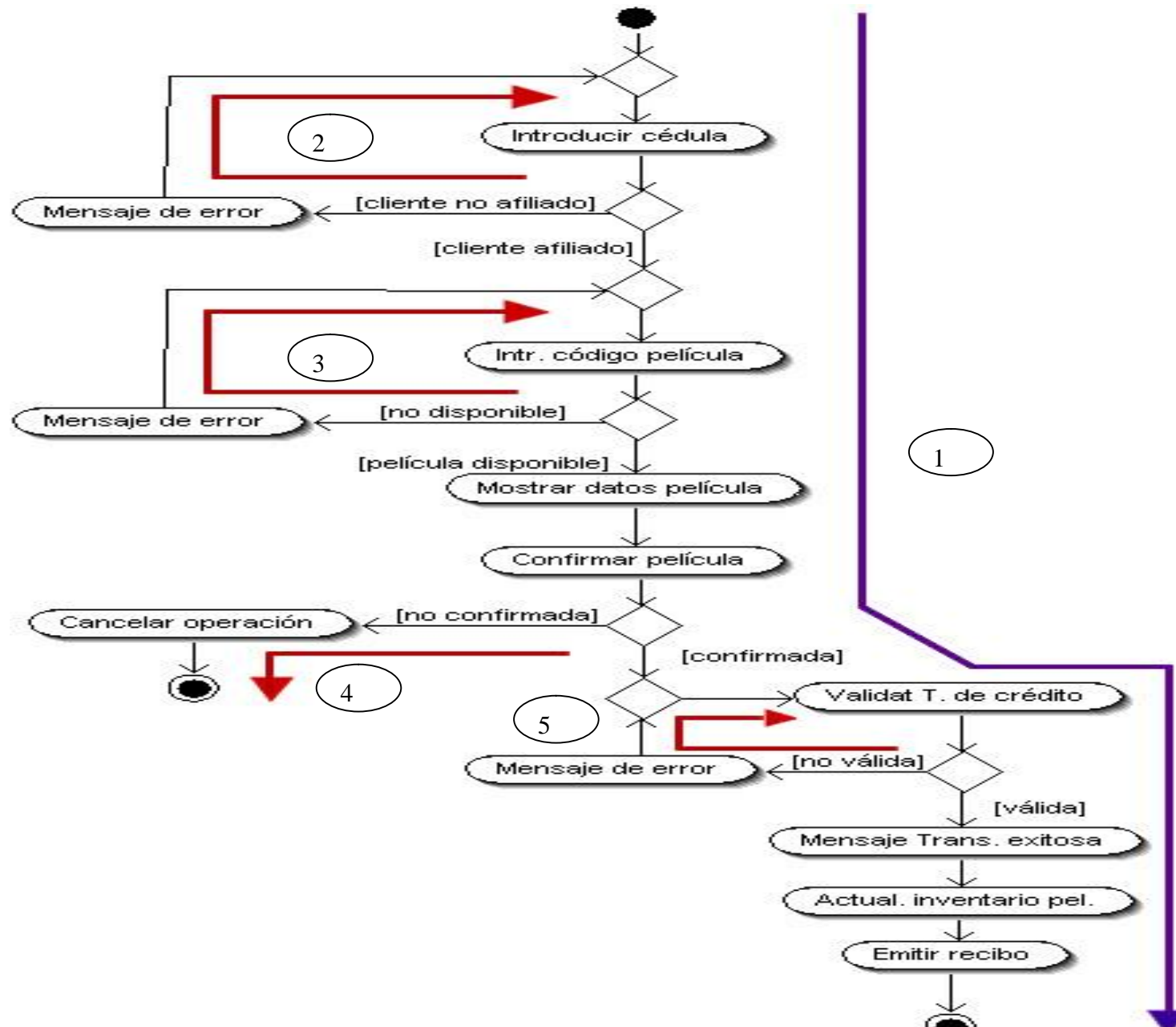
Flujo de Eventos...1

Curso Típico:	
Acción del Actor	Respuesta del sistema
	9.- El sistema pide los datos de la tarjeta de crédito.
10.-El empleado introduce los datos de la tarjeta.	
	11.- El sistema verifica con el banco la valides de la tarjeta de crédito.
	12.- El sistema muestra mensaje de transacción exitosa.
	13.- El sistema actualiza el inventario de películas
	14.- El sistema actualiza la BD del sistema contable.
	15.-El sistema genera un recibo de pago al cliente.
Curso Excepcional # 1: El cliente no aparece en el sistema	
Precondición:	En el paso 1 del flujo típico, el empleado introduce una cédula
Acción del Actor	Respuesta del Sistema
	1.- Falla la búsqueda del cliente
	2.- El sistema muestra un mensaje, indicando que el usuario no esta afiliado.
3.- El caso de uso continua en el paso 1 del flujo principal	

Flujo de Eventos...2

Curso Excepcional # 2: La película esta alquilada	
Precondición:	En el paso 3 del flujo típico, el empleado introduce el código de la película.
Acción del Actor	Respuesta del Sistema
	1.- El sistema encuentra que la película no esta disponible
	2.- El sistema muestra en pantalla un mensaje, indicando que la película no esta disponible.
3.- El caso de uso continua en el paso 3 del flujo del flujo típico	
Curso Excepcional # 3: Tarjeta de crédito invalida	
Precondición:	En el paso 11 del flujo típico, el sistema verifica la valides de la tarjeta de crédito.
Acción del Actor	Respuesta del Sistema
	1.- Falla la transacción con el sistema bancario.
	2.- El sistema muestra un mensaje de error donde indica que la tarjeta es invalida
3.- El caso de uso continua en el paso 10 del flujo típico.	

Diagrama de Actividad



Resumen

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

- Definir la importancia de los casos de uso.
- Definir cada uno de los elementos involucrados en un caso de uso.
- Describir el análisis para un caso de uso.
- Describir la importancia y los elementos de los diagramas de actividad.
- Elaborar un diagrama de casos de uso.
- Elaborar diagramas de actividad.

