Resumen diagramas UML

El lenguaje UML es un estándar OMG diseñado para visualizar, especificar, construir y documentar software orientado a objetos.

Un modelo es una simplificación de la realidad.

El modelado es esencial en la construcción de software para…

- Comunicar la estructura de un sistema complejo
- Especificar el comportamiento deseado del sistema
- **Comprender mejor** lo que estamos construyendo
- Descubrir oportunidades de simplificación y reutilización

Un modelo proporciona "los planos" de un sistema y puede ser más o menos detallado, en función de los elementos que sean relevantes en cada momento.

El modelo ha de capturar “lo esencial”.

Todo sistema puede describirse desde distintos puntos de vista:

- **Modelos estructurales (organización del sistema)**
- **Modelos de comportamiento (dinámica del sistema)**

Inconvenientes de UML

- Falta de integración con otras técnicas
  (p.ej. diseño de interfaces de usuario)
- UML es excesivamente complejo (y no está del todo libre de ambigüedades): “el 80% de los problemas puede modelarse usando alrededor del 20% de UML”

UML estandariza 9 tipos de diagramas para representar gráficamente un sistema desde distintos puntos de vista.

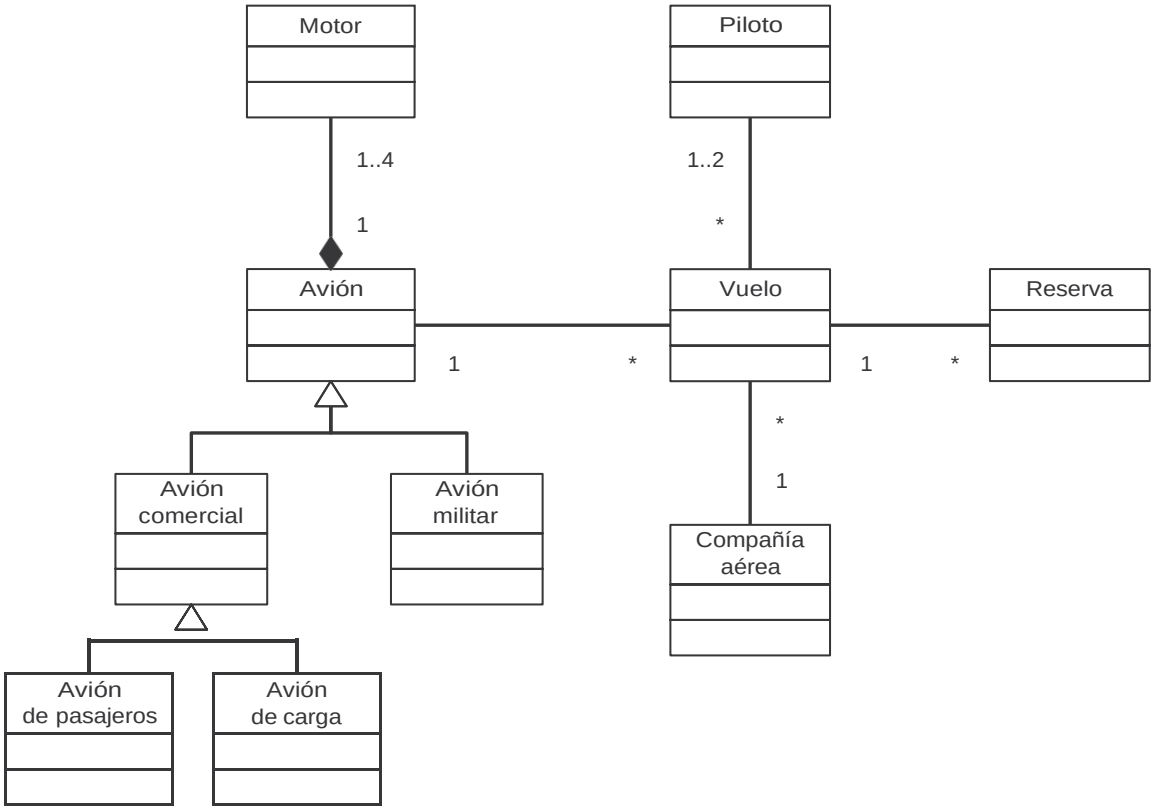
Los que más se suelen usar son los que ya hemos visto en clase:

- Diagrama de Clases (modelo estructural)
- Diagrama de Casos de uso (modelo de comportamiento)

diagramas UML para representar aspectos dinámicos del sistema

1) Diagramas de clases (estructural)

