

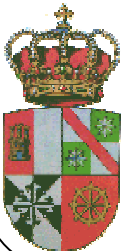
# Capítulo 3

## El lenguaje unificado de modelado

---

Diseño y Programación Orientado a Objetos  
Ingeniería Informática  
Ingeniería Técnica de Informática de Sistemas y Gestión  
Optativa (6 créditos)

<http://www.info-ab.uclm.es/asignaturas/42579>



# Objetivos

---

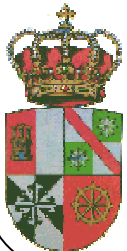
- Presentar el lenguaje de modelado UML, diseñado para desarrollar, documentar e intercambiar modelos de información significativa de sistemas
- Conocer algunas de las herramientas o diagramas proporcionadas por UML para su utilización en el siguiente capítulo de la asignatura (Patrones de diseño)



# Bibliografía

---

- UML gota a gota  
M. Fowler con Kendal Scott  
Prentice-Hall. 1999
- El lenguaje unificado de modelado  
G. Booch, J. Rumbaugh, I. Jacobson  
Addison-Wesley, 1999
- El lenguaje unificado de modelado: manual de referencia  
J. Rumbaugh, I. Jacobson, G. Booch.  
Addison Wesley, 2000



# Introducción

---

- Desde los inicios de la informática se han estado utilizando distintas formas de representar los diseños de una forma más bien personal o con algún modelo gráfico
- La falta de estandarización en la manera de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores
- Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: Unified Modeling Language).
- UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente de diseño



# Historia de UML

---

- El lenguaje UML comienza a gestarse en octubre de 1994
- Rumbaugh y Booch se unen en *Rational* con el objetivo de unificar dos métodos que habían desarrollado: el método Booch y el OMT (*Object Modelling Tool*)
- En 1995 se les une también Jacobson
- Todas estas colaboraciones conducen a la definición de la primera versión de UML



# ¿Qué es UML?

---

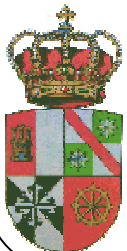
- UML es un **lenguaje**. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación
- Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo
- Los objetivos de UML son:
  - **Visualizar**, expresa de forma gráfica
  - **Especificar**, las características de un sistema
  - **Construir**, a partir de los modelos especificados
  - **Documentar**, los propios elementos gráficos sirven de documentación



# ¿Qué forma un modelo de UML?

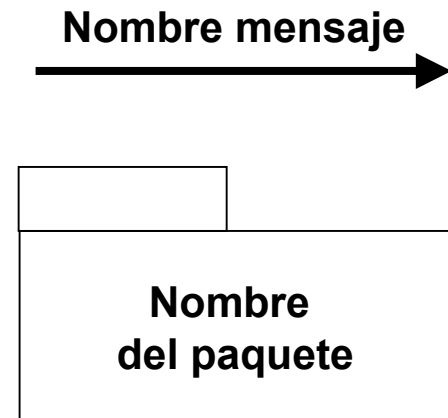
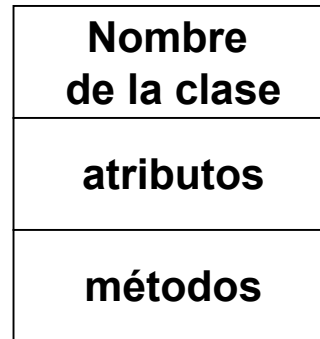
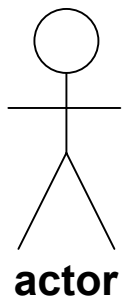
---

- Un modelo UML está compuesto por tres clases de bloques de construcción
  - Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
  - Relaciones: relacionan los elementos entre sí
  - Diagramas: Son colecciones de elementos con sus relaciones



# Elementos

- Pueden ser estructurales, de comportamiento de agrupación o de anotación
- Elementos **estructurales**: actores, casos de uso, clases, objetos
- Elementos de **comportamiento**: mensajes
- Elementos de **agrupación**: paquetes