Modelado Estructural Básico

Objetivos

Al finalizar esta unidad ud. debería ser capaz de:

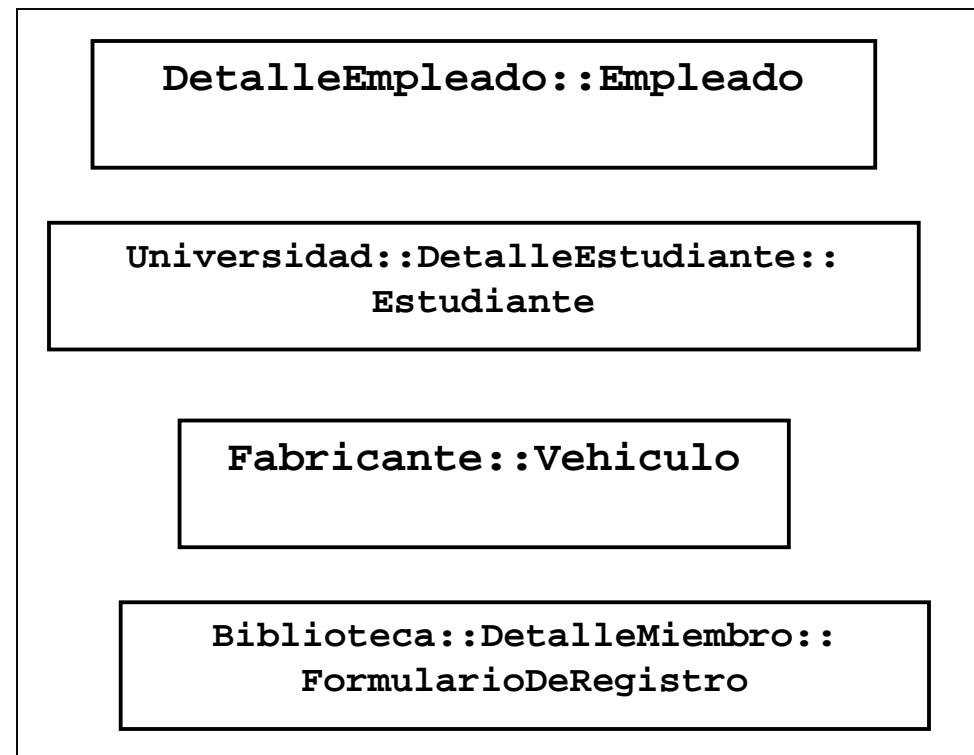
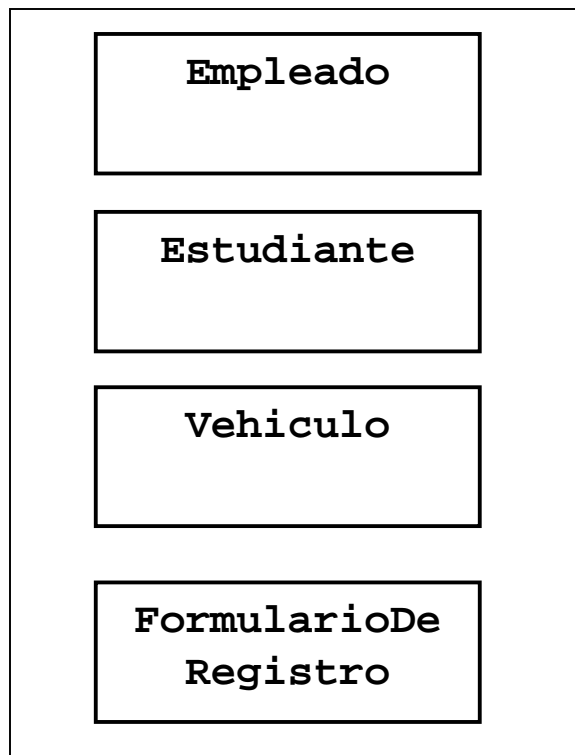
- Describir el modelado estructural de UML en forma básica.
- Explicar los principios de la orientación a objetos.
- Describir clases y sus relaciones.
- Conocer los mecanismos comunes que pueden ser aplicados a las clases.
- Elaborar diagramas de clases.

Modelado Estructural Básico

- El modelado estructural de UML especifica cómo está constituido el sistema completo.
- Revelan partes concretas del sistema entero.
- El modelado estructural trata con clases (abstracciones) y objetos (realizaciones concretas de las abstracciones).
- Los diagramas con los que estaremos tratando bajo el modelado estructural son los Diagrama de Clases y los Diagramas de Objetos.

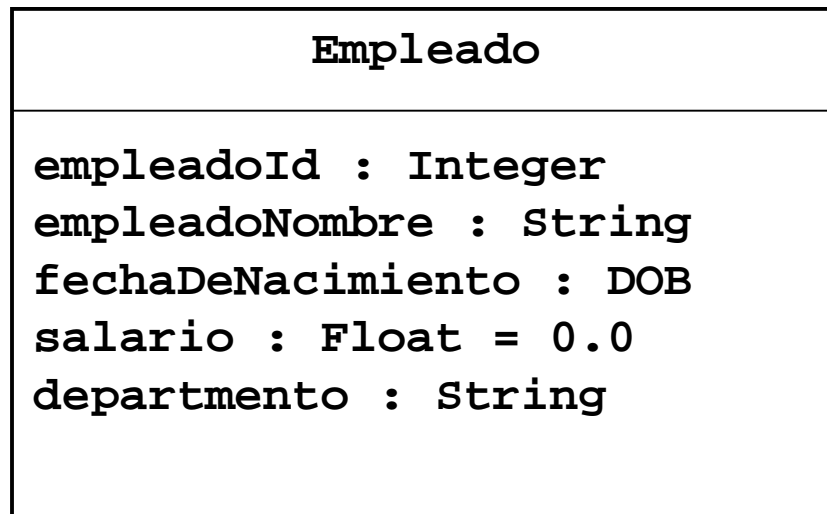
Nombres de Clases

- Cada clase se representa con un rectángulo y tiene un nombre. Puede ser Simple, en donde sólo el nombre de la clase se especifica.
- El signo `::` se usa para separar el nombre del paquete (izq) de la clase (derecha).



