

# Principios de la Tecnología de Objetos

---



## Unified Modeling Language

## Copyright

- Copyright (c) 2004  
José M. Ordax
- Este documento puede ser distribuido solo bajo los términos y condiciones de la Licencia de Documentación de javaHispano v1.0 o posterior.
- La última versión se encuentra en  
<http://www.javahispano.org/licencias/>

# ¿Qué es UML?

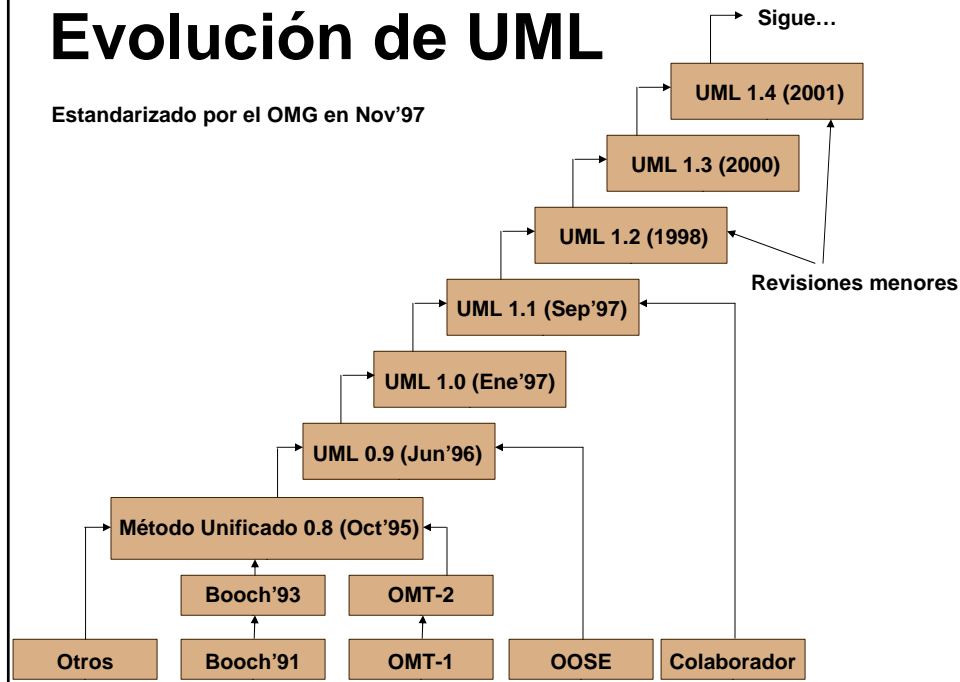
- Siglas de Unified Modeling Language.
- Es el sucesor de los métodos de Análisis y Diseño Orientado a Objetos de finales de los 80 y comienzo de los 90.
- Más específicamente unifica los métodos de:
  - Booch.
  - Rumbaugh (OMT).
  - Jacobson.
  - Más sus propias extensiones.

## Objetivo

- Lenguaje visual (diagramas) para modelar Sistemas.
- Modelos para:
  - Especificaciones.
  - Arquitectura.
  - Diseño.
  - Implementación.
- Distintos tipos de Sistemas:
  - Software.
  - Procesos de negocio.

# Evolución de UML

Estandarizado por el OMG en Nov'97

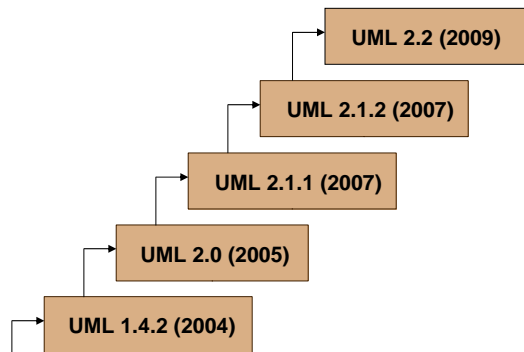


# Evolución de UML

Continuación...



<http://www.omg.org/spec/UML/2.2/>



## Tipos de diagramas UML

- Casos de Uso.
- Diagramas de Clase.
- Diagramas de Interacción:
  - Diagramas de Secuencia.
  - Diagramas de Colaboración.
- Diagramas de Estado.
- Diagramas de Actividad.
- Diagramas Físicos:
  - Diagramas de Componentes.
  - Diagramas de Implantación.