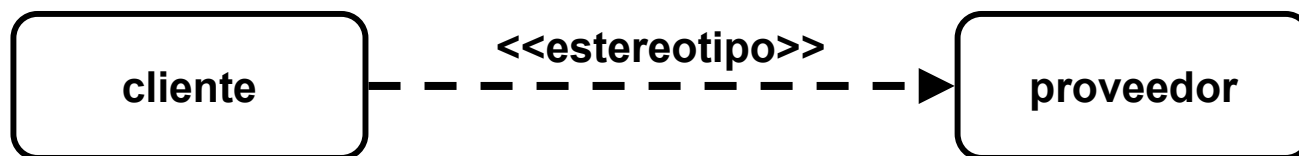


# Relaciones (i)

---

## Dependencia

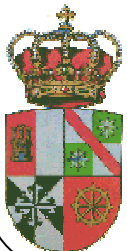
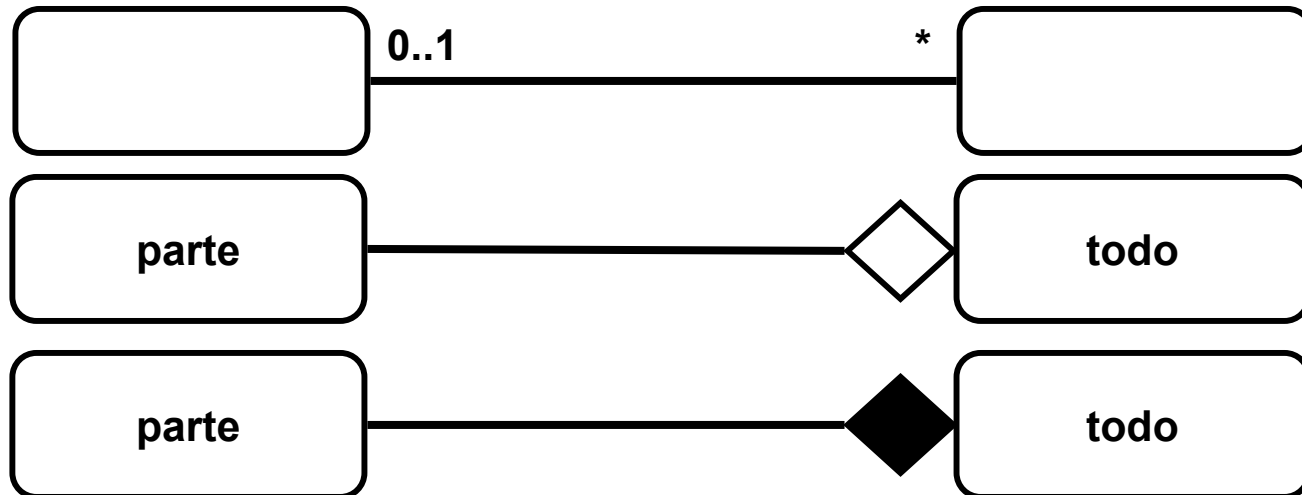
- Es una relación semántica entre dos elementos, en la cual un cambio en un elemento puede afectar a la semántica de otro elemento
- Existen varios tipos de dependencia predefinidas que se indican mediante estereotipos <<extend>> o <<include>> para casos de uso



# Relaciones (ii)

## Asociación

- Es una relación estructural entre dos elementos, que describen las conexiones entre ellos (suele ser bidireccional)
- Puede presentarse como agregación o composición

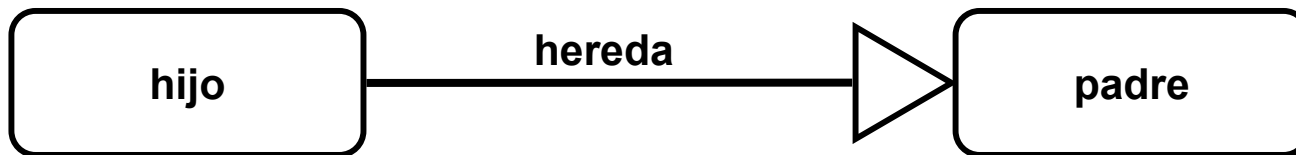


# Relaciones (iii)

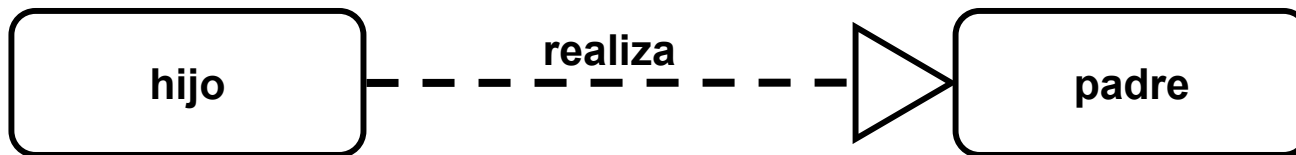
---

## Generalización

- Es una relación taxonómica entre un elemento más general (el padre) y un elemento más específico (el hijo)
- Se utiliza tanto en diagramas de clases como en diagramas de casos de uso



