



UCA

Programación Orientada a Objetos I

UML – MECANISMOS COMUNES

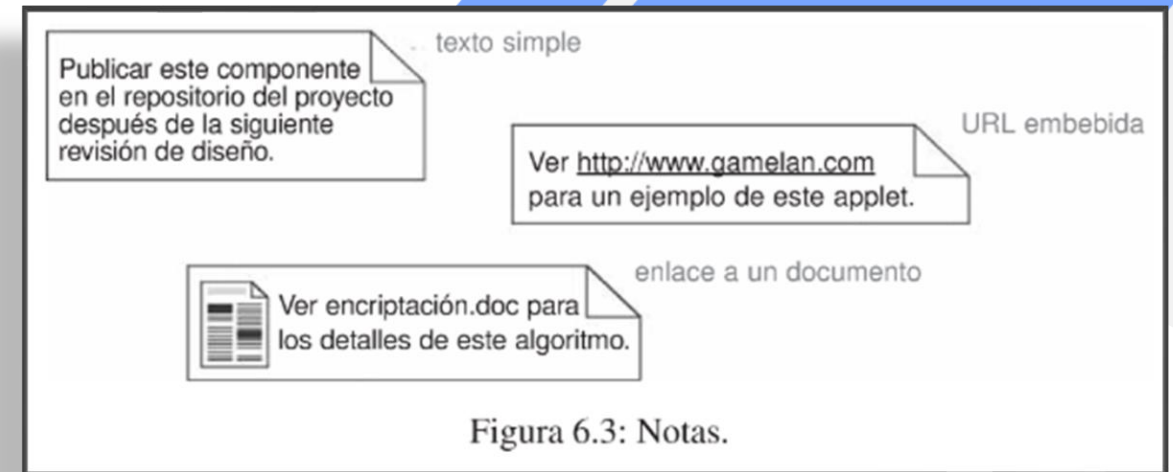
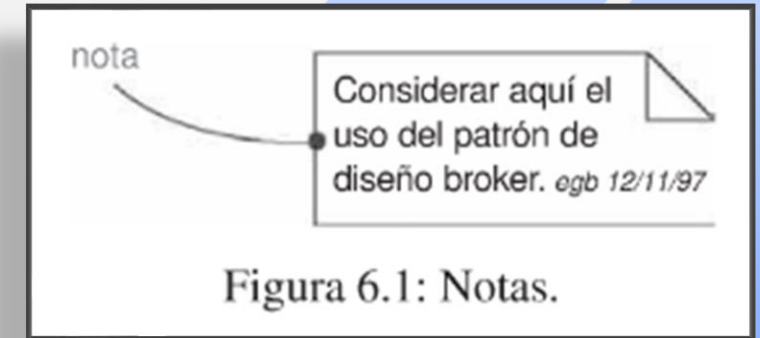
Mecanismos comunes

- A veces es necesario salirse fuera de los límites de una notación:
 - Para comunicar mejor una idea.
 - Para modificar algo puntual
 - Para extender el lenguaje
- Pero esas aclaraciones, extensiones o modificaciones deben ser en forma controlada
- UML nos ofrece varias formas de hacerlo:
 - Notas
 - Otros adornos
 - Estereotipos
 - Valores etiquetados
 - Restricciones