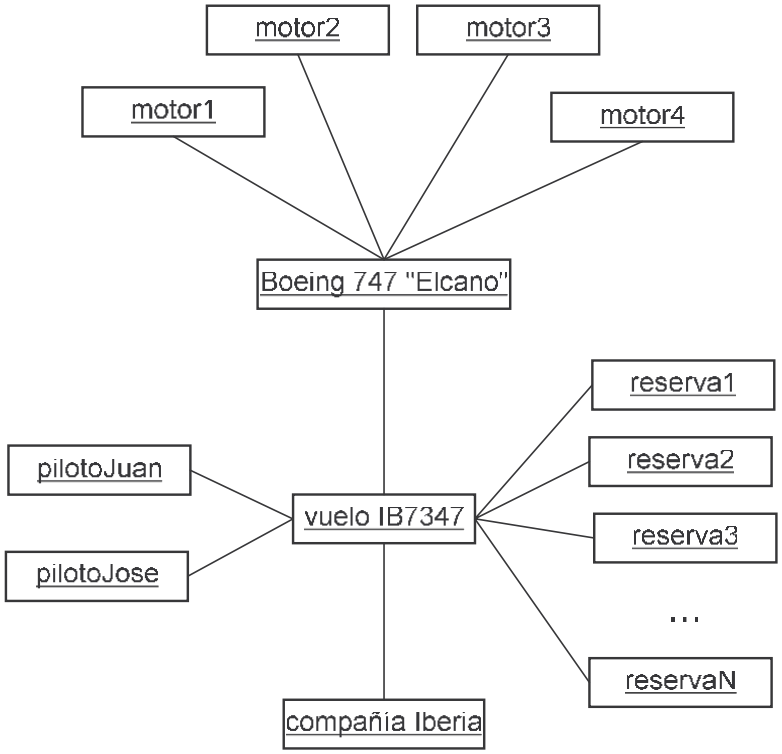
Muestran un conjunto de clases y sus relaciones



Los diagramas de clases proporcionan una **perspectiva estática** del sistema(representan su diseño estructural.

2) Diagramas de objetos (estructural)

Muestran un conjunto de objetos y sus relaciones
(una situación concreta en un momento determinado).



Los diagramas de objetos representan instantáneas de instanciasde los elementos que aparecen en los diagramas de clases

Un diagrama de objetos **expresa la parte estática** de una interacción.

Para ver los aspectos dinámicos de la interaccións utilizan los diagramas de interacción (diagramas de secuencia y diagramas de comunicación/colaboración)

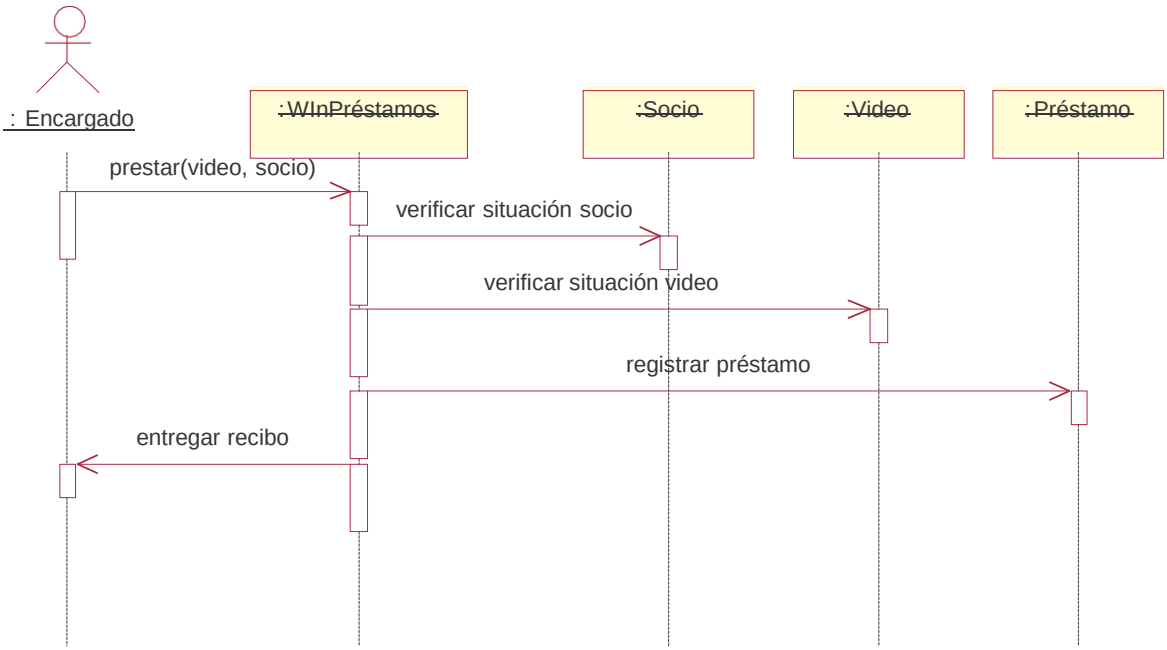
NOTA:
Los identificadores subrayados indican que se trata de objetos.