## Implementa (realiza)



# Diagramas (i)

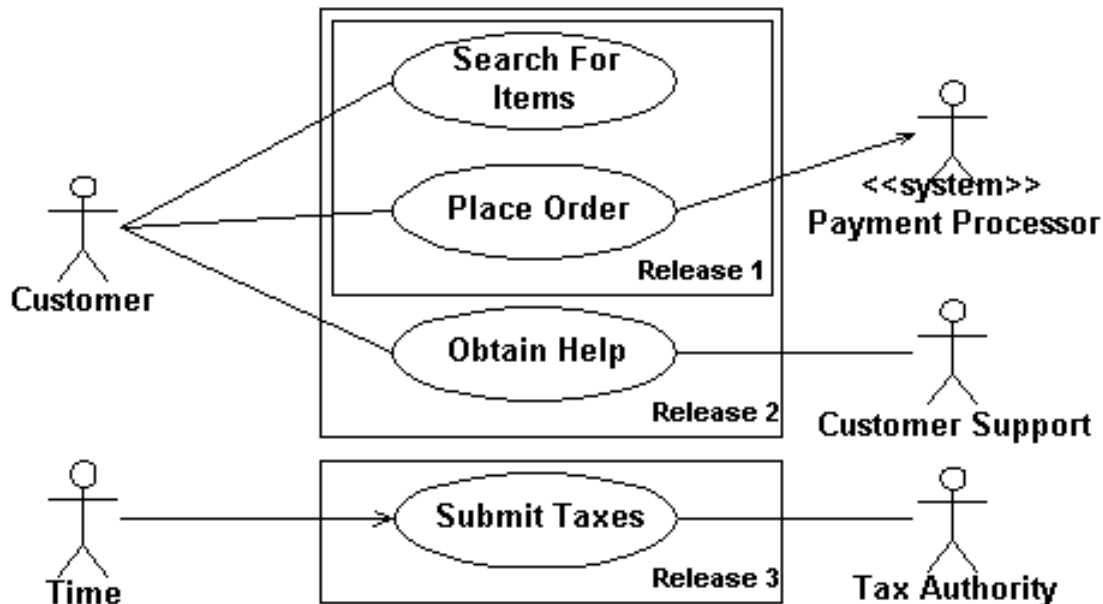
---

- Diagramas de casos de uso
- Diagramas de clases
- Diagramas de objetos
- Diagramas de interacción
  - Secuencia
  - Colaboración
- Diagramas de actividad
- Diagramas de estados
- Diagramas de componentes



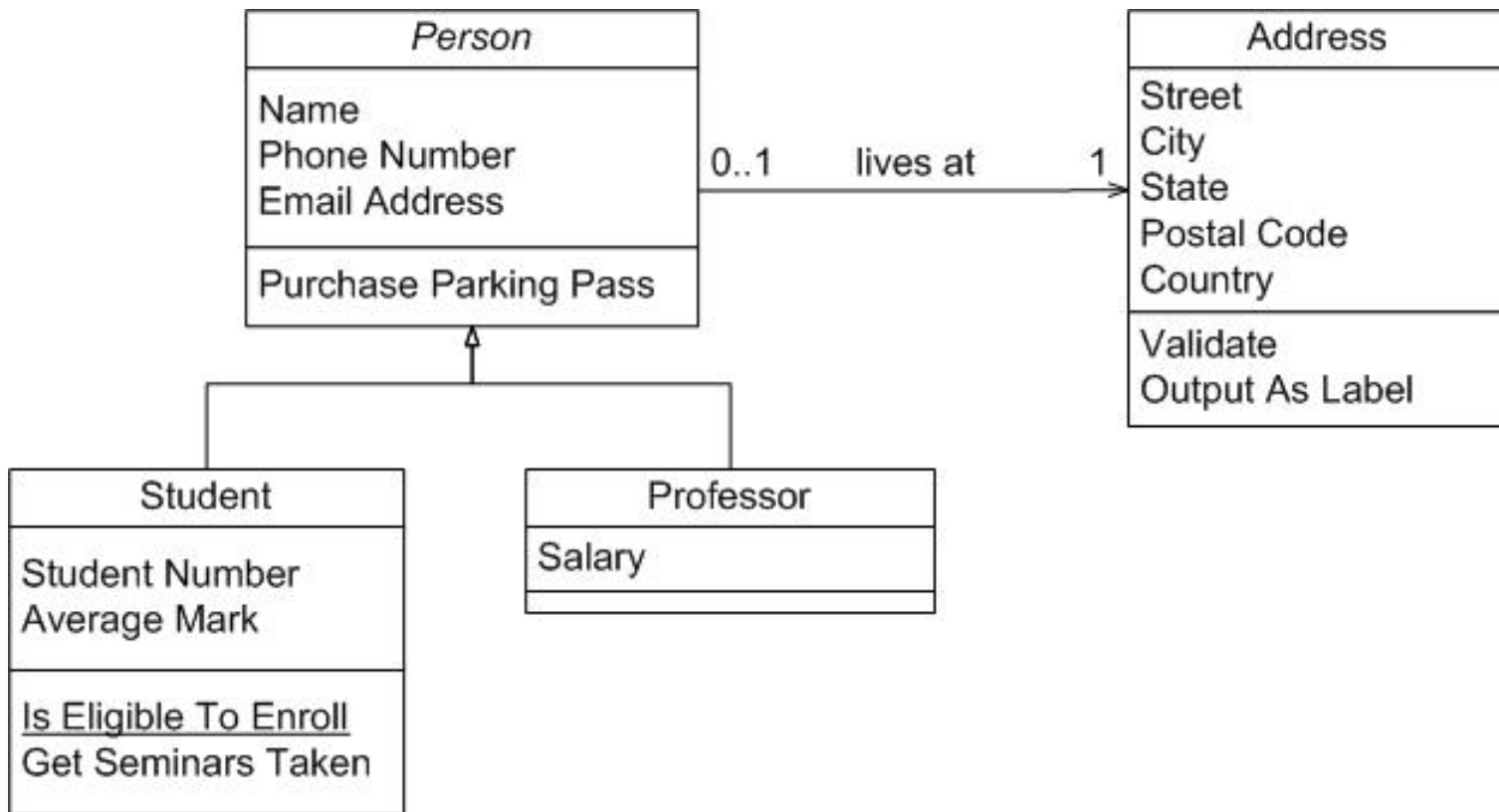
# Diagramas (ii)

