

## Construcción de una casa para "fido"



Puede hacerlo una sola persona  
 Requiere:  
 Modelado mínimo  
 Proceso simple  
 Herramientas simples

www.dsicupv.es/~uml

1

## Construcción de una casa

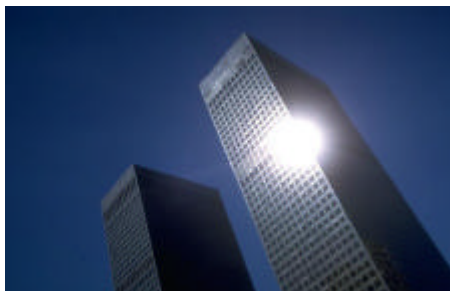


Construida eficientemente y en un tiempo razonable por un equipo  
 Requiere:  
 Modelado  
 Proceso bien definido  
 Herramientas más sofisticadas

www.dsicupv.es/~uml

2

## Construcción de un rascacielos



www.dsicupv.es/~uml

3

## Claves en Desarrollo de SI

Notación



Herramientas

Proceso

www.dsicupv.es/~uml

4

## Abstracción - Modelado Visual (MV)

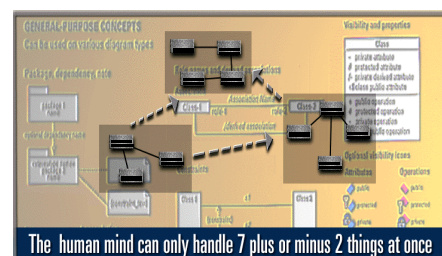
"El modelado captura las partes esenciales del sistema"



www.dsicupv.es/~uml

5

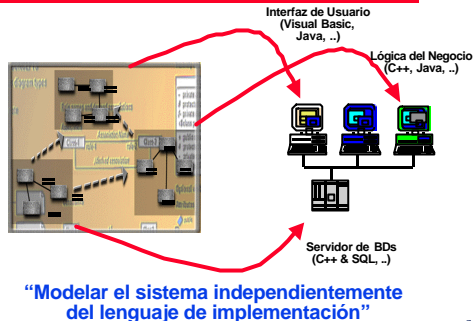
## MV para manejar la complejidad



www.dsicupv.es/~uml

6

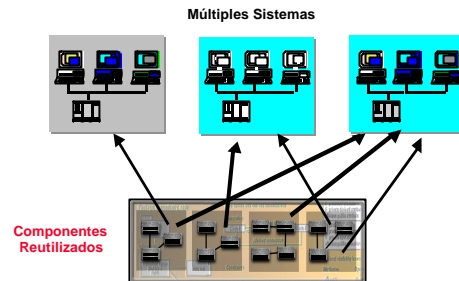
## MV para definir la Arquitectura del Software



www.dsic.upv.es/~uml

7

## MV promueve la reutilización



www.dsic.upv.es/~uml

8

## ¿Qué es UML?

- UML = Unified Modeling Language
- Un lenguaje de propósito general para el modelado orientado a objetos
- Documento "OMG Unified Modeling Language Specification"
- UML combina notaciones provenientes desde:
  - Modelado Orientado a Objetos
  - Modelado de Datos
  - Modelado de Componentes
  - Modelado de Flujos de Trabajo (Workflows)

www.dsic.upv.es/~uml

9

## Situación de Partida

- Diversos métodos y técnicas OO, con muchos aspectos en común pero utilizando distintas notaciones
- Inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc.
- Pugna entre distintos enfoques (y correspondientes gurús)
- => Necesidad de una notación estándar

www.dsic.upv.es/~uml

10

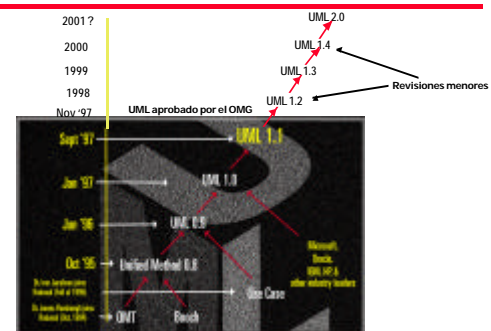
## Historia de UML

- Comenzó como el "Método Unificado", con la participación de Grady Booch y Jim Rumbaugh. Se presentó en el OOPSLA'95
- El mismo año se unió Ivar Jacobson. Los "Tres Amigos" son socios en la compañía Rational Software. Herramienta CASE Rational Rose

www.dsic.upv.es/~uml

11

## Historia de UML



www.dsic.upv.es/~uml

12

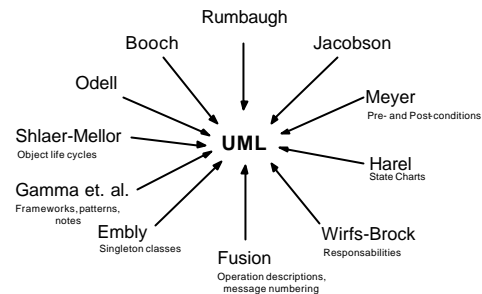
## Participantes en UML 1.0

- Rational Software
- (Grady Booch, Jim Rumbaugh y Ivar Jacobson)
- Digital Equipment
- Hewlett-Packard
- i-Logix (David Harel)
- IBM
- ICON Computing (Desmond D'Souza)
- Intellicorp and James Martin & co. (James Odell)
- MCI Systemhouse
- Microsoft
- ObjecTime
- Oracle Corp.
- Platinum Technology
- Sterling Software
- Taskon
- Texas Instruments
- Unisys