## Casos de Uso

- Un Caso de Uso es una manera específica de utilizar el Sistema que se está analizando.
- La terminología empleada en la redacción de los casos de uso es la empleada por los usuarios en su trabajo cotidiano.
- Describen una visión externa del comportamiento del Sistema desde el punto de vista del usuario.
- Constituye un modelo de lo que el Sistema hará sin tener en cuenta el cómo lo hará.

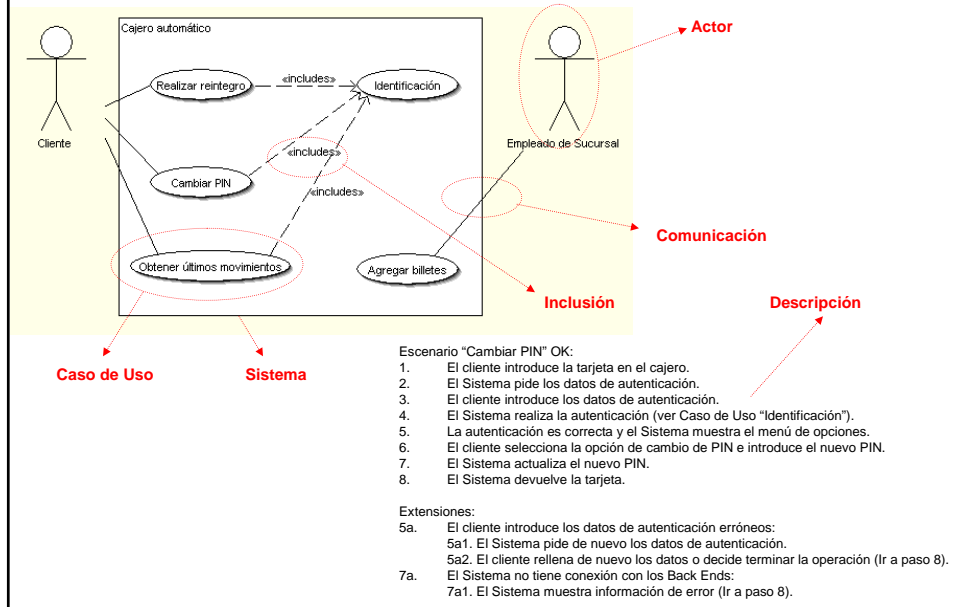
## Casos de Uso (cont.)

- Los Casos de Uso se construyen al comienzo para determinar los requisitos funcionales del Sistema.
- Consisten en:
  - Los diagramas de Casos de Uso.
  - Las descripciones de los Casos de Uso.

## Elementos

- Escenario: es una casuística concreta dentro de un Caso de Uso.
- Actor: representa un papel interpretado por una persona o cosa que interactúa con el Sistema.
- Relaciones: interacciones entre los distintos elementos:
  - Comunicación.
  - Inclusión.
  - Generalización.

# Ejemplo



# Diagrama de Clases

- Representa las clases que componen el Sistema y sus relaciones estáticas.
- Incluye:
  - Clases con sus atributos y operaciones.
  - Asociaciones entre las clases.
  - Restricciones sobre las clases y asociaciones.
  - Datos relevantes de las asociaciones.

# Clases, atributos y operaciones

Las clases se representan mediante un rectángulo dividido en tres partes.

El nombre de la clase va arriba.

La definición de los atributos va en medio.

*visibilidad nombre: tipo = valor por defecto*

La definición de los métodos va abajo.

*visibilidad nombre (dirección lista-parámetros) : tipo retorno*

## Ejemplo de una clase

```
edu.upco.einf.practica7.figuras:Punto
┌ x: int
└ y: int

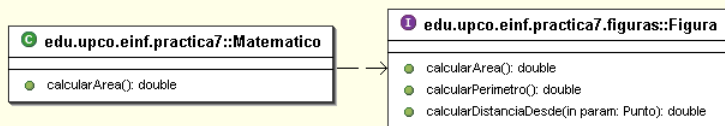
● Punto(in param1: int, in param2: int): void
● calcularArea(): double
● calcularDistanciaDesde(in param: Punto): double
● calcularPerimetro(): double
● getX(): int
● getY(): int
● setX(in x: int): void
● setY(in y: int): void
```