- Diagrama de casos de uso



# Diagramas (iii)

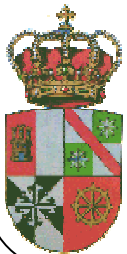
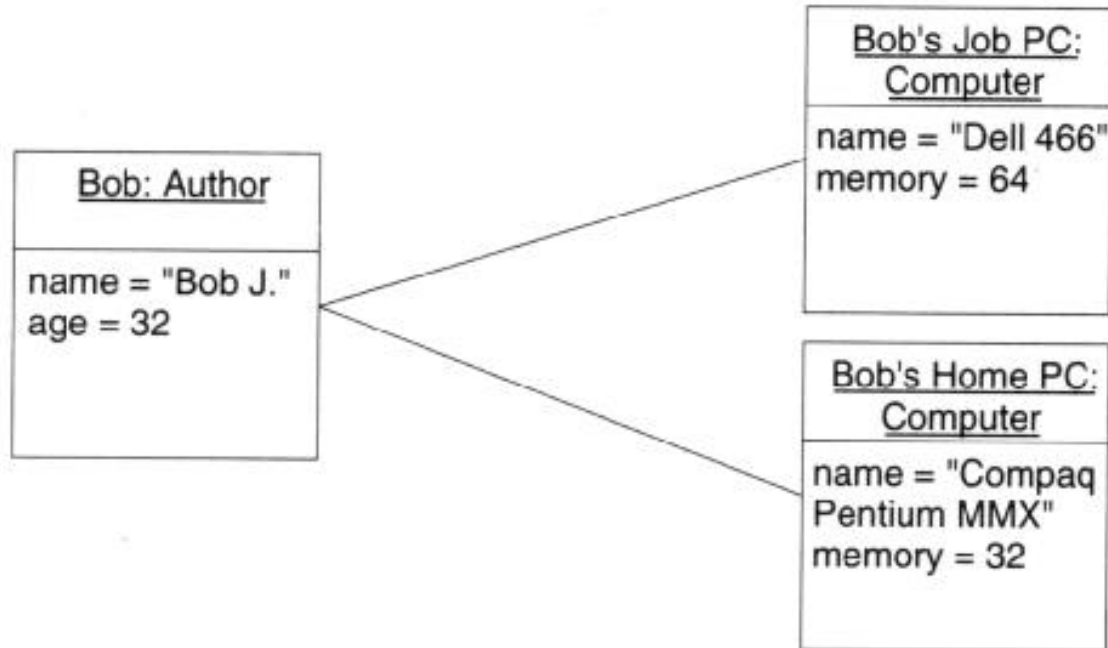
- Diagrama de clases