www.dsic.upv.es/~uml

13

## UML "aglutina" enfoques OO



www.dsic.upv.es/~uml

14

## Métodos Formales en Modelado

- Tipos de enfoques: no-formales, semi-formales y formales
- Las principales mejoras al utilizar métodos formales son:
  - ✦ Mayor rigor en la especificación
  - ✦ Mejores condiciones para realizar la verificación y validación en forma más exhaustiva
  - ✦ Mejores condiciones para automatización de procesos para la generación automática de prototipos y/o código final

www.dsic.upv.es/~uml

15

## Inconvenientes en UML

- Definición del proceso de desarrollo usando UML. **UML no es una metodología**
- Falta integración con respecto de otras técnicas tales como patrones de diseño, interfaces de usuario, documentación, etc.
- Ejemplos aislados
- "Monopolio de conceptos, técnicas y métodos entorno a UML"

www.dsic.upv.es/~uml

16

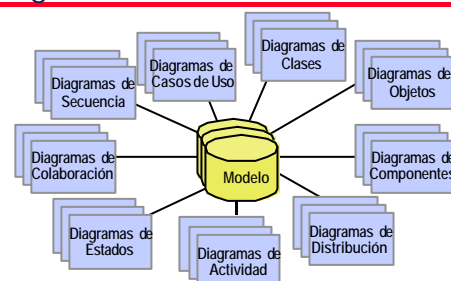
## Perspectivas de UML

- UML será el lenguaje de modelado orientado a objetos estándar predominante los próximos años
- Razones:
  - ✦ Participación de metodólogos influyentes
  - ✦ Participación de importantes empresas
  - ✦ Aceptación del OMG como notación estándar
- Evidencias:
  - ✦ Herramientas que proveen la notación UML
  - ✦ "Edición" de libros
  - ✦ Congresos, cursos, "camisetas", etc.

www.dsic.upv.es/~uml

17

## Diagramas de UML



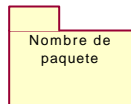
"Un modelo es una descripción completa de un sistema desde una perspectiva concreta"

www.dsic.upv.es/~uml

18

## Paquetes en UML

- Los paquetes ofrecen un mecanismo general para la organización de los modelos agrupando elementos de modelado
- Se representan gráficamente como:



www.dsic.upv.es/~uml

19

## ... Paquetes en UML

- Cada paquete corresponde a un subconjunto del modelo y contiene, según el modelo, clases, objetos, relaciones, componentes y diagramas asociados
- Un paquete puede contener otros paquetes, sin límite de anidamiento pero cada elemento pertenece a (está definido en) sólo un paquete

www.dsic.upv.es/~uml

20

## ... Paquetes en UML

- Una clase de un paquete puede aparecer en otro paquete por la importación a través de una relación de dependencia entre paquetes
- Todas las clases no son necesariamente visibles desde el exterior del paquete, es decir, un paquete encapsula a la vez que agrupa

www.dsic.upv.es/~uml

21

## El Paradigma Orientado a Objetos

www.dsic.upv.es/~uml

22

## ¿Por qué la Orientación a Objetos?

- Proximidad de los conceptos de modelado respecto de las entidades del mundo real
  - ✦ Mejora captura y validación de requisitos
  - ✦ Acerca el "espacio del problema" y el "espacio de la solución"
- Modelado integrado de propiedades estáticas y dinámicas del ámbito del problema
  - ✦ Facilita construcción, mantenimiento y reutilización

www.dsic.upv.es/~uml

23

## ¿Por qué la Orientación a Objetos?

- Conceptos comunes de modelado durante el análisis, diseño e implementación
  - ✦ Facilita la transición entre distintas fases
  - ✦ Favorece el desarrollo iterativo del sistema
  - ✦ Disipa la barrera entre el "qué" y el "cómo"
- Sin embargo, existen problemas ...

www.dsic.upv.es/~uml

24

## Problemas en OO

"...Los conceptos básicos de la OO se conocen desde hace dos décadas, pero su aceptación todavía no está tan extendida como los beneficios que esta tecnología puede sugerir"

"...La mayoría de los usuarios de la OO no utilizan los conceptos de la OO de forma purista, como inicialmente se pretendía. Esta práctica ha sido promovida por muchas herramientas y lenguajes que intentan utilizar los conceptos en diversos grados"

-- Wolfgang Strigel

www.dsic.upv.es/~uml

25

## Objetos

- Objeto = unidad atómica que integra estado y comportamiento
- La encapsulación en un objeto permite una alta **cohesión** y un bajo **acoplamiento**
- Un objeto puede caracterizar una entidad física (coche) o concepto (ecuación matemática)

www.dsic.upv.es/~uml

26

## ... Objetos

- El Modelado de Objetos permite representar el ciclo de vida de los objetos a través de sus interacciones
- En UML, un objeto se representa por un rectángulo con un nombre subrayado

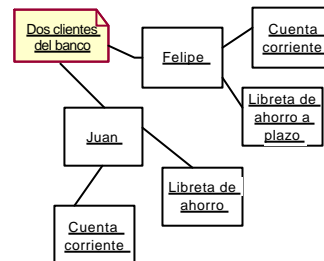


www.dsic.upv.es/~uml

27

## ... Objetos

- Ejemplo de varios objetos relacionados:

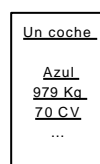


www.dsic.upv.es/~uml

28

## ... Objetos

- Objeto = Identidad + Estado + Comportamiento
- El estado está representado por los valores de los atributos
- Un atributo toma un valor en un dominio concreto



www.dsic.upv.es/~uml

29

## Identidad

- ✳ **Oid (Object Identifier)**
- ✳ Cada objeto posee un oid. El oid establece la identidad del objeto y tiene las siguientes características:
  - ✳ Constituye un identificador único y global para cada objeto dentro del sistema
  - ✳ Es determinado en el momento de la creación del objeto
  - ✳ Es independiente de la localización física del objeto, es decir, provee completa independencia de localización

www.dsic.upv.es/~uml

30

## ... Identidad

- ✦ Es independiente de las propiedades del objeto, lo cual implica independencia de valor y de estructura
- ✦ No cambia durante toda la vida del objeto. Además, un oid no se reutiliza aunque el objeto deje de existir
- ✦ No se tiene ningún control sobre los oids y su manipulación resulta transparente
- ✦ Sin embargo, es preciso contar con algún medio para hacer referencia a un objeto utilizando referencias del dominio (valores de atributos)

www.dsic.upv.es/~uml

31

## Estado

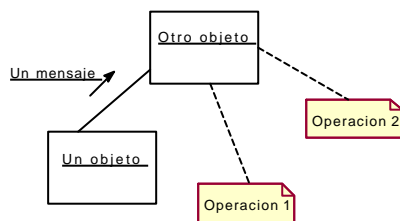
- ✦ El estado evoluciona con el tiempo
- ✦ Algunos atributos pueden ser constantes
- ✦ El comportamiento agrupa las competencias de un objeto y describe las acciones y reacciones de ese objeto
- ✦ Las operaciones de un objeto son consecuencia de un estímulo externo representado como mensaje enviado desde otro objeto

www.dsic.upv.es/~uml

32

## Comportamiento

- Ejemplo de interacción:



www.dsic.upv.es/~uml

33

## ... Comportamiento

- Los mensajes navegan por los enlaces, a priori en ambas direcciones
- Estado y comportamiento están relacionados
- Ejemplo: no es posible aterrizar un avión si no está volando. Está volando como consecuencia de haber despegado del suelo

www.dsic.upv.es/~uml

34

## Persistencia

- ✦ La persistencia de los objetos designa la capacidad de un objeto trascender en el espacio/tiempo
- ✦ Un objeto persistente conserva su estado en un sistema de almacenamiento permanente (usualmente memoria secundaria)
- ✦ Podremos después reconstruirlo, es decir, cogerlo de memoria secundaria para utilizarlo en la ejecución (materialización del objeto)
- ✦ Los lenguajes OO no proponen soporte adecuado para la persistencia, pues ésta debería ser transparente, un objeto existe desde su creación

www.dsic.upv.es/~uml

35

## Comunicación

- Un sistema informático puede verse como un conjunto de objetos autónomos y concurrentes que trabajan de manera coordinada en la consecución de un fin específico
- El comportamiento global se basa pues en la comunicación entre los objetos que la componen

www.dsic.upv.es/~uml

36

## ... Comunicación

- Categorías de objetos :
  - Activos – Pasivos
  - Clientes – Servidores
- Objeto Activo: posee un hilo de ejecución (thread) propio y puede iniciar una actividad
- Objeto Pasivo: no puede iniciar una actividad pero puede enviar estímulos una vez que se le solicita un servicio
- Cliente es el objeto que solicita un servicio. Servidor es el objeto que provee el servicio solicitado

www.dsic.upv.es/~uml

37

## El Concepto de Mensaje

- La unidad de comunicación entre objetos se llama mensaje.
- El mensaje es el soporte de una comunicación que vincula dinámicamente los objetos que fueron separados previamente en el proceso de descomposición.
- Una operación es la especificación y la implementación de una función efectuada por un objeto.

www.dsic.upv.es/~uml

38

## El Concepto de Operación

- Las operaciones manipulan los atributos del objeto.
- Pueden tener parámetros de entrada y/o salida.
- Un mensaje de un objeto a otro involucra la ejecución de una operación.

www.dsic.upv.es/~uml

39

## El Concepto de Operación

- El nombre del mensaje es el de la operación.
- Los parámetros del mensaje son los parámetros de la operación.

www.dsic.upv.es/~uml

40

## Casos de Uso

www.dsic.upv.es/~uml

41

## Casos de Uso

- Los Casos de Uso (Ivar Jacobson) describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el p.d.v. del usuario
- Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno
- Los Casos de Uso son descripciones de la funcionalidad del sistema independientes de la implementación
- Comparación con respecto a los Diagramas de Flujo de Datos del Enfoque Estructurado

www.dsic.upv.es/~uml

42

### ... Casos de Uso

- Los Casos de Uso particionan el conjunto de necesidades atendiendo a la categoría de usuarios que participan en el mismo
- Están basado en el lenguaje natural, es decir, es accesible por los usuarios

www.dsic.upv.es/~uml

43

### Ejemplo de Casos de Uso

**Caso de Uso:** Comprar productos

**Actores:** Cliente, Cajero

**Tipo:** Primario



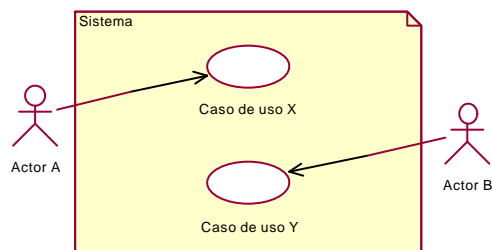
**Descripción:** Un cliente llega a la caja registradora con los artículos que comprará. El cajero registra los artículos y cobra el importe. Al terminar la operación el cliente se marcha con los productos.