Atributos de las Clases

- Los atributos de una clase son aquellos que caracterizan a la clase.
- Los atributos especifican un rango de valores que los objetos de la clase pueden tener.
- Sintaxis completa:
[visibilidad] nombre [multiplicidad] [: tipo] [= valor inicial]
[{propiedad de la cadena}]



Nombre de clase

Atributos

Operaciones de las Clases

- Son las funcionalidades o comportamiento que una clase ofrece.
- Es la operación la que es invocada por un objeto para obtener un servicio de otro objeto.
- Sintaxis:

[visibilidad] nombre [(lista de parámetros)] [: tipo del retorno]
[{propiedad de la cadena}]

Empleado
empleadoId : Integer empleadoNombre : String ...
colocarDetalleEmpleado() obtenerDetalleEmpleado() colocarSalario(salario : Float) obtenerSalario() : Float

Nombre de Clase



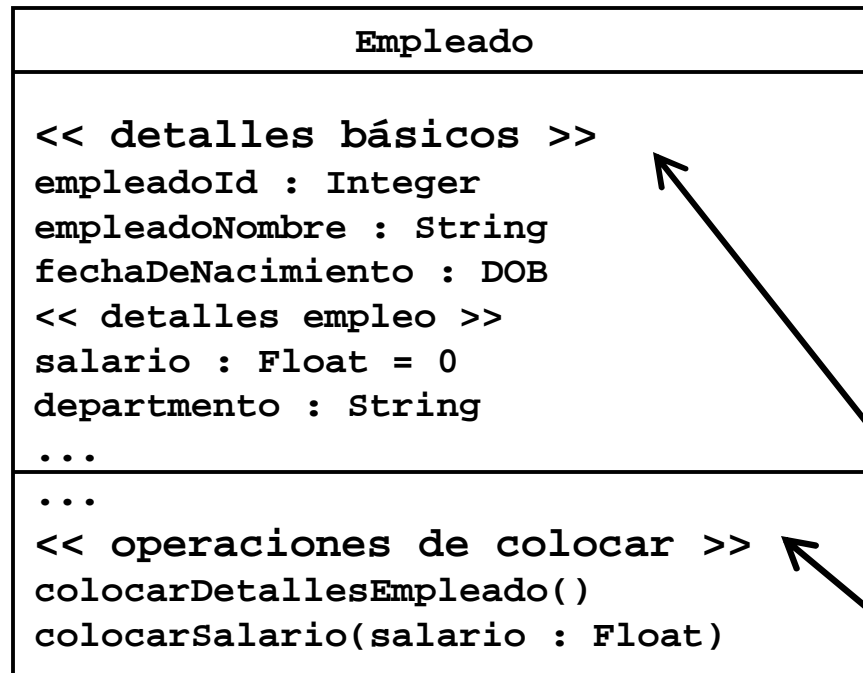
Atributos



Operaciones



Estereotipo de la Clase

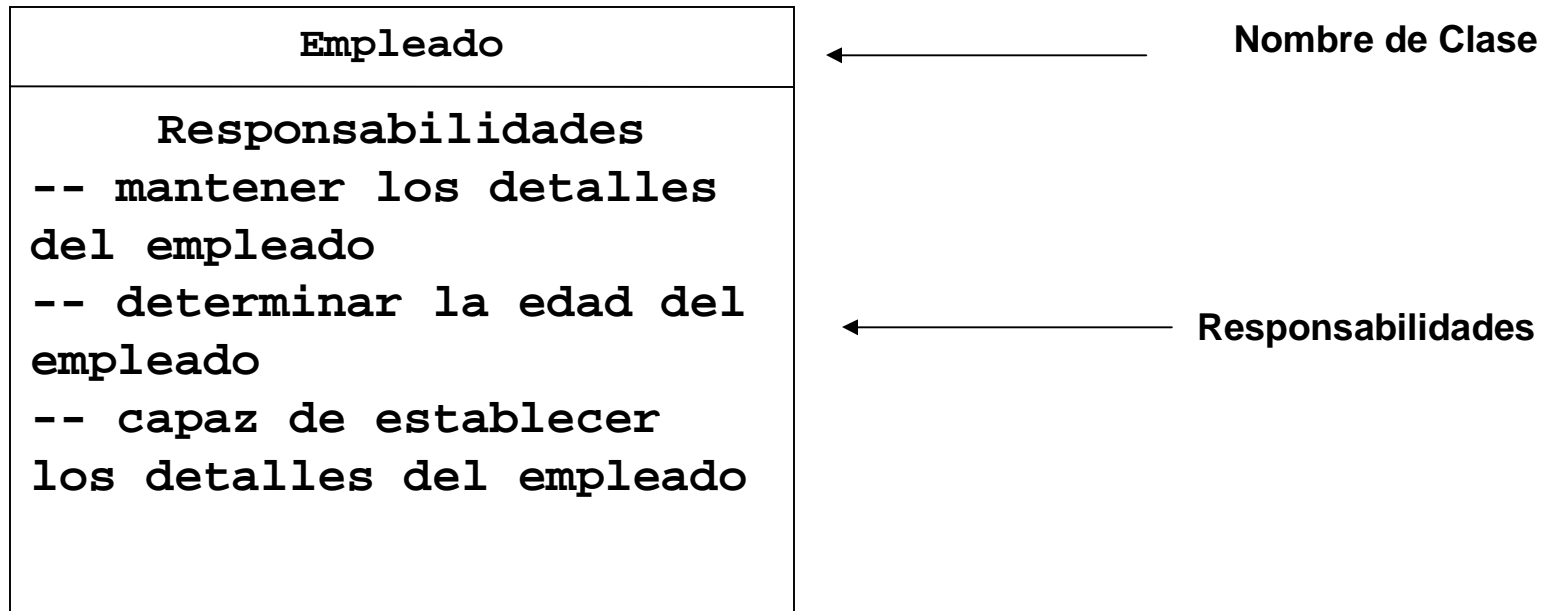


- Permiten categorizar con un nombre los atributos y operaciones
- Ayudan a agrupar operaciones o atributos similares

Estereotipos

Responsabilidades de la Clase

- Son aquellas establecidas por una clase y que es capaz de realizar.
- Se muestran en forma textual, en simple español.