# Diagramas (iv)

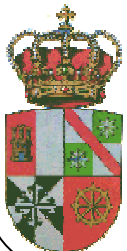
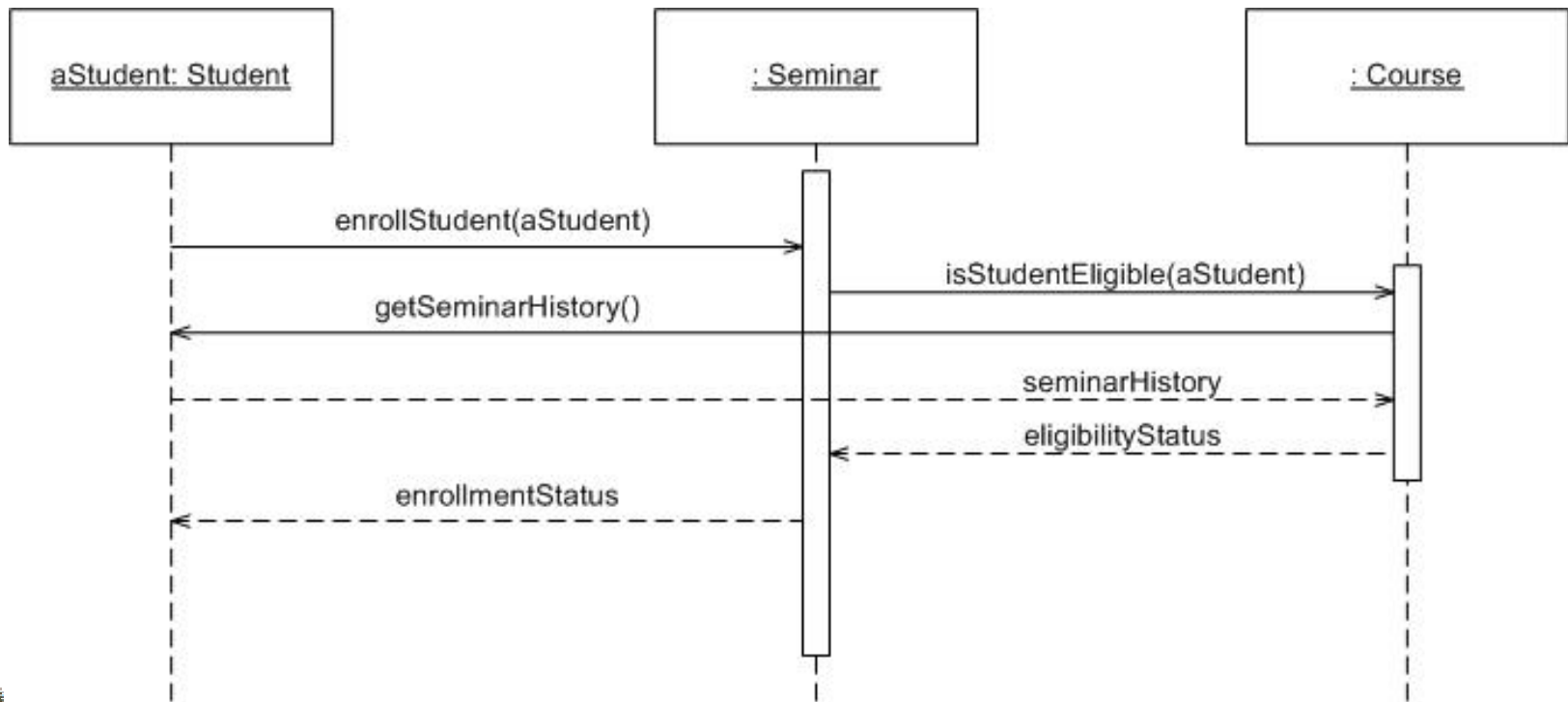
---

- Diagrama de objetos



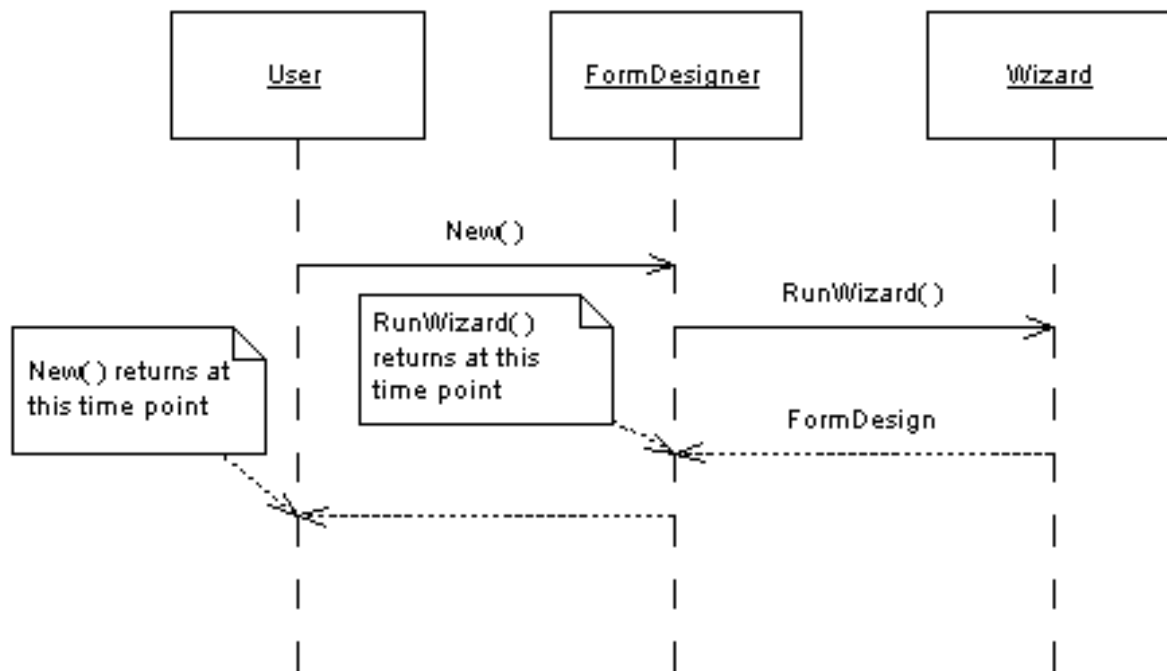
# Diagramas (v)

- Diagramas de interacción (secuencia)