# Asociaciones

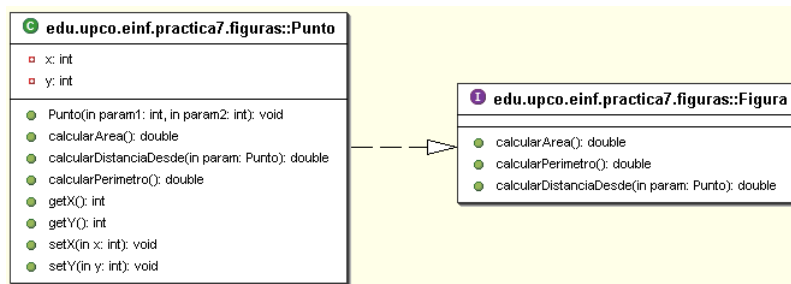
- Representan relaciones entre clases.
- Son bidireccionales. Su navegación no siempre.
- Cada extremo puede llevar un nombre llamado Rol. Explica el significado de la asociación en ese sentido.
- Pueden poseer multiplicidad:
  - 1: uno y sólo uno.
  - 0..1: cero o uno.
  - 0..\* ó \*: cero o muchos.
  - 1..\*: uno o muchos.

## Asociaciones (cont.)

- Existen varios tipos:
  - Asociación normal: relación simple.

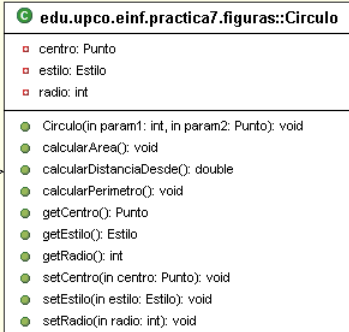
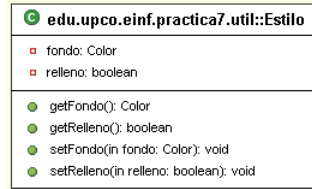


- Herencia: relación jerárquica.

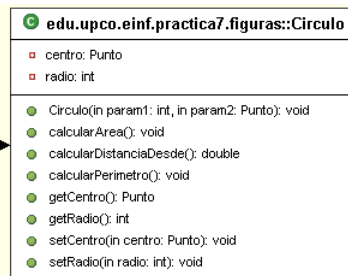
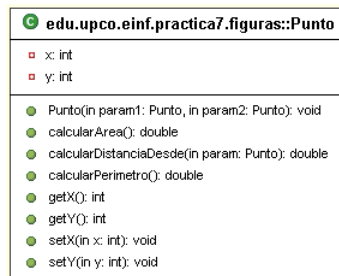




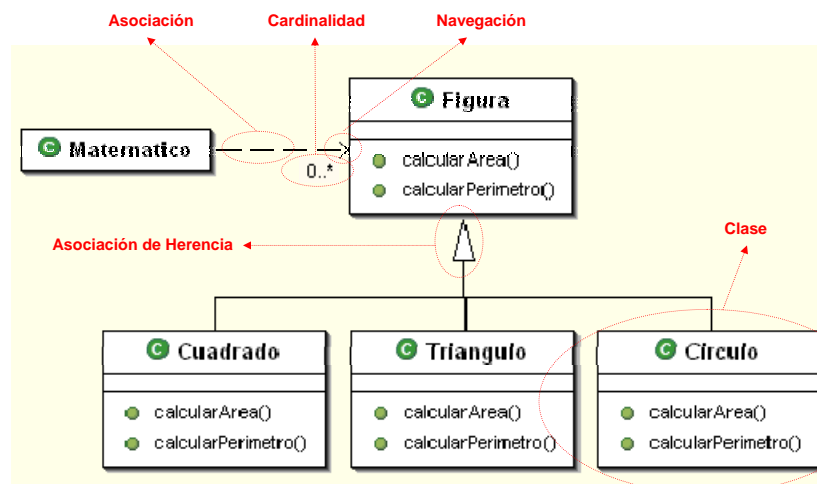
## Agregación: contenido en...