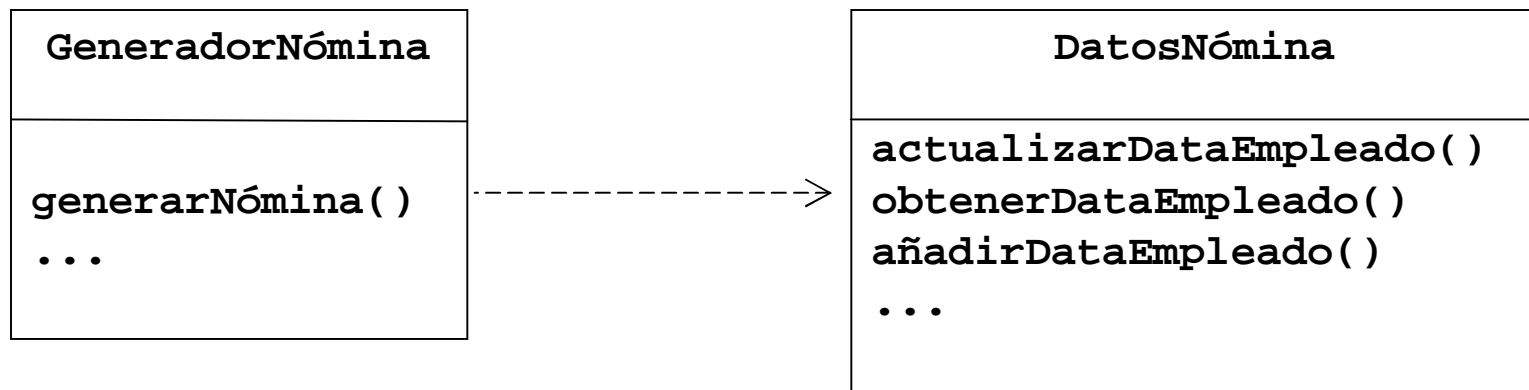
Ámbito y Visibilidad

- Ámbito:
 - Ámbito de clase.
 - Ámbito de instancia.
- Visibilidad:
 - Pública (+)
 - Protegida (#)
 - Privada (-)
 - Paquete (~)

Empleado
<ul style="list-style-type: none">-cuentaEmpleado# empleadoId# empleadoNombre
<ul style="list-style-type: none">+ colocarEmpleadoId()+ colocarEmpleadoNombre()+ obtenerEmpleadoId()+ obtenerEmpleadoNombre()+ obtenerCuentaEmpleado()...