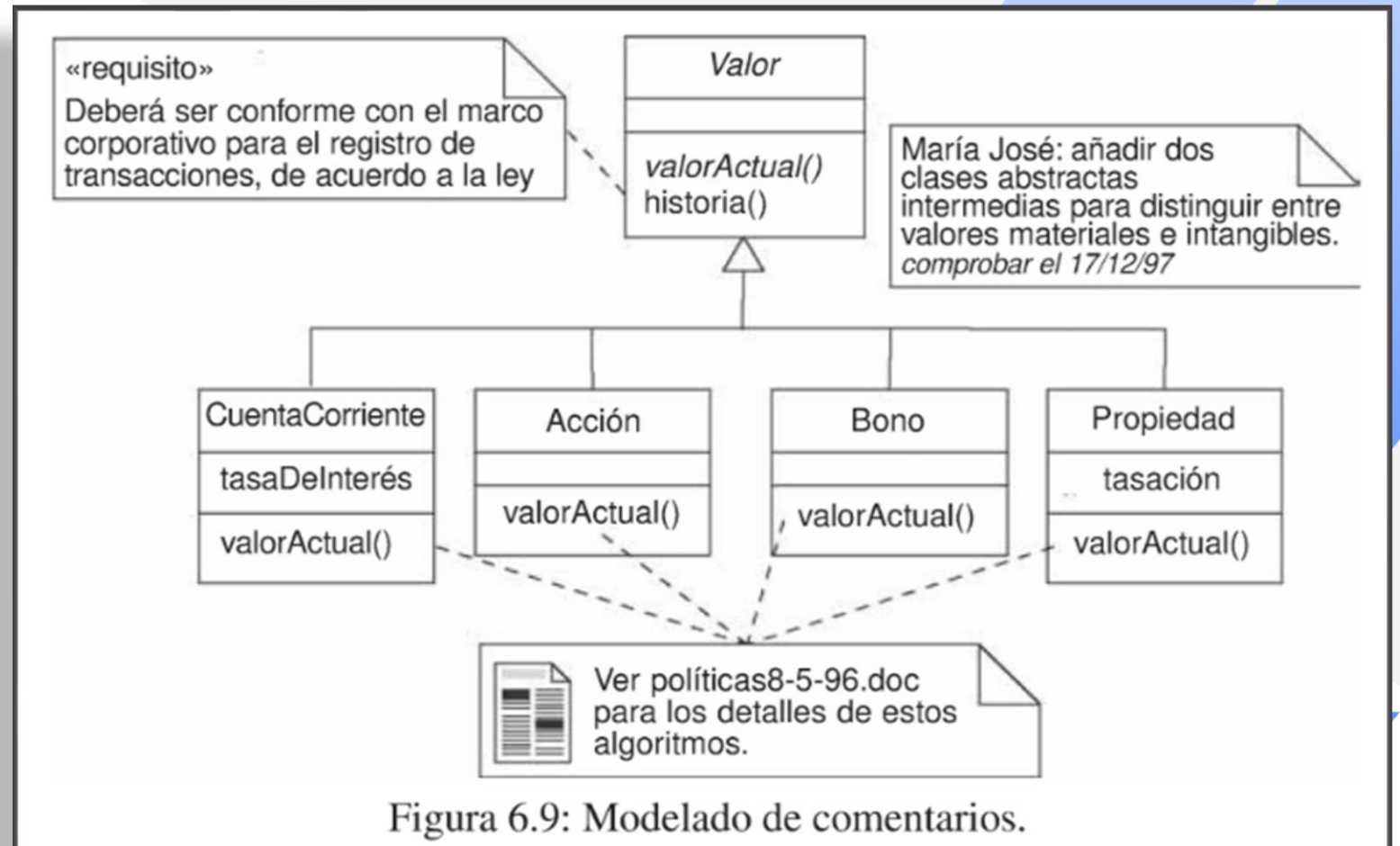
Notas

- Una nota es un elemento gráfico que nos permite especificar cosas como requisitos, observaciones, revisiones, explicaciones, etc.
- Una nota puede contener cualquier combinación de texto o gráficos.
- Incluso puede enlazar con otro documento
- Se representa con un rectángulo con un triángulo en un extremo (simula una hoja de papel con una esquina doblada)
- Normalmente se une al elemento al cual se relaciona mediante una relación de dependencia (línea punteada).



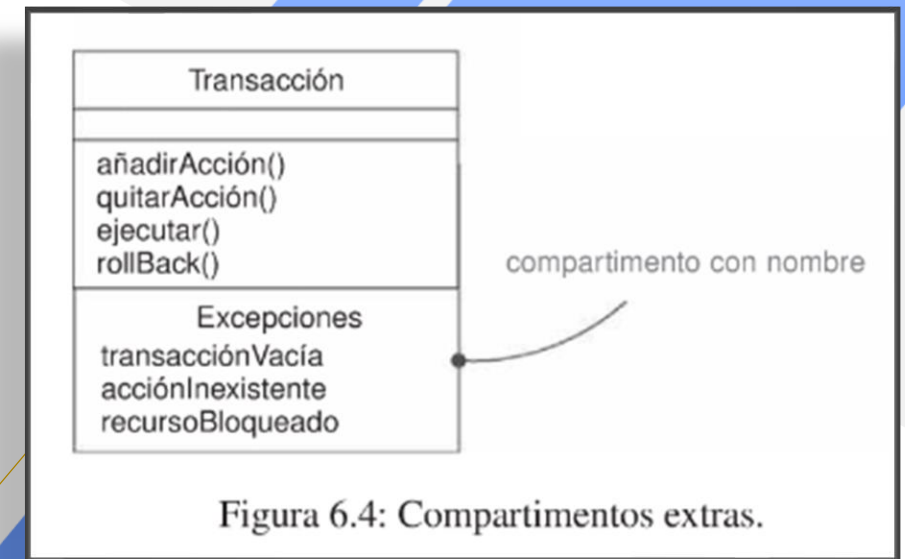
Notas

- Una nota se conecta mediante una dependencia (línea punteada), aunque a veces no es necesario.
- Podemos utilizar las notas conectadas a un elemento cuando tiene que ver con él
- A varios elementos, si hace referencia a más de uno
- O sin conectar cuando no tienen que ver con algún elemento en especial y/o tiene que ver con todo el diagrama o se quiere aclarar algo.
- Como siempre, no debe entorpecer la lectura y debe aportar claridad



Otros adornos

- Un adorno es un mecanismo que permite mostrar detalles de especificación
- Vimos que hay adornos que podemos colocar en las relaciones que nos dan detalles sobre la relación, pero no es el único lugar donde podemos utilizarlos.
- La mayoría de los adornos se muestran colocando texto junto al elemento o añadiendo algún símbolo gráfico, pero a veces hace falta aportar más que eso.
- Por ejemplo, durante la ejecución de una operación de una clase (método) sabemos que pueden producirse errores (excepciones), y también sabemos que debemos tratarlas. Podemos "ampliar" la especificación de la clase con un nuevo compartimento que las muestre.