www.dsic.upv.es/~uml

44

### ... Casos de Uso

■ Ejemplo:



www.dsic.upv.es/~uml

45

### ... Casos de Uso

- Actores:
  - ✦ Principales: personas que usan el sistema
  - ✦ Secundarios: personas que mantienen o administran el sistema
  - ✦ Material externo: dispositivos materiales imprescindibles que forman parte del ámbito de la aplicación y deben ser utilizados
  - ✦ Otros sistemas: sistemas con los que el sistema interactúa
- La misma persona física puede interpretar varios papeles como actores distintos
- El nombre del actor describe el papel desempeñado

www.dsic.upv.es/~uml

46

### ... Casos de Uso

- Los Casos de Uso se determinan observando y precisando, actor por actor, las secuencias de interacción, los escenarios, desde el punto de vista del usuario
- Un escenario es una instancia de un caso de uso
- Los casos de uso intervienen durante todo el ciclo de vida. El proceso de desarrollo estará dirigido por los casos de uso

www.dsic.upv.es/~uml

47

### Casos de Uso: Relaciones

- UML define cuatro tipos de relación en los Diagramas de Casos de Uso:

✦ **Comunicación:**



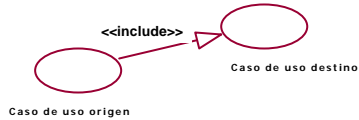
www.dsic.upv.es/~uml

48



### ... Casos de Uso: Relaciones

✳ **Inclusión** : una instancia del Caso de Uso origen incluye también el comportamiento descrito por el Caso de Uso destino



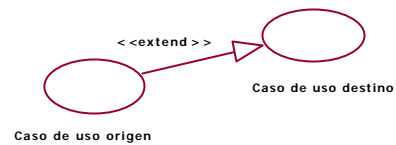
En UML se estereotipa como <<include>>

www.dsic.upv.es/~uml

49

### ... Casos de Uso: Relaciones

✳ **Extensión** : el Caso de Uso origen extiende el comportamiento del Caso de Uso destino

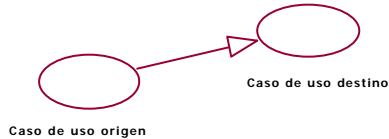


www.dsic.upv.es/~uml

50

### ... Casos de Uso: Relaciones

✳ **Herencia** : el Caso de Uso origen hereda la especificación del Caso de Uso destino y posiblemente la modifica y/o amplía

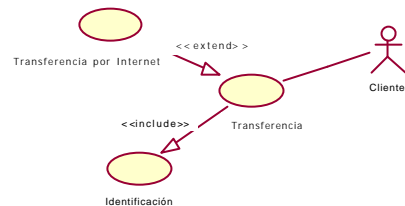


www.dsic.upv.es/~uml

51

### ... Casos de Uso: Relaciones

■ Ejemplo:



www.dsic.upv.es/~uml

52

### Casos de Uso: Construcción

- Un caso de uso debe ser simple, inteligible, claro y conciso
- Generalmente hay pocos actores asociados a cada Caso de Uso
- Preguntas clave:
  - ✳ ¿cuáles son las tareas del actor?
  - ✳ ¿qué información crea, guarda, modifica, destruye o lee el actor?
  - ✳ ¿debe el actor notificar al sistema los cambios externos?
  - ✳ ¿debe el sistema informar al actor de los cambios internos?

www.dsic.upv.es/~uml

53

### ... Casos de Uso: Construcción

- La descripción del Caso de Uso comprende:
  - ✳ el inicio: cuándo y qué actor lo produce?
  - ✳ el fin: cuándo se produce y qué valor devuelve?
  - ✳ la interacción actor-caso de uso: qué mensajes intercambian ambos?
  - ✳ objetivo del caso de uso: ¿qué lleva a cabo o intenta?
  - ✳ cronología y origen de las interacciones
  - ✳ repeticiones de comportamiento: ¿qué operaciones son iteradas?
  - ✳ situaciones opcionales: ¿qué ejecuciones alternativas se presentan en el caso de uso?

www.dsic.upv.es/~uml

54

<b>RF-id del requisito</b>	<b>&lt;nombre del requisito funcional&gt;</b>
<b>Versión</b>	<numero de versión y fecha>
<b>Autores</b>	<autor>
<b>Fuentes</b>	<fuente de la versión actual>
<b>Objetivos asociados</b>	<nombre del objetivo>
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso (concreto cuando «evento de activación», abstracto durante la realización de los casos de uso «lista de casos de uso»)
<b>Precondición</b>	<precondición del caso de uso>
<b>Secuencia</b>	<b>Paso Acción</b>
<b>Normal</b>	1 (El «actor», el sistema) «acción realizada por el actor o sistema», se realiza el caso de uso «caso de uso RF-x»
	2 Si «condición», (el «actor», el sistema) «acción realizada por el actor o sistema», se realiza el caso de uso «caso de uso RF-x»
	3
	4
	5
	6
	n
<b>Postcondición</b>	<postcondición del caso de uso>
<b>Excepciones</b>	<b>Paso Acción</b>
	1 Si «condición de excepción», (el «actor», el sistema) «acción realizada por el actor o sistema», se realiza el caso de uso «caso de uso RF-x», a continuación este caso de uso (continúa, aborta)
	2
	3
<b>Rendimiento</b>	<b>Paso Cota de tiempo</b>
	1 n segundos
	2 n segundos
<b>Frecuencia esperada</b>	<nº de veces> veces / <unidad de tiempo>
<b>Importancia</b>	(no importante, importante, vital)
<b>Urgencia</b>	(puede esperar, hay presión, inmediatamente)
<b>Comentarios</b>	<comentarios adicionales>

## Modelado de Interacciones

www.dsic.upv.es/~uml

56

## Interacción

- Los objetos **interactúan** para realizar colectivamente los servicios ofrecidos por las aplicaciones. Los diagramas de interacción muestran cómo se comunican los objetos en una interacción
- Existen dos tipos de diagramas de interacción: los **Diagramas de Colaboración** y los **Diagramas de Secuencia**.