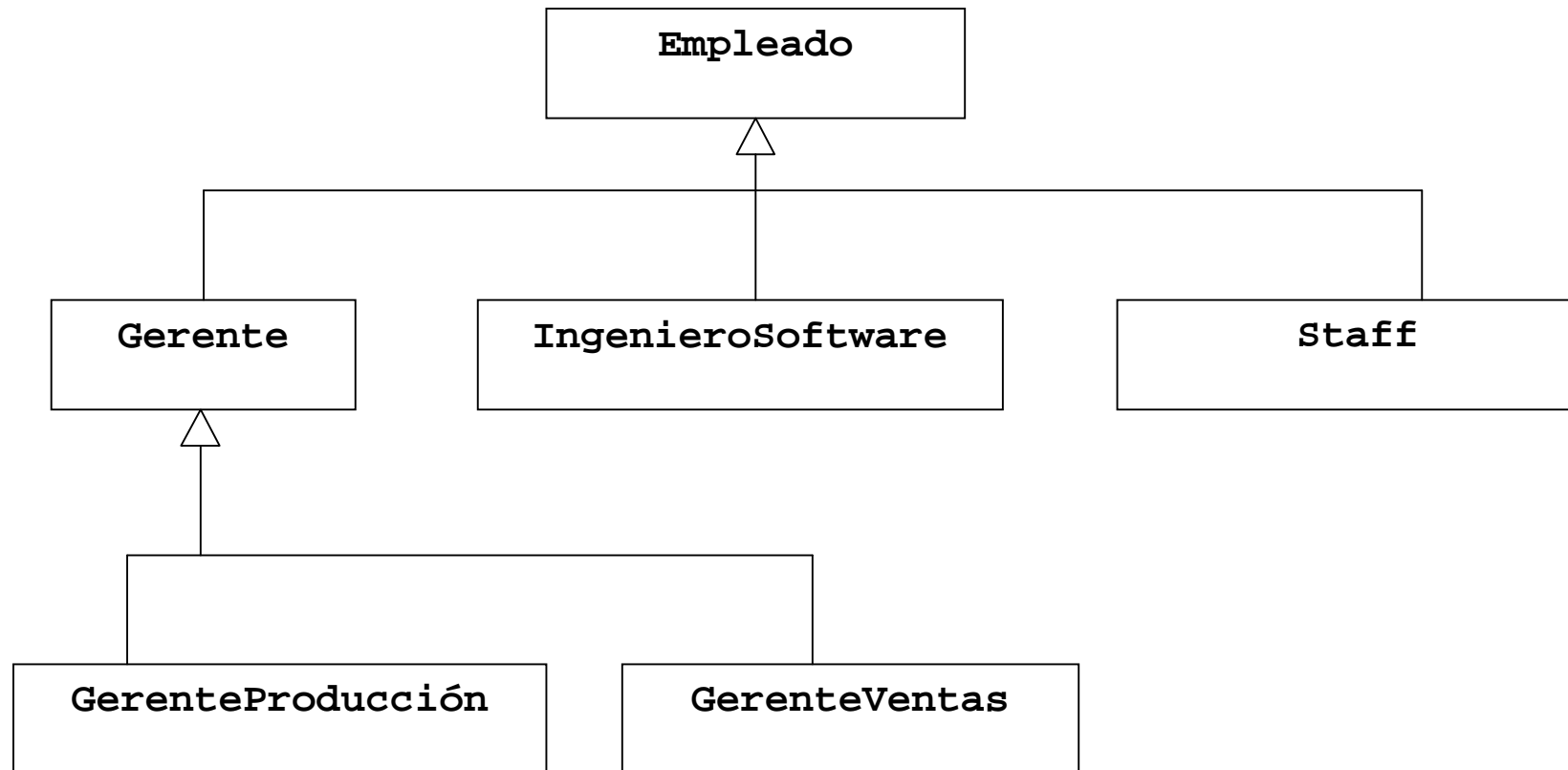
Relación de Dependencia

- La relación de dependencia es la relación “**utiliza**”.
- En UML esta relación se denota con **una línea punteada dirigida** (con una cabeza de flecha abierta), apuntando hacia la clase de la cual se depende.



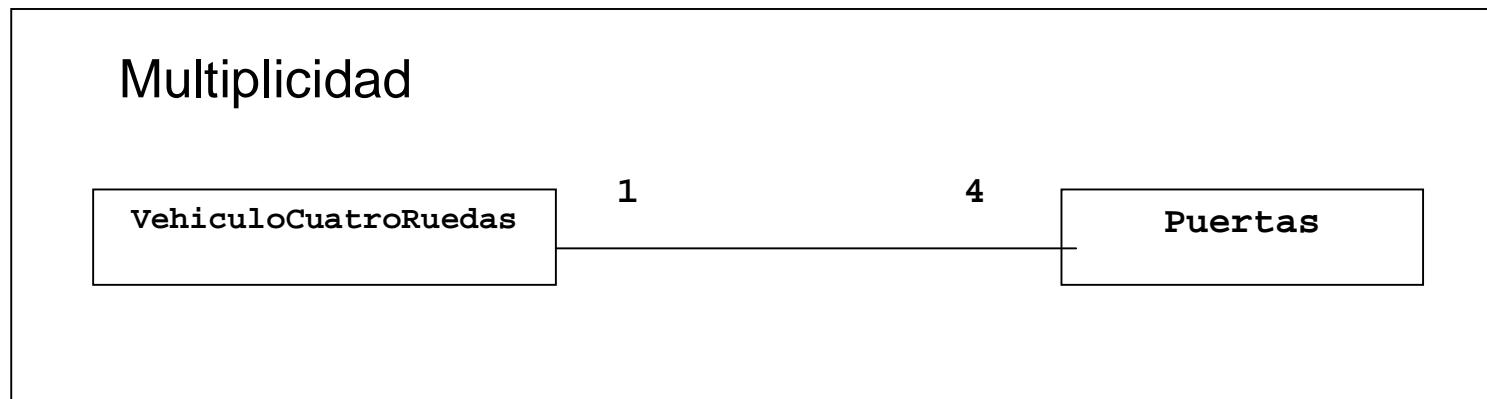
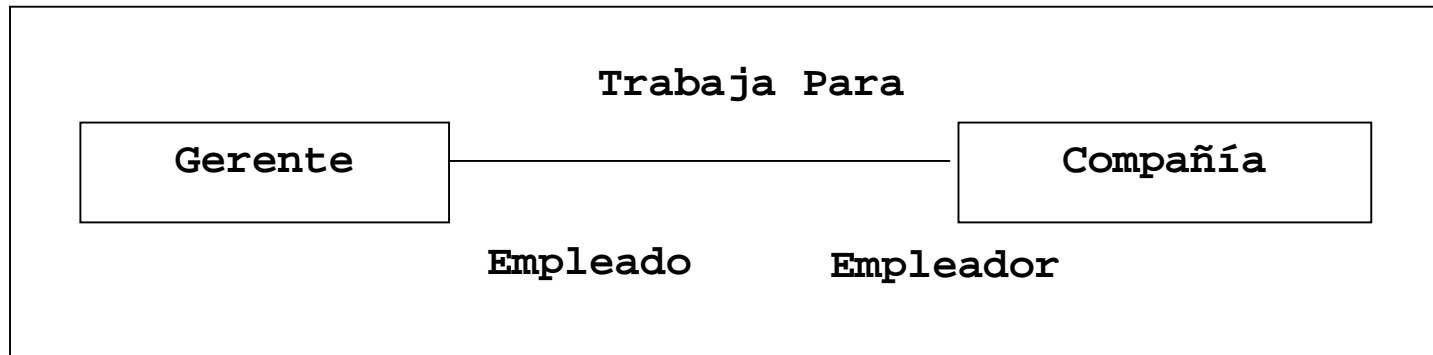
Relación de Generalización

- La superclase representa un concepto general mientras que la subclase representa un concepto más específico.
- Esta relación también es llamada una relación **“es un tipo de”**.



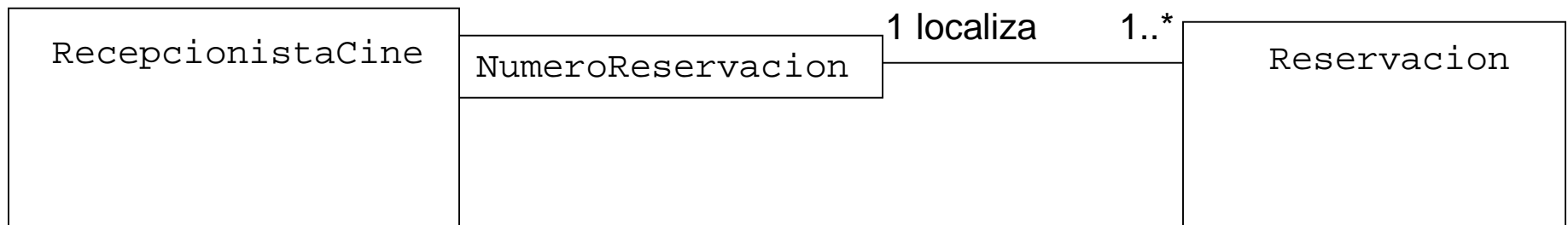
Relación de Asociación

- Una relación de asociación denota la **relación estructural entre elementos**.
- La notación UML para una asociación es **una línea sólida conectando los dos elementos participantes** en la relación de asociación.