3) Diagramas de interacción (comportamiento)

Muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos.

Diagramas de secuencia

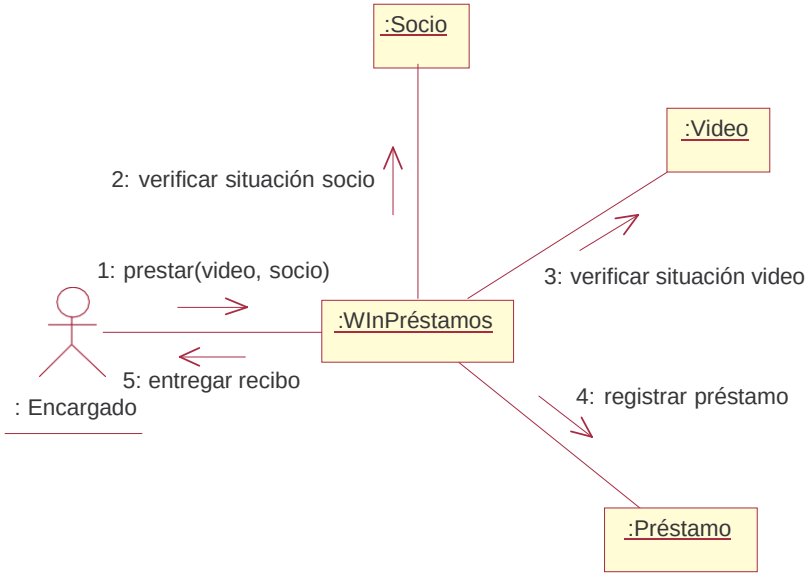
Resaltan la ordenación temporal de los mensajes que se intercambian.



4) Diagramas de comunicación (UML 2.0) (comportamiento)

(llamados Diagramas de colaboración en UML 1.x)

Resaltan la organización estructural de los objetos que intercambian mensajes.



Los diagramas de secuencia y de comunicación son isomorfos:

- Un diagrama de secuencia se puede transformar mecánicamente en un diagrama de comunicación.
- Un diagrama de comunicación se puede transformar automáticamente en un diagrama de secuencia.

Diagramas de secuencia

Muestran la secuencia de mensajes entre objetos durante un escenario concreto (paso de mensajes).

- En la parte superior aparecen los objetos que intervienen.
- La dimensión temporal se indica verticalmente (el tiempo transcurre hacia abajo).
- Las líneas verticales indican el período de vida de cada objeto.
- El paso de mensajes se indica con flechas horizontales u oblicuas (cando existe demora entre el envío y la atención del mensaje).
- La realización de una acción se indica con rectángulos sobre las líneas de actividad del objeto que realiza la acción.

Diagramas de comunicación/colaboración

La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto de los demás

- La perspectiva estática del sistema viene dada por las relaciones existentes entre los objetos (igual que en un diagrama de objetos).
- La vista dinámica de la interacción viene indicada por el envío de mensajes a través de los enlaces existentes entre los objetos.

NOTA: Los mensajes se numeran para ilustrar el orden en que se emiten.