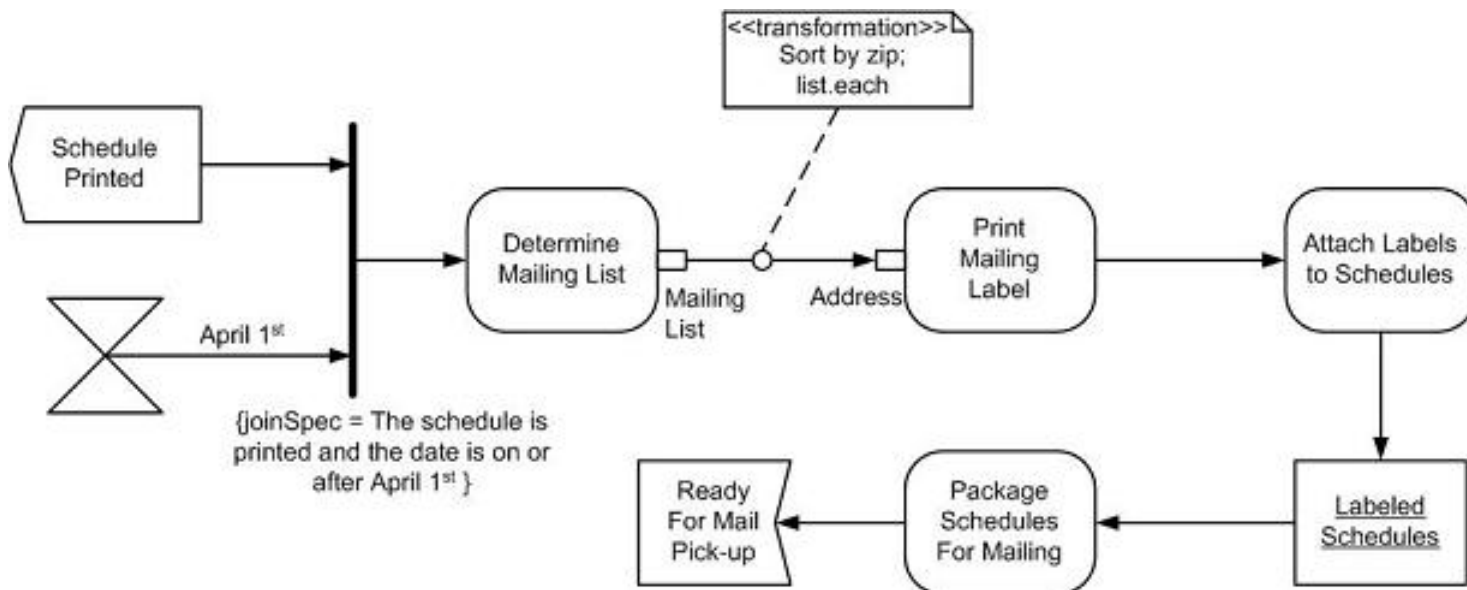
# Diagramas (vi)

- Diagrama de interacción (colaboración)



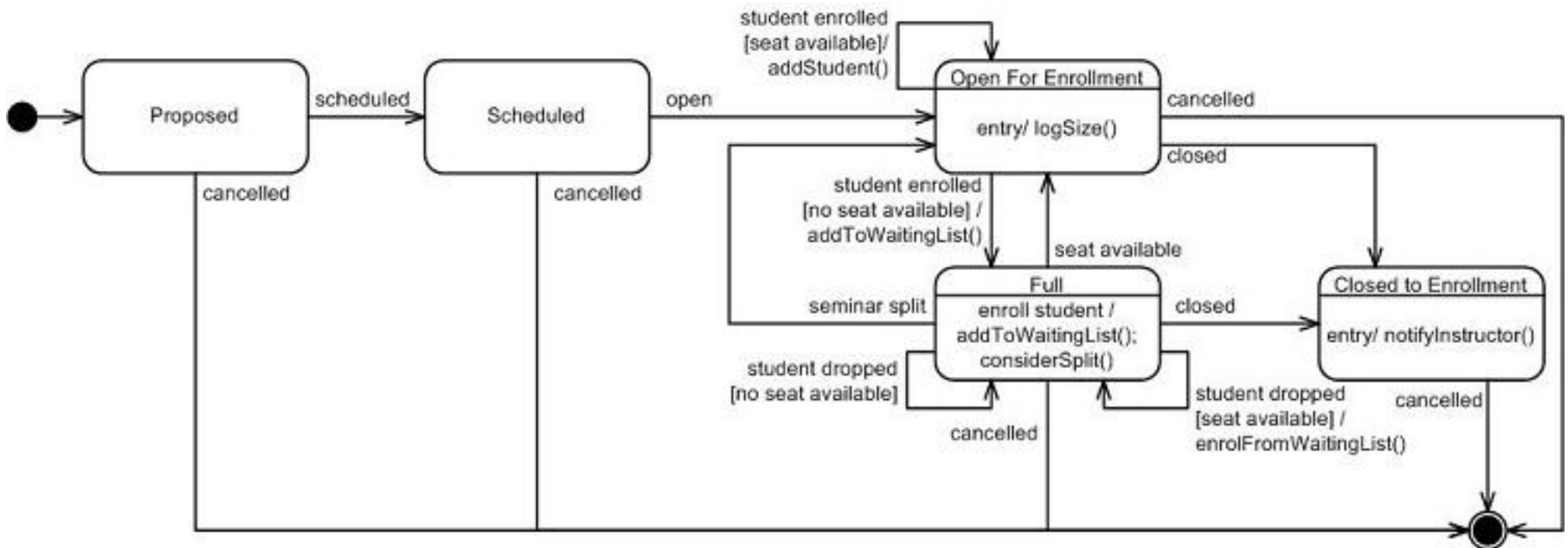
# Diagramas (vii)