## Composición: agregación más fuerte, uno no existe sin otro.



## Ejemplo

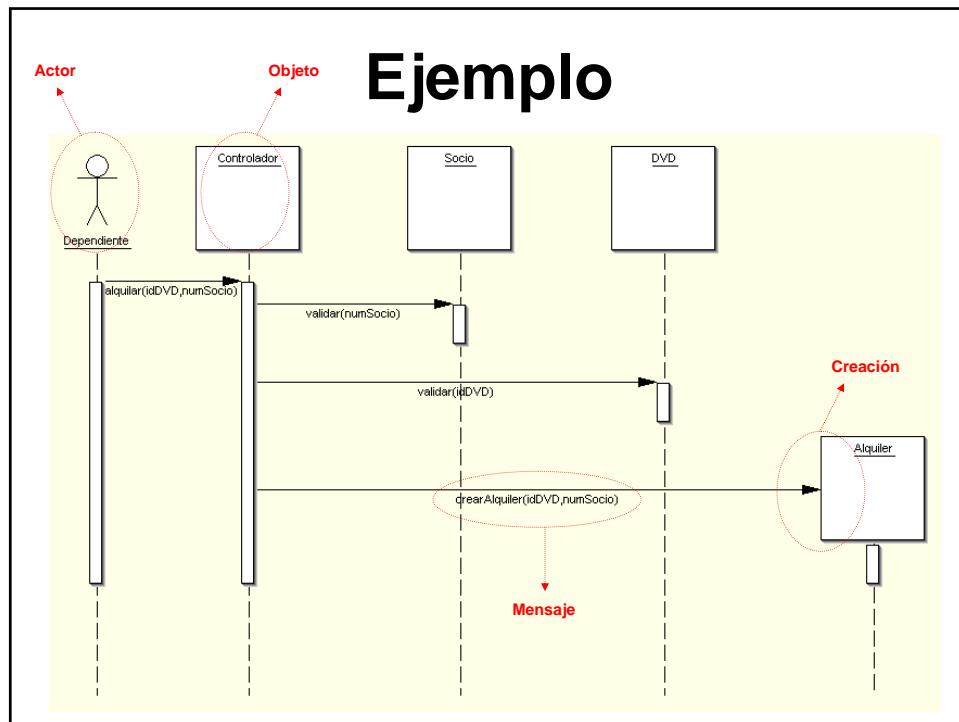


## Diagrama de Interacción




- Muestran cómo los objetos de la aplicación cooperan e interactúan para cumplir con los requisitos.
- Suele construirse uno para cada uno de los escenarios de cada Caso de Uso.
- Existen dos tipos:
  - Diagramas de Secuencia.
  - Diagramas de Colaboración.

## Diagrama de Secuencia

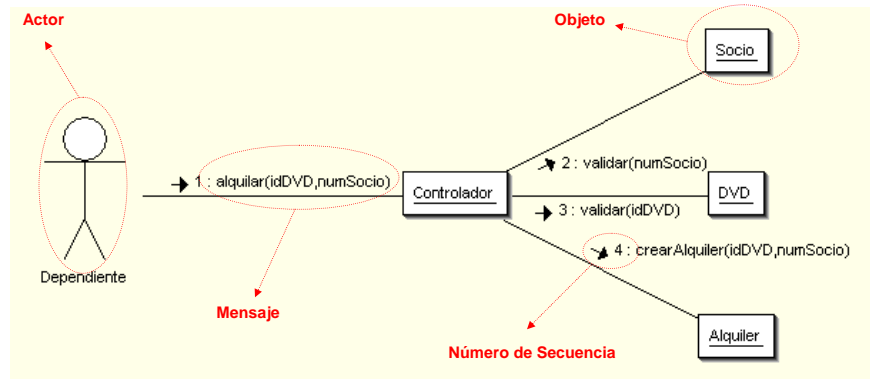
- Muestran cómo interactúan los objetos.
- Se centran en las secuencias de Mensajes, es decir, cómo los mensajes son enviados y recibidos por los Objetos.
- El tiempo se muestra en el eje vertical y los objetos en el eje horizontal.



## Diagrama de Colaboración

- 
 Muestran las interacciones y los enlaces entre un conjunto de objetos que colaboran entre sí.
- 
 Mientras los Diagramas de Secuencia se centran en el tiempo, los de Colaboración se centran en el espacio, es decir, no muestran dimensión temporal.
- 
 Un Diagrama de Colaboración comienza con un mensaje que inicializa la interacción.