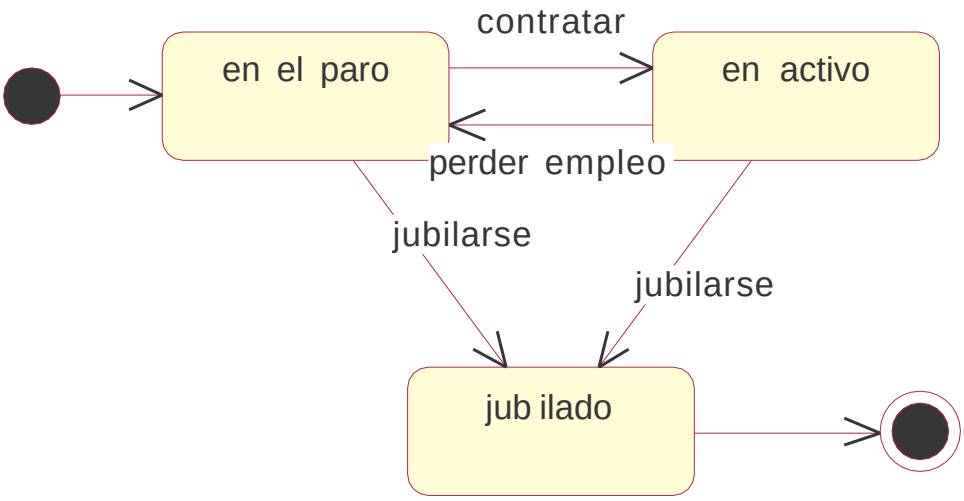
5) *Diagramas de casos de uso (comportamiento)*

(actores y casos de uso del sistema)

Los diagramas de uso se suelen utilizar en el modelado del sistema desde el punto de vista de sus usuarios para representar las acciones que realiza cada tipo de usuario.

6) *Diagramas de estado s(comportamiento)*

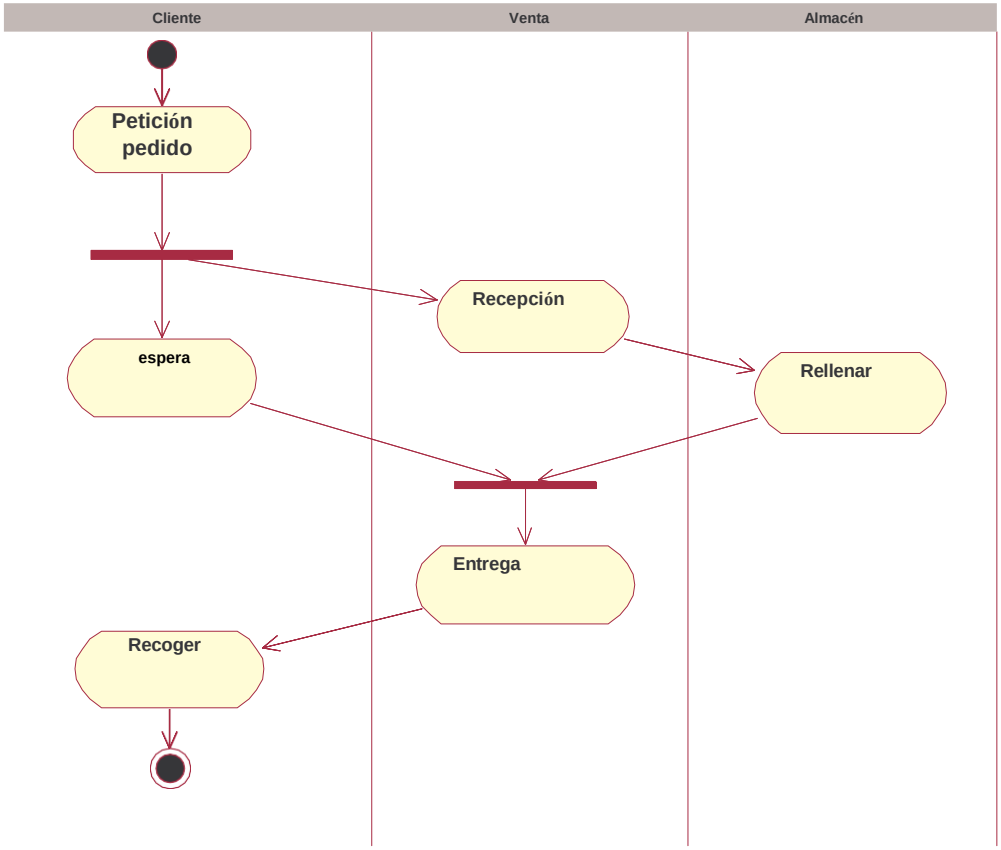
(estados y transiciones entre estados),



Los diagramas de estados son especialmente importantes para describir el comportamiento de un sistema reactivo (cuyo comportamiento está dirigido por eventos).