- Diagramas de actividad



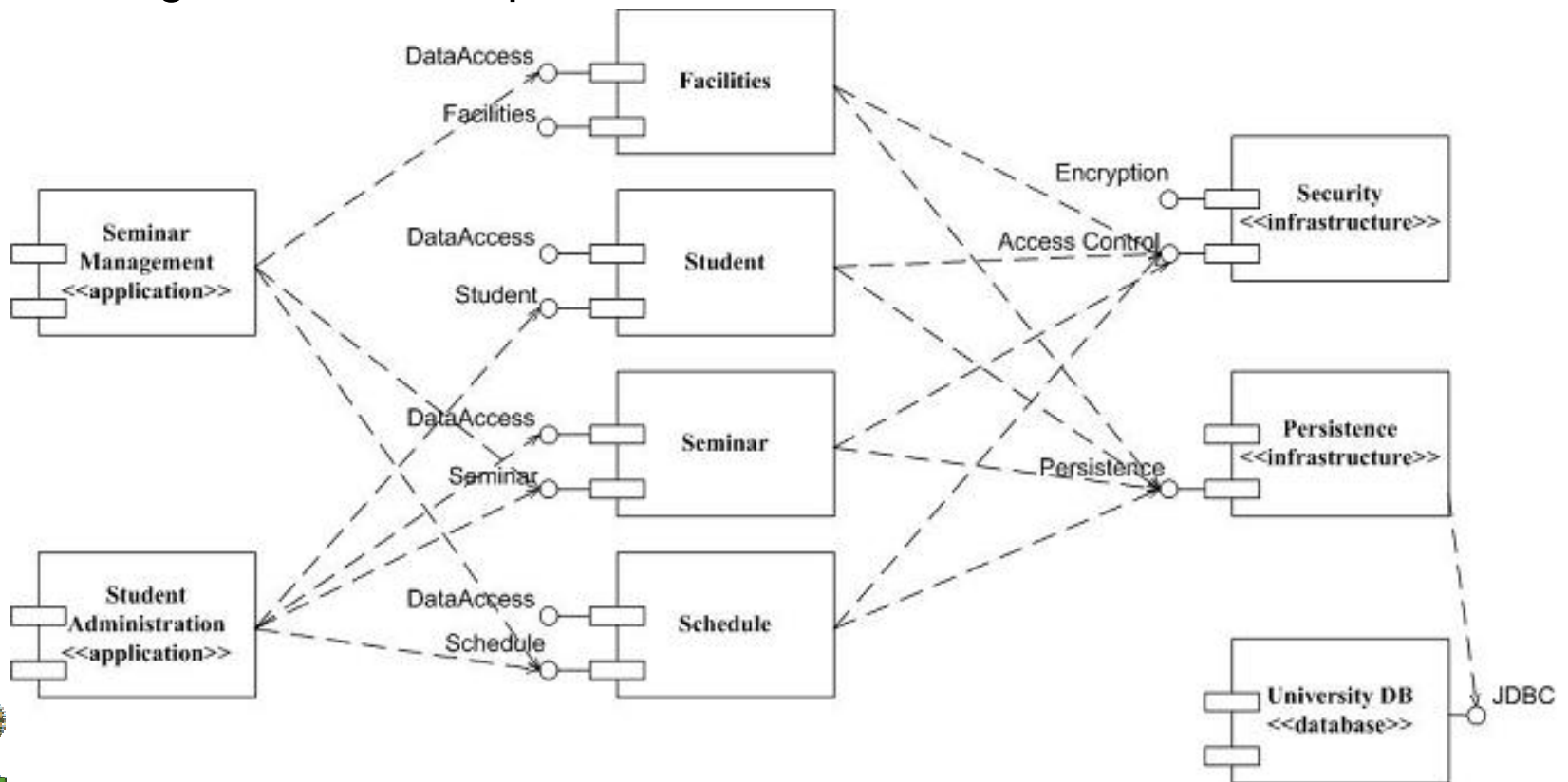
# Diagramas (viii)

- Diagramas de estados



# Diagramas (ix)

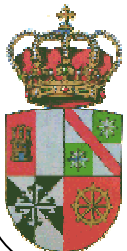
- Diagramas de componentes



# Resumen

---

- Es fácil predecir que UML será el lenguaje de modelado de software de uso universal
  - En el desarrollo han participado investigadores de reconocido prestigio
  - Ha sido apoyado por prácticamente todas las empresas importantes de informática
  - Se ha aceptado como un estándar por la OMG
  - Prácticamente todas las herramientas CASE y de desarrollo la han adaptado como lenguaje de modelado
- UML resuelve un viejo problema de desarrollo como es el modelado gráfico