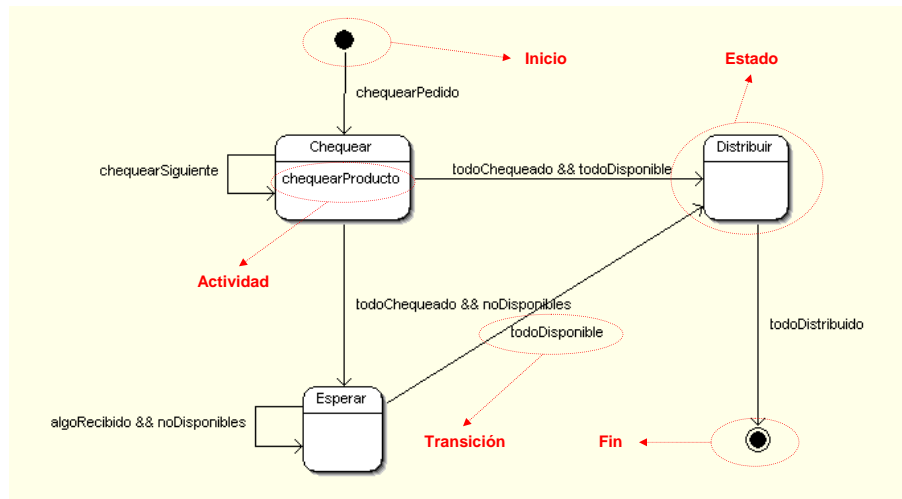
# Ejemplo



## Diagrama de Estados

- Representa el estado de un objeto en el tiempo.
- Consiste básicamente en:
  - Estado: situación concreta en la que se encuentra un objeto durante un tiempo indefinido determinado por el valor de sus atributos.
  - Evento: Algo que ocurre en un momento dado de forma asíncrona.
  - Transición: Respuesta de un objeto en un estado concreto a un evento. Suelen suponer un cambio de estado en el objeto.
  - Actividad: operación que realiza un objeto mientras se encuentra en un estado determinado.

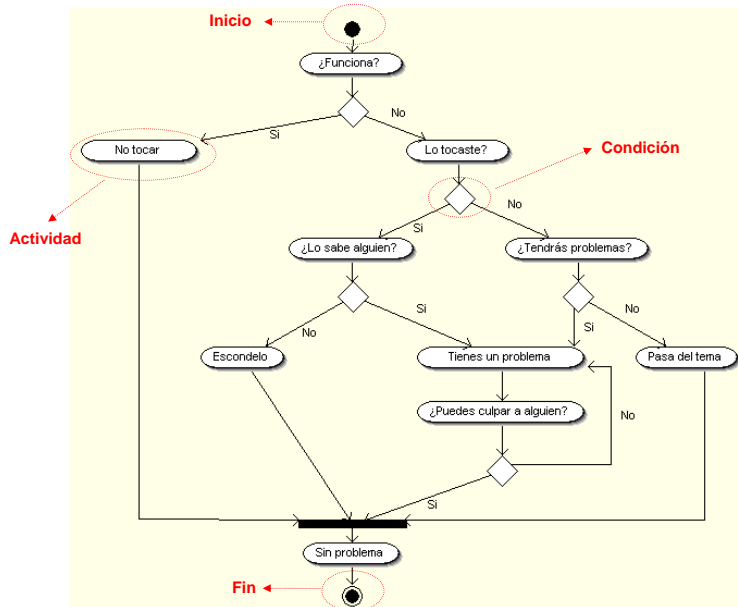
# Ejemplo



## Diagrama de Actividad

- Muestra los pasos a realizar para llevar a cabo una operación.
- Consta de:
  - Punto inicial y punto final para indicar dónde comienza y dónde termina la implementación de la operación.
  - Acciones que muestran lo que la operación va realizando.
  - Transiciones entre acciones.
  - Puntos de decisión o bifurcaciones (con rombos).