www.dsic.upv.es/~uml

57

## Diagramas de interacción

- Los Diagramas de Secuencia son más adecuados para observar la perspectiva cronológica de las interacciones
- Los Diagramas de Colaboración ofrecen una mejor visión espacial mostrando los enlaces de comunicación entre objetos
- Normalmente el D. de Colaboración se obtiene a partir del correspondiente D. de Secuencia

www.dsic.upv.es/~uml

58

## Diagramas de Secuencia

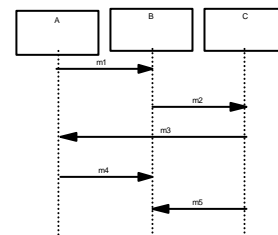
- Muestra la secuencia de mensajes entre objetos durante un escenario concreto.
- Cada objeto viene dado por una barra vertical.
- El tiempo transcurre de arriba abajo.
- Cuando existe demora entre el envío y la atención se puede indicar usando una línea oblicua.

www.dsic.upv.es/~uml

59

## ... Diagramas de Secuencia

- Un ejemplo:



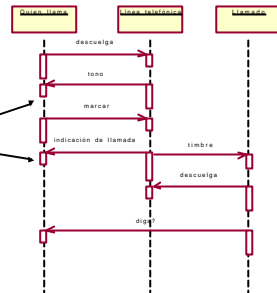
www.dsic.upv.es/~uml

60

## ... Diagramas de Secuencia

### Ejemplo

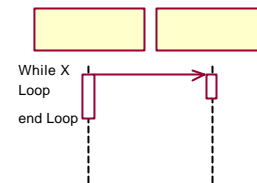
Las bandas rectangulares representan los periodos de actividad de los objetos



61

## ... Estructuras de control

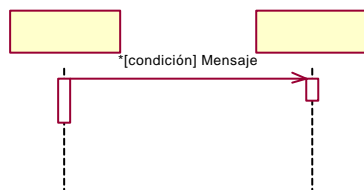
Podemos representar iteraciones en el envío de mensajes, p.e., mientras se cumpla una condición:



62

## ... Estructuras de control

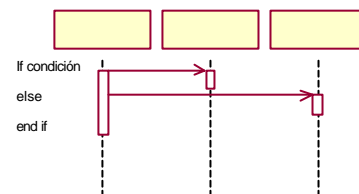
La iteración puede expresarse también como parte del mensaje:



63

## ... Estructuras de control

Las bifurcaciones condicionales pueden representarse de esta forma:



64

## Diagramas de Colaboración

- Son útiles en la fase exploratoria para identificar objetos
- La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto de los demás
- La estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes por los enlaces

65

## Mensajes

- Un mensaje desencadena una acción en el objeto destinatario
- Un mensaje se envía si han sido enviados los mensajes de una lista (sincronización):



66

### ... Mensajes

- Un mensaje se envía iterada y secuencialmente a un conjunto de instancias:

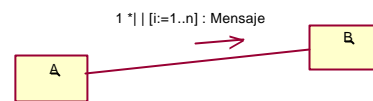


www.dsic.upv.es/~uml

67

### ... Mensajes

- Un mensaje se envía iterada y concurrentemente a un conjunto de instancias:

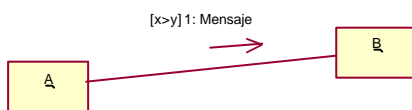


www.dsic.upv.es/~uml

68

### ... Mensajes

- Un mensaje se envía de manera condicionada:

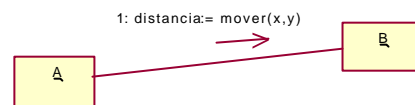


www.dsic.upv.es/~uml

69

### ... Mensajes

- Un mensaje que devuelve un resultado:



www.dsic.upv.es/~uml

70

### ... Mensajes

- Los argumentos de un mensaje pueden ser valores obtenidos como consecuencia de las llamadas anteriores
- Los argumentos pueden ser también expresiones de navegación construidas a partir del objeto cliente
- Los argumentos pueden omitirse en el diagrama

www.dsic.upv.es/~uml

71

## Modelado Conceptual

www.dsic.upv.es/~uml

72

## Clases

- Modelado Conceptual:
  - Organización del conocimiento del dominio del problema en un conjunto de abstracciones ordenadas de forma que se obtiene un conocimiento más profundo del problema

www.dsic.upv.es/~uml

73

## Clases

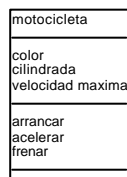
- La clase define el ámbito de definición de un conjunto de objetos
- Cada objeto pertenece a una clase
- Los objetos se crean por instanciación de las clases

www.dsic.upv.es/~uml

74

## Clases: Notación Gráfica

- Cada clase se representa en un rectángulo con tres compartimentos:
  - nombre de la clase
  - atributos de la clase
  - operaciones de la clase



www.dsic.upv.es/~uml

75

## Clases: Encapsulación

- La encapsulación presenta dos ventajas básicas:
  - Se protegen los datos de accesos indebidos
  - El acoplamiento entre las clases se disminuye
  - Favorece la modularidad y el mantenimiento
- Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos

www.dsic.upv.es/~uml

76

## Relaciones entre Clases

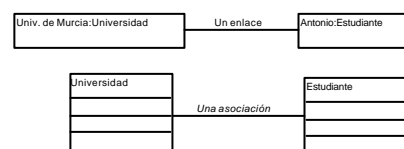
- Los enlaces entre objetos pueden representarse entre las respectivas clases
- Formas de relación entre clases:
  - Asociación y Agregación (vista como un caso particular de asociación)
  - Generalización/Especialización
- Las relaciones de Agregación y Generalización forman jerarquías de clases

www.dsic.upv.es/~uml

77

## Asociación

- La asociación expresa una conexión bidireccional entre objetos
- Una asociación es una abstracción de la relación existente en los enlaces entre los objetos