# Herramienta: **poseidon** for uml

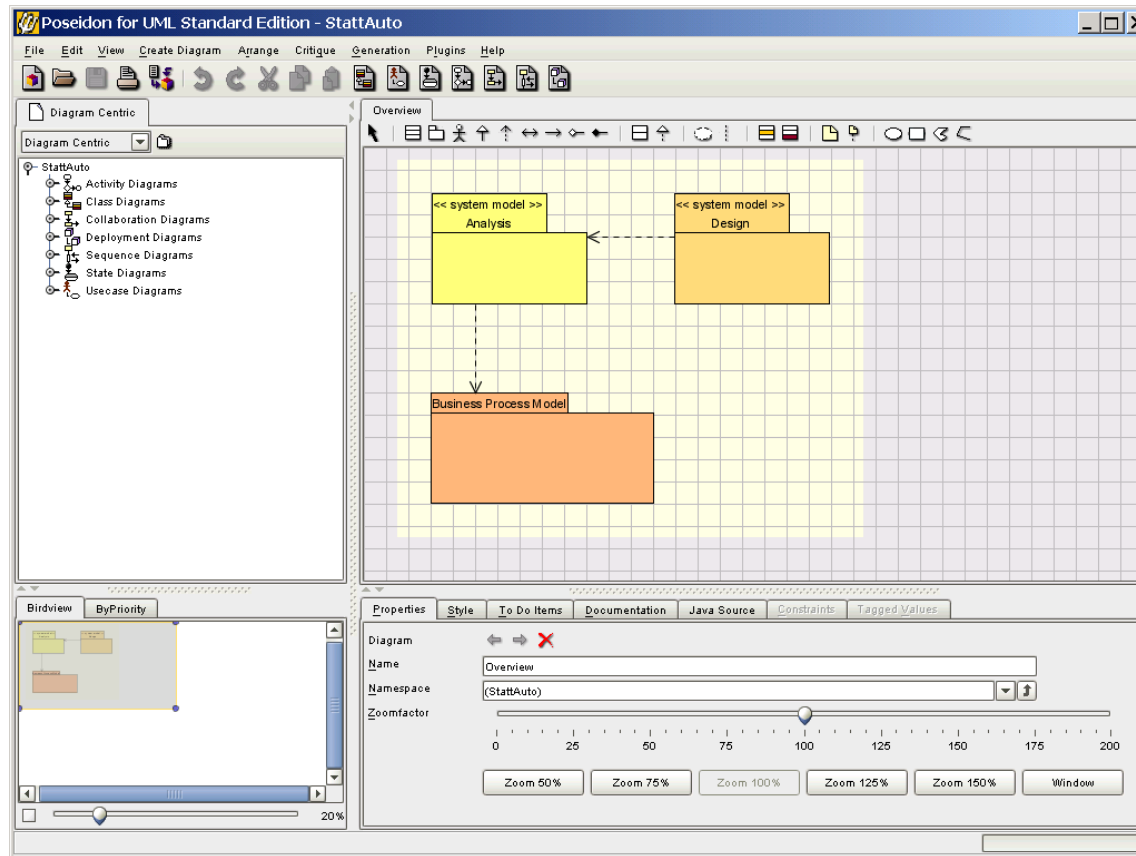
---



**poseidon** for uml  
community edition 2.0



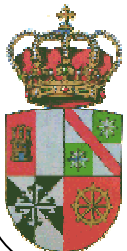
# poseidon for uml



# Diagramas de clases: atributos

The screenshot shows the 'Properties' tab of a UML class editor. The attribute is named 'Number' and is of type 'String'. It has a multiplicity of '1' and a stereotype of 'Key'. The owner is 'Reservation'. The visibility is set to 'private'. The modifiers are 'static', 'final', 'transient', and 'volatile'. The accessor methods are 'getNumber' and 'setNumber'.

Property	Value
Name	Number
Multiplicity	1
Stereotype	Key
Owner	Reservation
Visibility	private
Modifiers	static, final, transient, volatile
Type	String
Initial Value	
Accessor methods	getNumber, setNumber



# Diagramas de clases: métodos

Properties | Style | To Do Items | Documentation | Java Source | Constraints | Tagged Values

Operation

Name: userRegistration

<< >> Stereotype: [ ]

Owner: UserReservationProcessObject

Accessed Attribute: [ ]

Visibility:  public  protected  package  private

Modifiers:  abstract  final  root  query  static

Concurrency:  sequential  synchronized  concurrent

Parameters: return, login, password

Raised Signals: none



# Capítulo 3

## El lenguaje unificado de modelado

---

Diseño y Programación Orientado a Objetos  
Ingeniería Informática  
Ingeniería Técnica de Informática de Sistemas y Gestión  
Optativa (6 créditos)

<http://www.info-ab.uclm.es/asignaturas/42579>