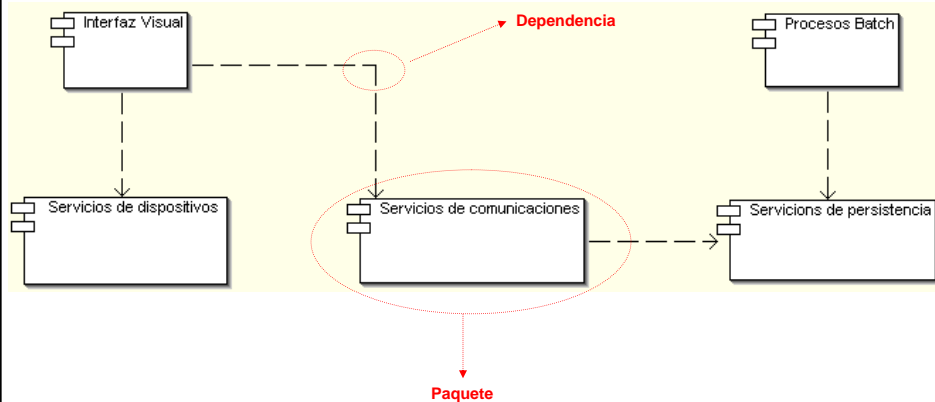
## Ejemplo



## Diagrama de Componentes

- Muestra varios componentes en un sistema y sus dependencias.
- Un componente representa un módulo de código Físico (paquete, clases, frameworks,...).
- Las dependencias entre los componentes muestran como los cambios en un componente pueden provocar cambios en otros.

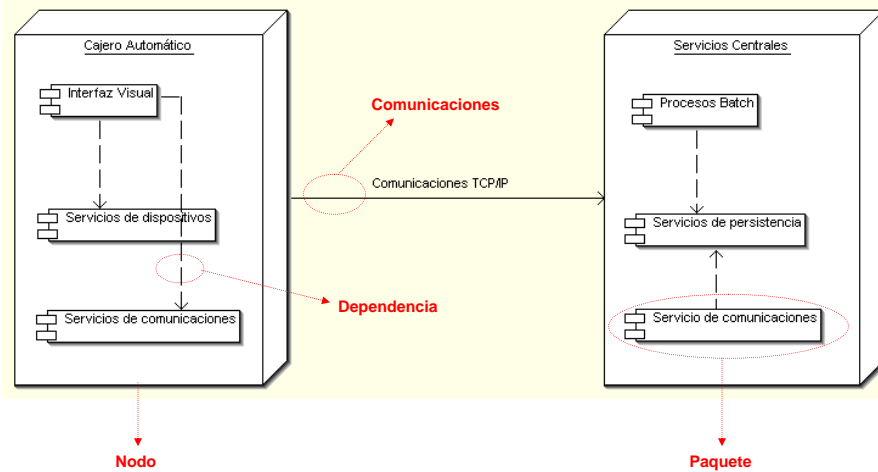
## Ejemplo



## Diagrama de Implantación

- Muestra las relaciones físicas entre los componentes hardware y software de un sistema.
- Un nodo en un Diagrama de Implantación representa una unidad computacional, normalmente una pieza hardware.
- Las conexiones entre nodos representan las vías de comunicación.

# Ejemplo



# Herramientas

Existen múltiples herramientas para trabajar con UML.

Plugins para el entorno Eclipse:

EclipseUML Studio: <http://www.omondo.com>



Magic Draw UML: <http://www.magicdraw.com>



Slime UML: <http://www.slimeuml.de>



Rational Software Modeler:



<http://www.ibm.com/software/awdtools/modeler/swmodeler>

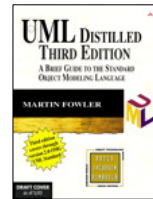


# Bibliografía



## UML Distilled (3<sup>rd</sup> edition).

A brief guide to the standard Object Modeling Language.  
Martin Fowler with Kendall Scott.  
Addison Wesley



## Fundamentals of Object-Oriented Design in UML

Meiler Page-Jones  
Addison Wesley



## Learning UML

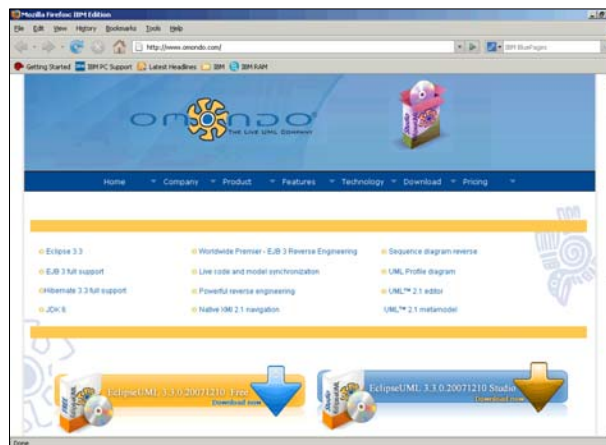
Sinan Si Alhir  
O'Reilly



# Apéndice A: EclipseUML



Descargar EclipseUML 3.3 Free de la web:



<http://www.omondo.com>

## Apéndice A: EclipseUML

- Asegurarse que tenemos Eclipse R3.3.x instalado y funcionando correctamente.
- Ejecutar el fichero JAR con el siguiente comando:  


```
java -jar eclipseUML_E330_2007_freeEdition_3.3.0.v20071210.jar
```

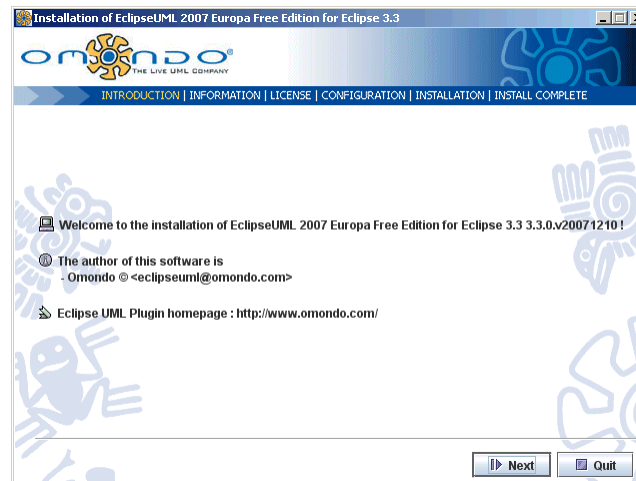
## Apéndice A: EclipseUML

- Seleccionar el idioma de la instalación.




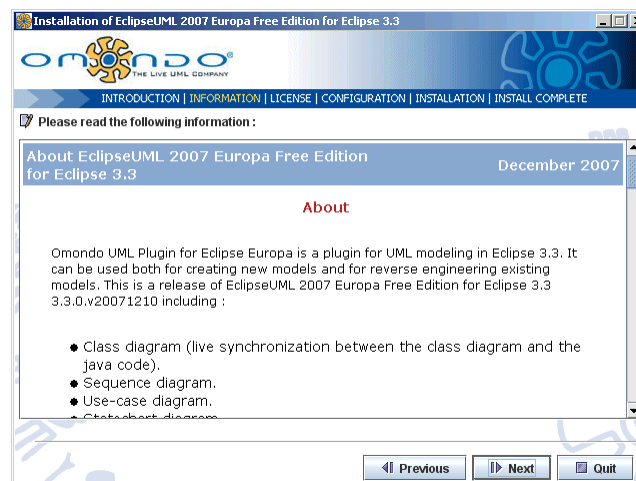
# Apéndice A: EclipseUML

 Pulsar “Siguiente” en la pantalla de introducción.



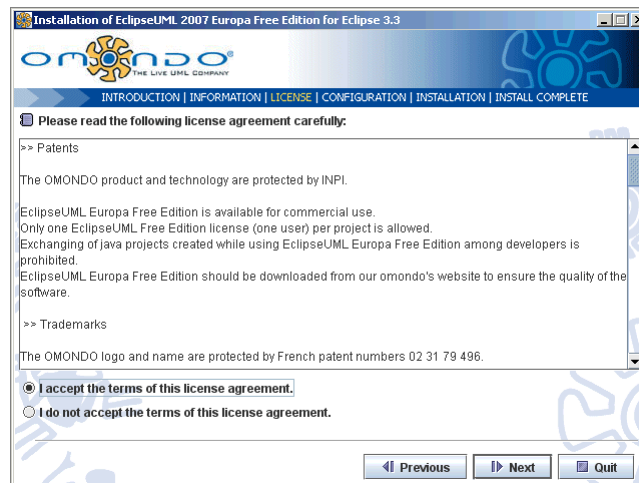
# Apéndice A: EclipseUML

 Pulsar “Siguiente” en la pantalla de información.