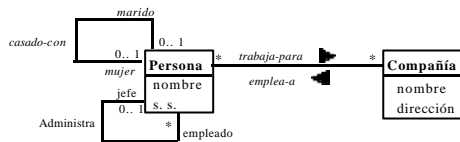


www.dsic.upv.es/~uml

78

## ... Asociación

### Ejemplo:



www.dsic.upv.es/~uml

79

## ... Asociación

### Especificación de multiplicidad (mínima...máxima)

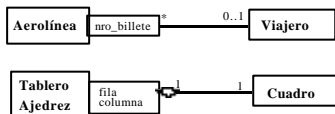
- ✦ 1 Uno y solo uno
- ✦ 0..1 Cero o uno
- ✦ M..N Desde M hasta N (enteros naturales)
- ✦ \* Cero o muchos
- ✦ 0..\* Cero o muchos
- ✦ 1..\* Uno o muchos (al menos uno)

- ✦ La multiplicidad mínima  $\geq 1$  establece una restricción de existencia

www.dsic.upv.es/~uml

80

## Asociación Cualificada



Reduce la multiplicidad del rol opuesto al considerar el valor del cualificador

www.dsic.upv.es/~uml

81

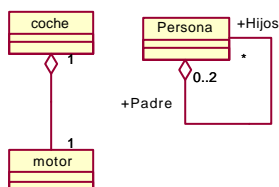
## Agregación

- ✦ La agregación representa una relación parte\_de entre objetos
- ✦ En UML se proporciona una escasa caracterización de la agregación
- ✦ Puede ser caracterizada con precisión determinando las relaciones de comportamiento y estructura que existen entre el objeto agregado y cada uno de sus objetos componentes

www.dsic.upv.es/~uml

82

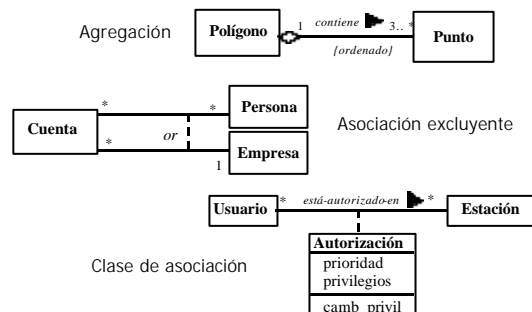
## Ejemplos



www.dsic.upv.es/~uml

83

## ... Ejemplos



www.dsic.upv.es/~uml

84

## Jerarquías de Generalización/Especialización

- Permiten gestionar la complejidad mediante un ordenamiento taxonómico
- Se obtiene usando los mecanismos de abstracción de Generalización y/o Especialización
- La Generalización consiste en factorizar las propiedades comunes de un conjunto de clases en una clase más general

www.dsic.upv.es/~uml

85

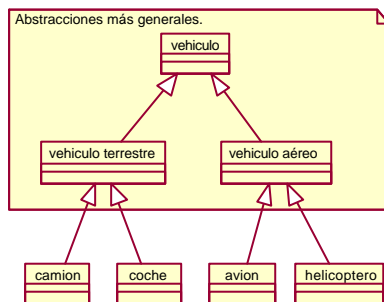
## ... Jerarquías de Generalización/Especialización

- Nombres usados: clase padre - clase hija, superclase - subclase, clase base - clase derivada
- Las subclases **heredan** características de sus superclases, es decir, atributos y operaciones (y asociaciones) de la superclase están disponibles en sus subclases

www.dsic.upv.es/~uml

86

## ... Jerarquías de Generalización/Especialización

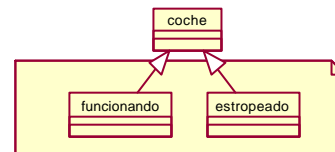


www.dsic.upv.es/~uml

87

## ... Jerarquías de Generalización/Especialización

- La especialización es una técnica muy eficaz para la extensión y reutilización



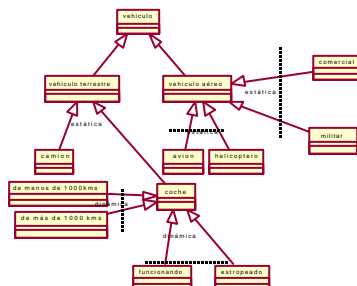
- Caracterización de la generalización en UML:
  - ✦ disjunta - no disjunta
  - ✦ total (completa) - parcial (incompleta)

www.dsic.upv.es/~uml

88

## ... Jerarquías de Generalización/Especialización

- Un ejemplo combinado:



www.dsic.upv.es/~uml

89

## Herencia Múltiple

- Se presenta cuando una subclase tiene más de una superclase
- La herencia múltiple debe manejarse con precaución. Algunos problemas son el conflicto de nombre y el conflicto de precedencia
- Se recomienda un uso restringido y disciplinado de la herencia. Java y Ada 95 simplemente no ofrecen herencia múltiple

www.dsic.upv.es/~uml

90

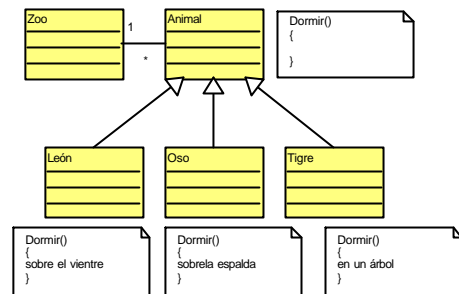
## Polimorfismo

- El término polimorfismo se refiere a que una característica de una clase puede tomar varias formas
- El polimorfismo representa la posibilidad de desencadenar operaciones distintas en respuesta a un mismo mensaje
- Cada subclase hereda las operaciones pero tiene la posibilidad de modificar localmente el comportamiento de estas operaciones