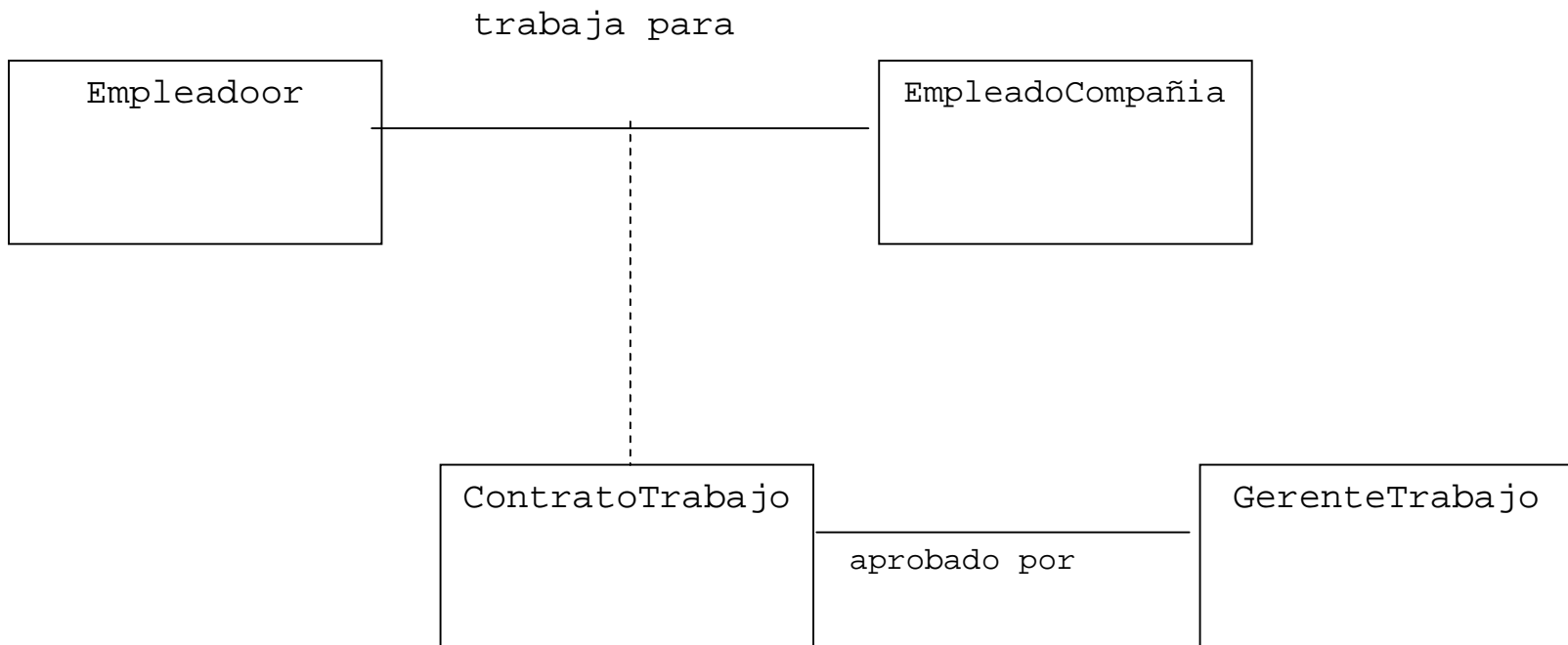
Asociación Calificada

- Una clase debe valerse de un atributo en específico para localizar un objeto adecuado. Este atributo debe ser un identificador único.



Clase Asociación

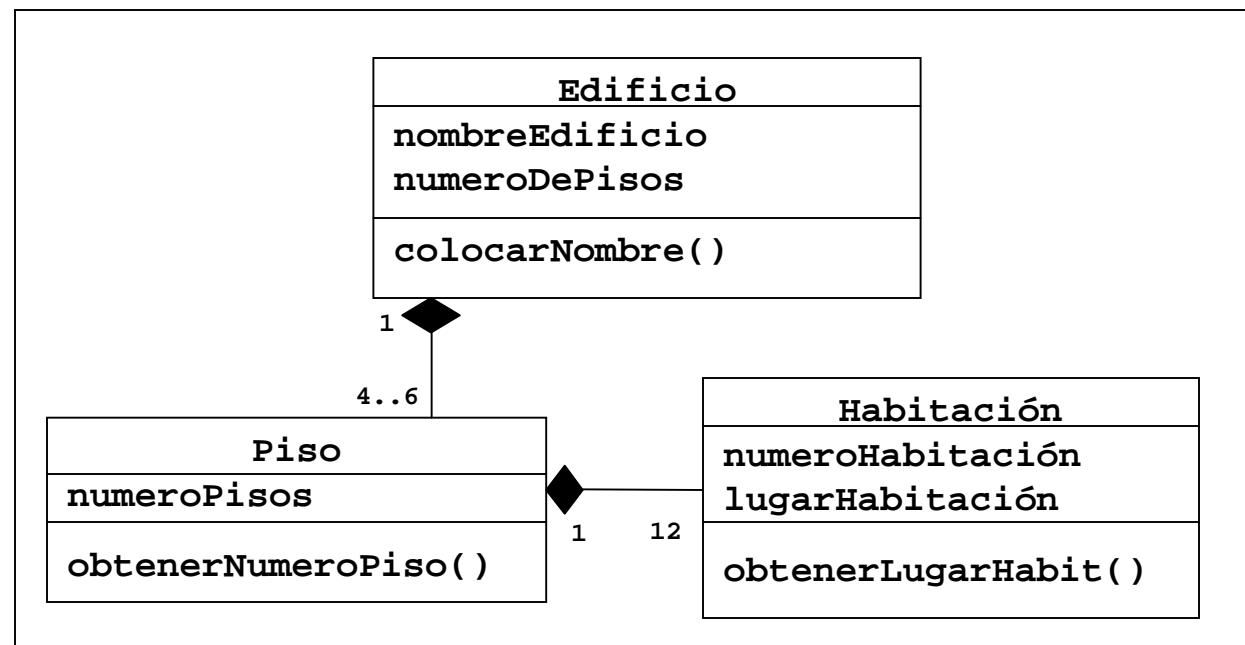
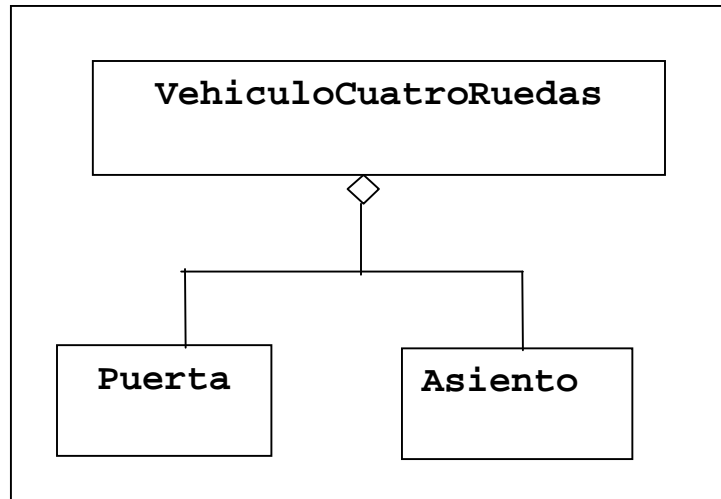
- Modela atributos y operaciones de una asociación.
- La conexión se realiza mediante una línea discontinua entre la asociación de dos clases relacionadas.