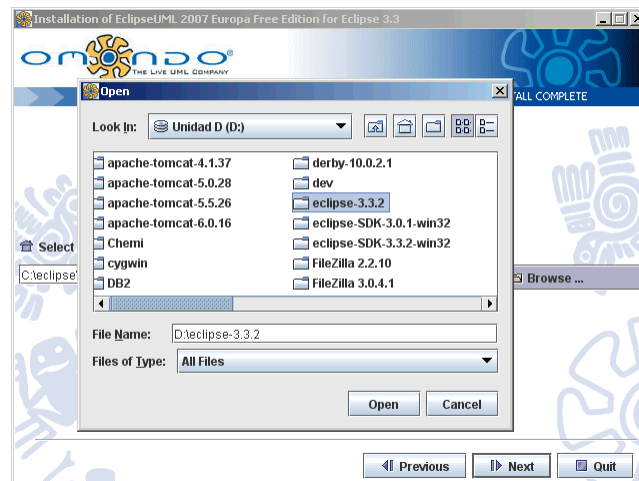
# Apéndice A: EclipseUML

- Aceptar los términos de la licencia y pulsar “Siguiente” en la pantalla de licencia.




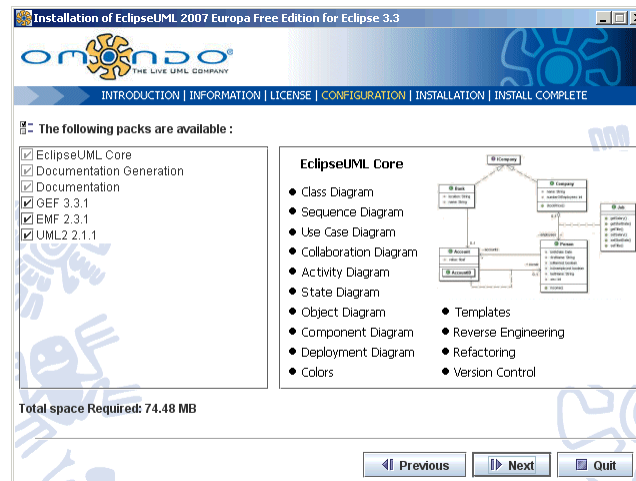
# Apéndice A: EclipseUML

- Seleccionar la ubicación donde se encuentra la instalación de Eclipse R3.3.x y pulsar “Siguiente”.




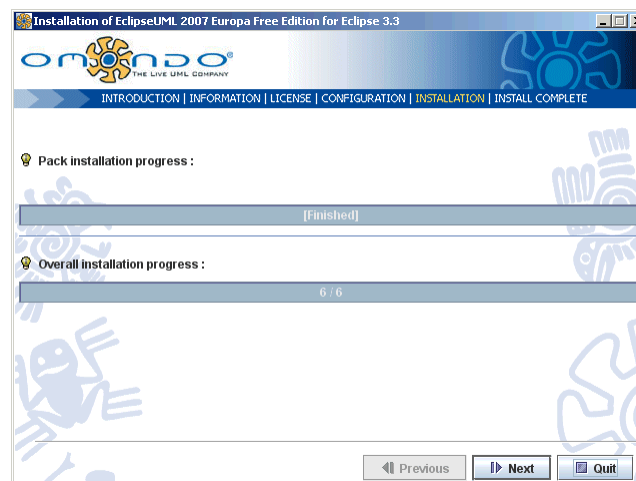
# Apéndice A: EclipseUML

 Pulsar “Siguiente” en la pantalla de configuración.



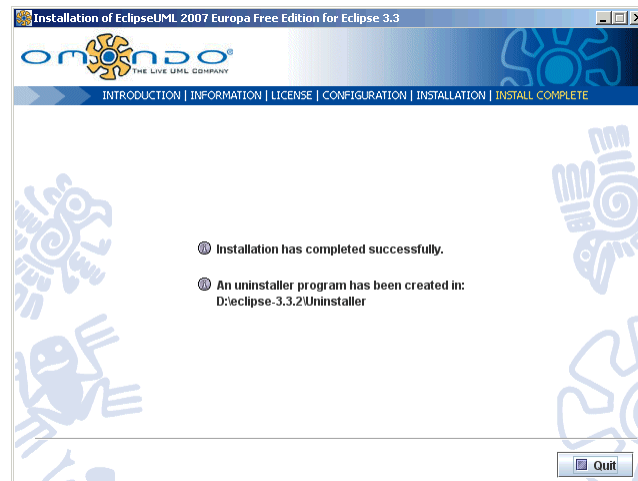
# Apéndice A: EclipseUML

 Pulsar “Siguiente” en la pantalla de instalación una vez haya terminado.



# Apéndice A: EclipseUML

- Pulsar “Salir” en la última pantalla.



# Apéndice A: EclipseUML

- Una vez haya rearrancado ya podremos desarrollar diagramas UML.