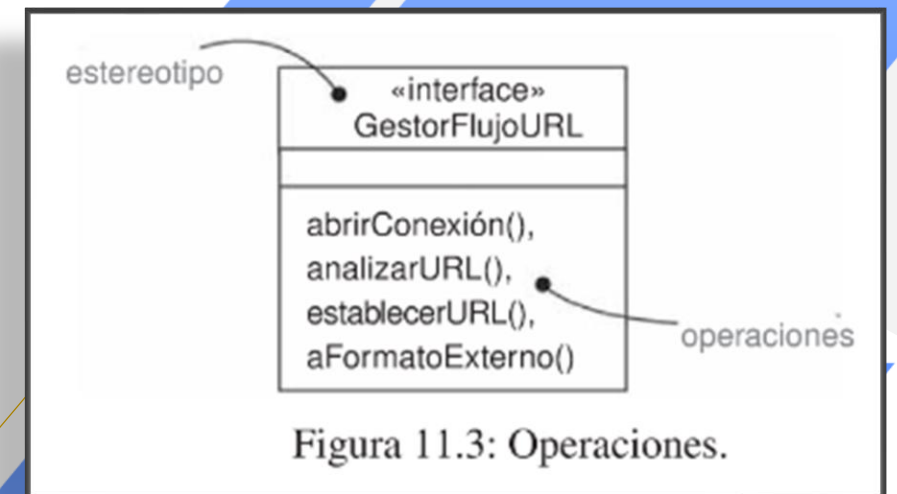


Estereotipos

- A veces se desea introducir nuevos elementos relacionados con el vocabulario del dominio y que parezcan bloques de construcción primitivos
- Un estereotipo no es lo mismo que una clase padre en una relación de generalización.
- Es como un "metatipo", un tipo que define otros tipos
- Al aplicar un estereotipo sobre una clase, se está creando un nuevo bloque de construcción con sus propias características, semántica (reglas y restricciones) y notación (texto o gráfico)
- Un estereotipo se representa con un nombre entre comillas francesas (<<nombre>>)
- Se coloca sobre el nombre del elemento y puede tener un ícono
- También puede utilizarse algún color para agregar una sutil señal visual (UML no prohíbe los colores, pero tampoco los aconseja)

Estereotipos

- Estereotipos más utilizados:
 - «interface»: indica que la clase es una interfaz
 - «abstract»: indica que la clase es abstracta
 - «enumeration»: indica que la clase es una enumeración, que define un conjunto de valores discretos
 - «entity»: indica que la clase representa una entidad en un modelo de datos, como puede ser una tabla en una base de datos relacional
 - «control»: indica que la clase es un controlador que coordina acciones entre diferentes partes del sistema
 - «service»: indica que la clase es un servicio que proporciona alguna funcionalidad (utilizada en SOA, arquitectura orientada a servicios)



Valores etiquetados

- Todo elemento tiene su propio conjunto de propiedades:
 - Clases: nombre, atributos, operaciones, responsabilidad
 - Relaciones: nombre, roles, cardinalidad, etc.
- Los valores etiquetados (tag) agregan nuevas propiedades a los elementos
- Normalmente tienen un nombre y un valor
- Para las clases por ejemplo, no son nuevos atributos dentro una clase, ***son nuevos atributos de la especificación de la clase***
- Podría ser una nueva sección para describir algo importante
- En este ejemplo, el nombre es "Notas" y el valor "Esta clase representa una sección"

ClaseEjemplo
- atributo1: int - atributo2: String - atributo3: boolean
+ metodo1(): void + metodo2(int): int + metodo3(int): String

ClaseEjemplo
- atributo1: int - atributo2: String - atributo3: boolean
+ metodo1(): void + metodo2(int): int + metodo3(int): String
<div>Notas</div> <div>Esta clase representa un ejemplo de un valor etiquetado llamado Notas que agrega a la especificación de la clase una sección</div>



Restricciones

- Las restricciones agregan una nueva semántica o modifican una regla existente. Recordar que hablar de semántica para una clase es hablar de su significado o interpretación dentro del contexto de lo que se está modelando, es hablar de su propósito y de como se relaciona con otras clases
- Una restricción especifica condiciones que deben cumplirse.
- Se colocan con el texto descriptivo entre llaves
- Pueden:
 - afectar a un solo elemento
 - afectan a varios elementos
- Aplicadas a las relaciones, se puede modelar que una clase pueda tener solo un tipo de relación con otra clase a la vez