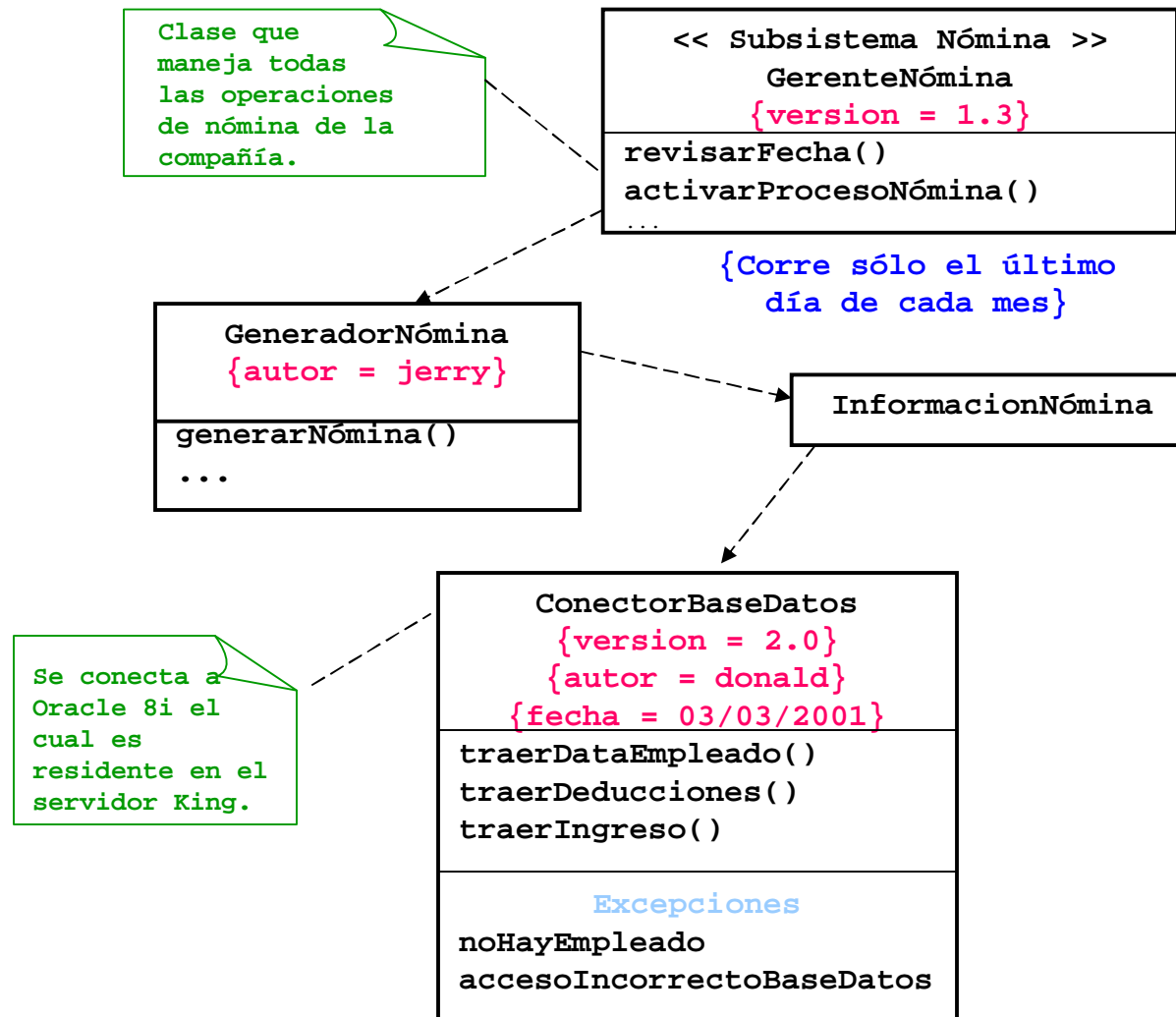
Relación de Agregación

- Extendiendo la relación básica de asociación podemos especificar una relación todo-parte. Esta relación significa una relación de “**es parte de**” o “**tiene un**”.
- Una agregación es **Simple** cuando el tiempo de vida del todo y las partes no están enlazados, mientras que en una **Composición** sí están enlazados, esto es, si el todo deja de existir las partes automáticamente dejan de existir.
- La notación UML utiliza un diamante sin rellenar para agregación Simple y relleno para Composición, ubicado cerca de la clase que forma el todo.