www.dsicupv.es/~uml

91

## Polimorfismo



www.dsicupv.es/~uml

92

## ... Polimorfismo

- La búsqueda automática del código que en cada momento se va a ejecutar es fruto del enlace dinámico
- El cumplimiento del Principio de Sustitución permite obtener un comportamiento y diseño coherente

www.dsicupv.es/~uml

93

## Diagramas de Estados

www.dsicupv.es/~uml

94

## Diagramas de Estados

- Los Diagramas de Estados representan autómatas de estados finitos, desde el p.d.v. de los estados y las transiciones
- Son útiles sólo para los objetos con un comportamiento significativo
- El resto de objetos se puede considerar que tienen un único estado
- El formalismo utilizado proviene de los Statecharts (Harel)

www.dsicupv.es/~uml

95

## ... Diagramas de Estados

- Cada objeto está en un estado en cierto instante
- El estado está caracterizado parcialmente por los valores de los atributos del objeto
- El estado en el que se encuentra un objeto determina su comportamiento
- Cada objeto sigue el comportamiento descrito en el D. de Estados asociado a su clase
- Los D. de Estados y escenarios son complementarios

www.dsicupv.es/~uml

96



### ... Diagramas de Estados

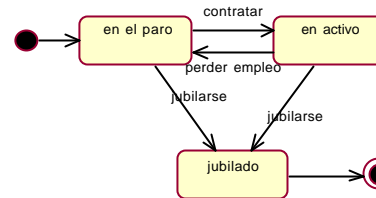
- Los D. de Estados son autómatas jerárquicos que permiten expresar concurrencia, sincronización y jerarquías de objetos
- Los Diagramas de Estados son grafos dirigidos
- Los D. De Estados de UML son deterministas
- Los estados inicial y final están diferenciados del resto
- La transición entre estados es instantánea y se debe a la ocurrencia de un evento

www.dsic.upv.es/~uml

97

### ... Diagramas de Estados

- Ejemplo de un Diagrama de Estados para la clase persona:

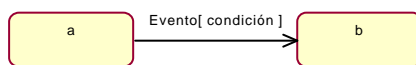


www.dsic.upv.es/~uml

98

### ... Diagramas de Estados

- Las guardas permiten condicionar la transición:

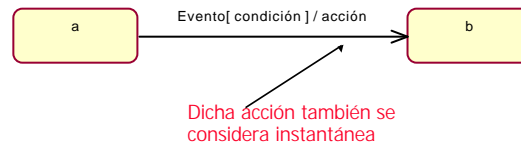


www.dsic.upv.es/~uml

99

### Acciones

- Podemos especificar la ejecución de una acción como consecuencia de la transición:

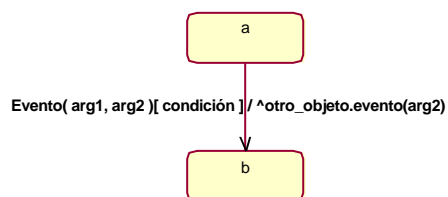


www.dsic.upv.es/~uml

100

### ... Acciones

- Podemos especificar el envío de un evento a otro objeto como consecuencia de la transición:

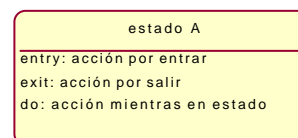


www.dsic.upv.es/~uml

101

### ... Acciones

- Se puede especificar el hacer una acción como consecuencia de entrar, salir o estar en un estado:

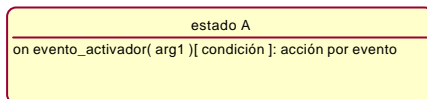


www.dsic.upv.es/~uml

102

## .. Acciones

- Se puede especificar el hacer una acción cuando ocurre en dicho estado un evento que no conlleva salir del estado:



www.dsicupv.es/~uml

103

## Actividades

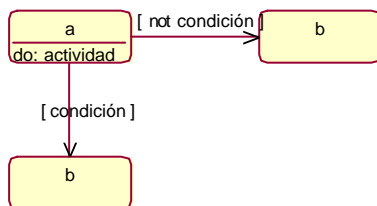
- Las actividades son similares a las acciones pero tienen duración y se ejecutan dentro de un estado del objeto
- Las actividades pueden interrumpirse en todo momento, cuando se desencadena la operación de salida del estado

www.dsicupv.es/~uml

104

## ... Actividades

- Cuando una actividad finaliza se produce una transición automática de salida del estado



www.dsicupv.es/~uml

105

## Generalización de Estados

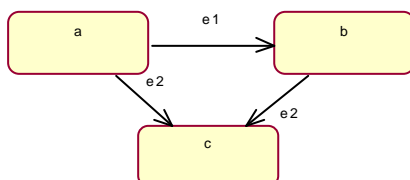
- Podemos reducir la complejidad de estos diagramas usando la generalización de estados
- Distinguimos así entre **superestado** y **subestados**
- Un estado puede contener varios subestados disjuntos
- Los subestados heredan las variables de estado y las transiciones externas

www.dsicupv.es/~uml

106

## Generalización de Estados

- Ejemplo:

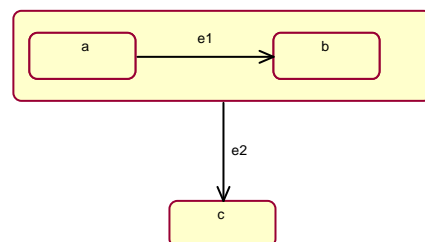


www.dsicupv.es/~uml

107

## Generalización de Estados

- Quedaría como:

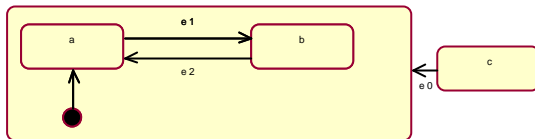


www.dsicupv.es/~uml

108

### ... Generalización de Estados

- Es preferible tener estados iniciales de entrada a un nivel de manera que desde los niveles superiores no se sepa a qué subestado se entra:



www.dsic.upv.es/~uml

109

### ... Generalización de Estados

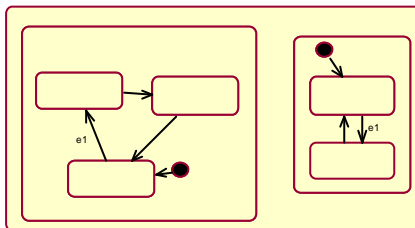
- La agregación de estados es la composición de un estado a partir de varios estados independientes
- La composición es concurrente por lo que el objeto estará en alguno de los estados de cada uno de los subestados concurrentes

www.dsic.upv.es/~uml

110

### ... Generalización de Estados

- Ejemplo:



www.dsic.upv.es/~uml

111

### Dstrucción del Objeto

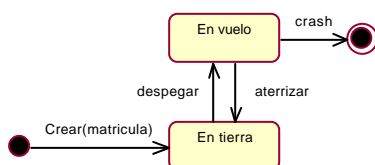
- La destrucción de un objeto es efectiva cuando el flujo de control del autómata alcanza un estado final no anidado
- La llegada a un estado final anidado implica la "subida" al superestado asociado, no el fin del objeto

www.dsic.upv.es/~uml

112

### ... Dstrucción de Objeto

- Ejemplo:



www.dsic.upv.es/~uml

113

### Transiciones temporizadas

- Las esperas son actividades que tienen asociada cierta duración
- La actividad de espera se interrumpe cuando el evento esperado tiene lugar
- Este evento desencadena una transición que permite salir del estado que alberga la actividad de espera. El flujo de control se transmite entonces a otro estado

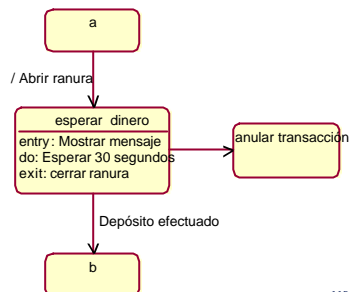
www.dsic.upv.es/~uml

114

### ... Transiciones temporizadas

#### ■ Ejemplo:

Si en 30 segundos no se introduce el dinero se termina la actividad pasando a anular la transacción. En cualquier caso se cierra la ranura.

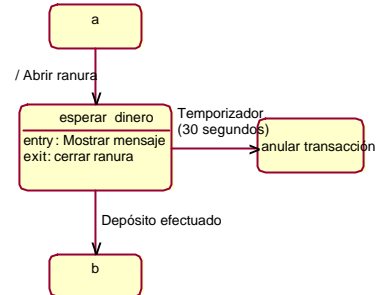


www.dsic.upv.es/~uml

115

### ... Transiciones temporizadas

#### ■ Ejemplo v.2:



www.dsic.upv.es/~uml

116

## Modelado de Componentes

www.dsic.upv.es/~uml

117

## Diagrama de Componentes

- Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones
- Muestran las opciones de realización incluyendo código fuente, binario y ejecutable

www.dsic.upv.es/~uml

118

### ... Diagramas de Componentes

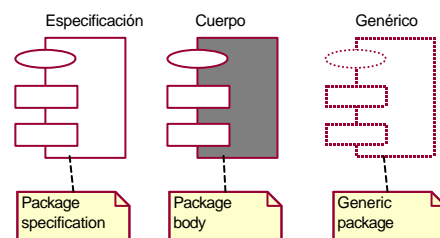
- Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc.
- Cada clase del modelo lógico se realiza en dos componentes: la **especificación** y el **cuerpo**

www.dsic.upv.es/~uml

119

### ... Diagramas de Componentes

- La representación gráfica es la siguiente:



www.dsic.upv.es/~uml

120

## Dependencias entre Componentes

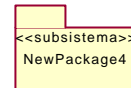
- Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente

www.dsic.upv.es/~uml

121

## Subsistemas

- Los distintos componentes pueden agruparse en paquetes según un criterio lógico y con vistas a simplificar la implementación
- Son paquetes estereotipados en `<<subsistema>>`



www.dsic.upv.es/~uml

122

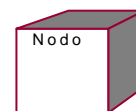
## Modelado de Distribución

www.dsic.upv.es/~uml

123

## Diagramas de Distribución

- Los Diagramas de Distribución muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos



www.dsic.upv.es/~uml

124

## ... Diagramas de Distribución

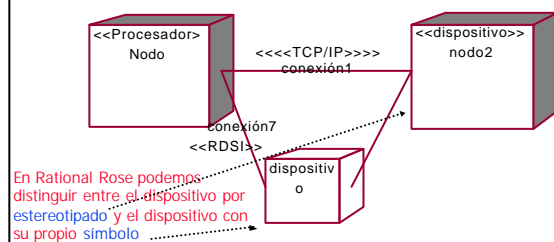
- Los estereotipos permiten precisar la naturaleza del equipo:
  - ✖ Dispositivos
  - ✖ Procesadores
  - ✖ Memoria
- Los nodos se interconectan mediante soportes bidireccionales (en principio) que pueden a su vez estereotiparse

www.dsic.upv.es/~uml

125

## ... Diagramas de Distribución

- Ejemplo de conexión entre nodos:



www.dsic.upv.es/~uml

126