7) *Diagramas de actividades (comportamiento)*

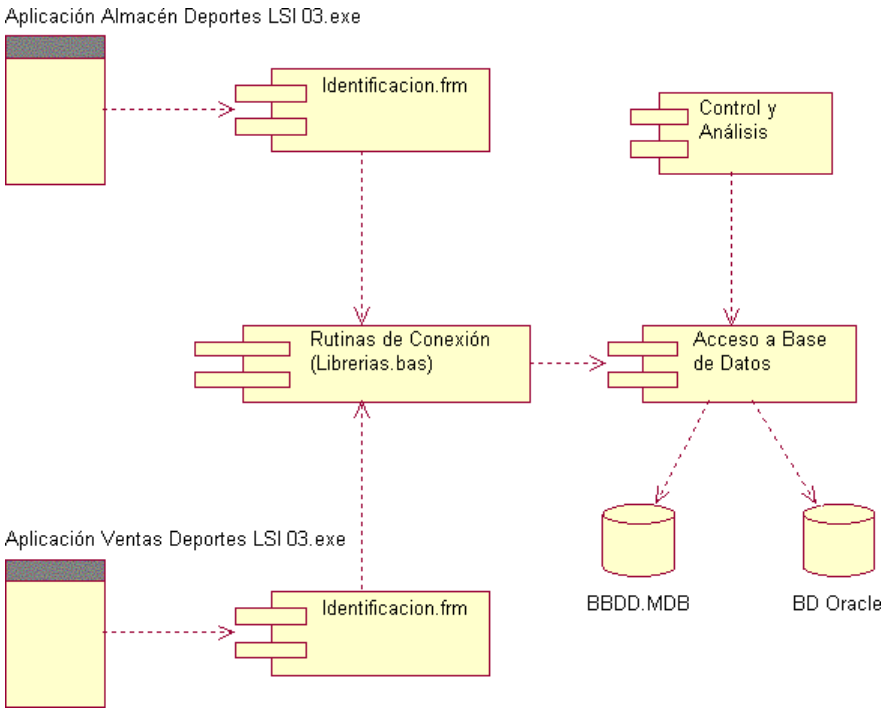
(flujo de control en el sistema)



Los diagramas de actividades muestran el orden en el que se van realizando tareas dentro de un sistema (el flujo de control de las actividades).

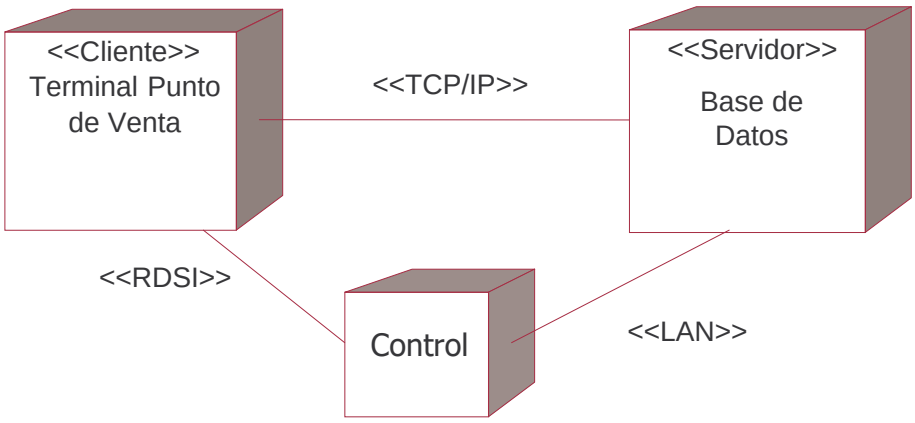
8) *Diagramas de componentes (estructural)*

(componentes y dependencias entre ellos)
Organización lógica de la implementación de un sistema



9) *Diagramas de despliegue (estructural)*

(nodos de procesamiento y componentes)
Configuración del sistema en tiempo de ejecución



Referencias

Páginas web

http://www.uml.org/
Página oficial de UML, uno de los estándares promovidos por el OMG.

http://www.cetus-links.org/oo_uml.html
Colección de enlaces relacionados con UML.

http://www.agilemodeling.com/essays/umlDiagrams.htm
Información práctica acerca de todos los diagramas UML 2

http://www.ootips.org/
Ideas clave en programación orientada a objetos.

Libros

Martin Fowler: *"UML Distilled:*
A Brief Guide to the Standard Object Modeling Language"
3rd edition. Addison-Wesley, 2004. ISBN 0321193687

Grady Booch et al.:
"Object-Oriented Analysis and Design with Applications"
3rd edition. Addison-Wesley, 2004. ISBN 020189551X

Craig Larman: *"Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process"*
2nd edition. Prentice-Hall, 2001. ISBN 0130925691

Robert C. Martin:
"Agile Software Development: Principles, Patterns, and Practices"
Prentice-Hall, 2003. ISBN 0135974445

…