Relación de Agregación...1



Ejemplo de Mecanismos Comunes



Leyendas:

Nota

Valor Etiquetado

Restricción

Compartimiento extra

Trabajando con Diagramas de Clases

- Un diagrama de clases gráficamente describe un conjunto de clases, interfaces, colaboraciones y las relaciones entre ellas.
- Los diagramas de clases pueden contener notas y restricciones.
- Esto es, un diagrama entero de clases puede ser comentado o una restricción puede ser dada para todo el diagrama de clases.
- **Ejemplo Ilustrativo: Tienda de Video** (Leer en la Guía del Estudiante)

Trabajando con Diagramas de Clases...1

Identificación de las Clases:



Trabajando con Diagramas de Clases...2

Identificación de Atributos y Operaciones:

Empleado
+ id_empleado - cargo - fecha_ingreso - status_emp
+ incluirEmpleado() + retirarEmpleado()

Usuario
+ nombre + apellido + edad + telefono + direccion + e_civil + name + cedula

T_credito
- num_tarjeta - fecha_vencimiento - limite_credito

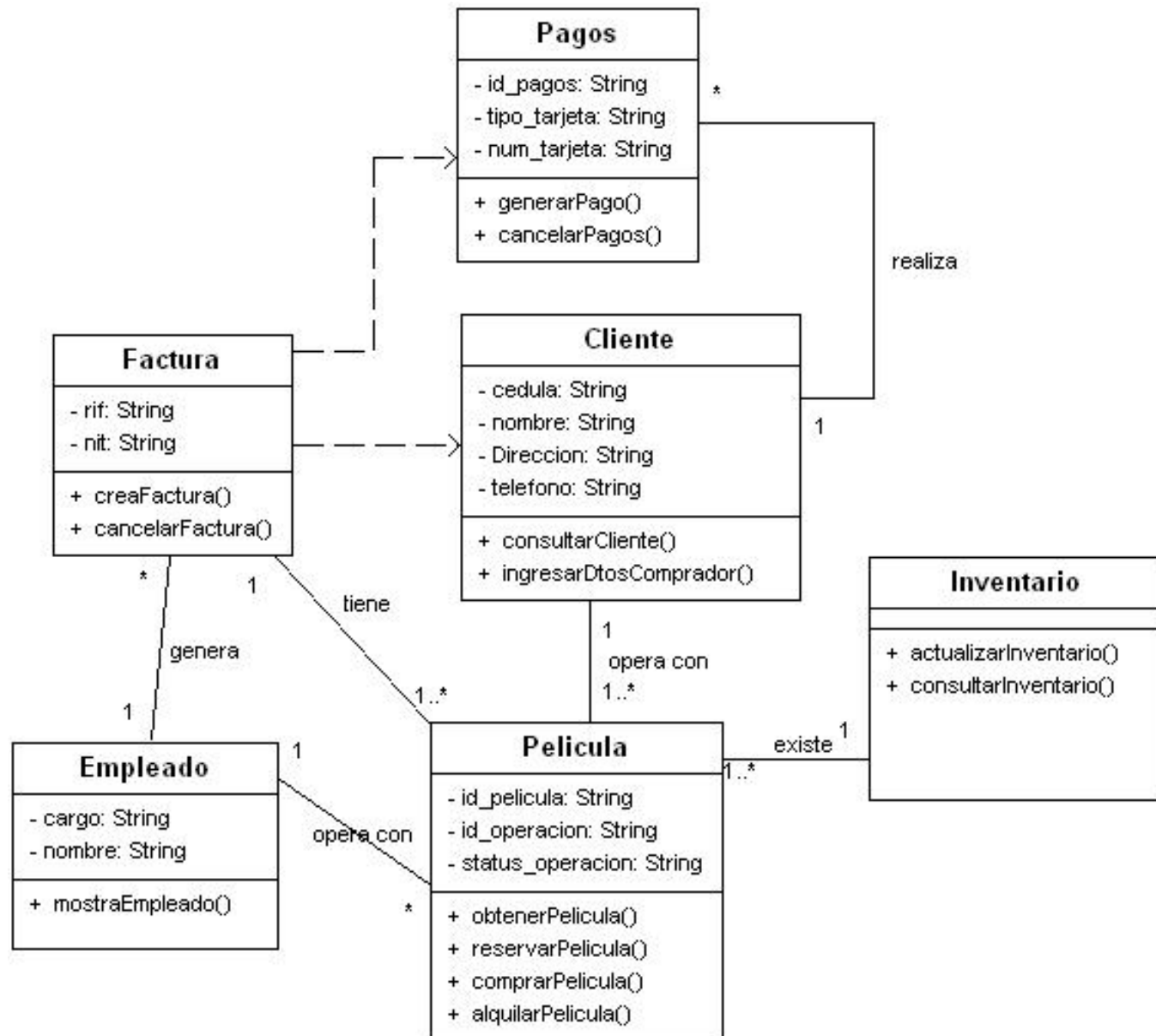
Reservacion
- id_reservacion - fecha_reservacion - status
+ realizarReservacion() + eliminarReservacion() + consultar stock()

Pelicula
- id_pelicula - titulo - genero - monto - deccripcion
+ alquilarPelicula() + comprarPelicula() + devolverPelicula() + consultar_pelicula() + llenarPelicula()

Inventario
- id_pelicula - cantidad
+ actualizarInventarioi() + consultarPelicula() + consultarInventario()

Factura
- fecha - Idfactura - monto - status
+ crearFactura() + imprimirFactura() + almacenarFactura() + anularFactura()

Trabajando con Diagramas de Clases...3



Resumen

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

- Describir el modelado estructural de UML en forma básica.
- Explicar los principios de la orientación a objetos.
- Describir clases y sus relaciones.
- Conocer los mecanismos comunes que pueden ser aplicados a las clases.
- Elaborar diagramas de clases.