

Notación UML

Georg Lehner

22 de octubre de 2002

UML Unified Modeling Language

El proceso de percepción y desarrollo de aplicaciones informáticas requiere metodologías sistemáticas cuando la escala de la aplicación requiere la colaboración de un equipo de programadores. Estos son llamadas técnicas de ingeniería de Software.

Las fases de desarrollo de tales proyectos se determinan comúnmente en

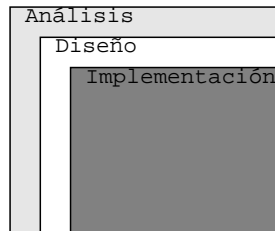


Figura 1: Ciclo de fases de proyectos de Software

1. Análisis: comprensión del dominio del problema a resolver
2. Diseño: Especificación de una mecanismo técnico para resolver el problema
3. Implementación: Hechura del mecanismo divisado

La figura 1 insinúa, que estas fases no son aisladas: durante cada fase subsiguiente sigue pendiente completar / recurrir a fases anteriores.

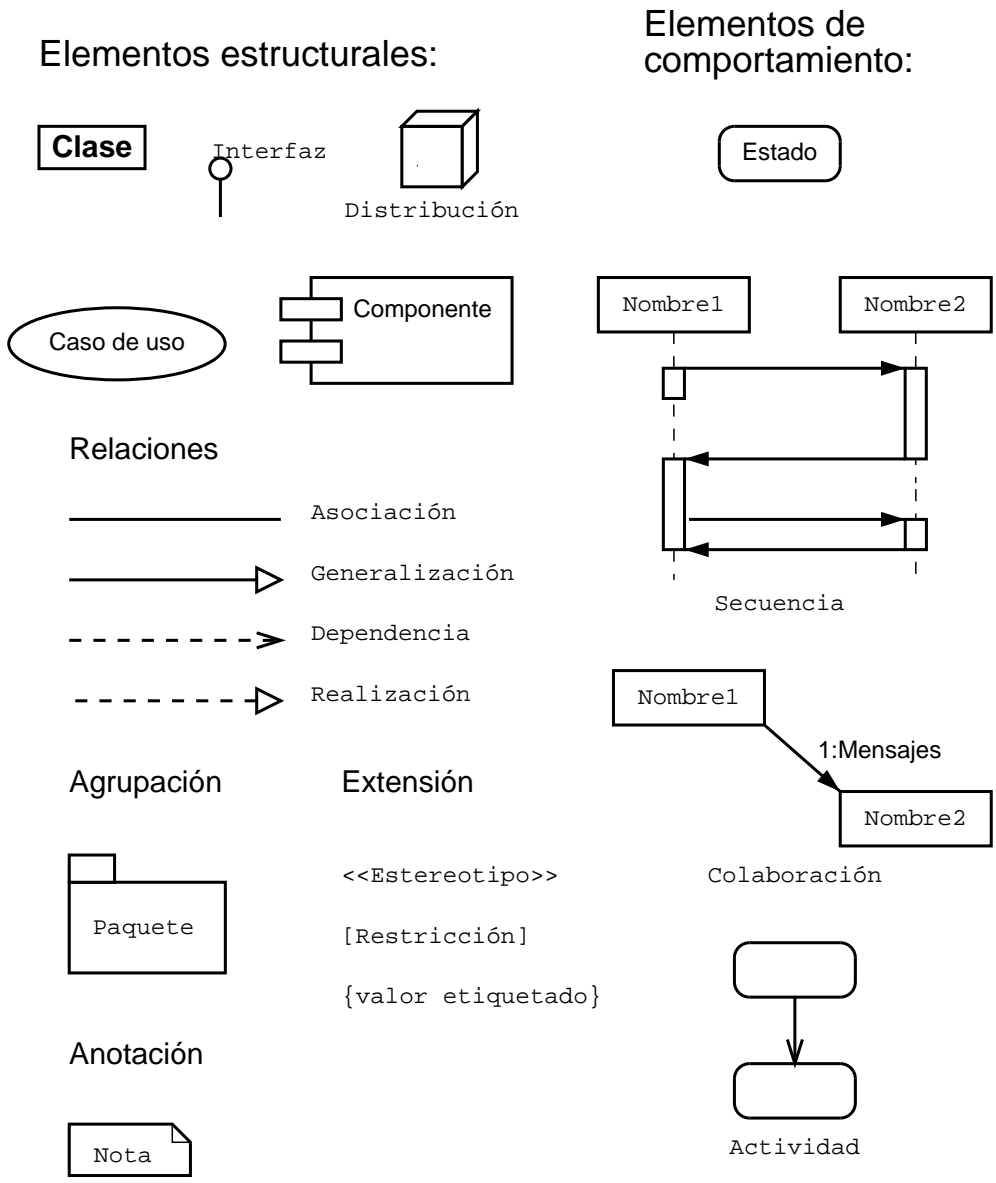


Figura 2: Elementos gráficos UML

Variedades de metodología han sido creados, para sistematizar el proceder durante proyectos de desarrollo de software . UML es uno de ellos que intenta resolver:

1. Modelación → Unidad dos: Modelación
2. Comunicación

Del análisis de un proyecto de Software se desarrolla un *Modelo* del problema a resolver, que es presentado en el *Lenguaje Gráfico UML*.

La normalización de este lenguaje permite *comunicar* el modelo entre diferentes participantes del proyecto sin pérdida de contenido.

Un lenguaje gráfico consiste en un juego de símbolos con significado predefinido y reglas para su “correcta” ubicación en el plano bidimensional.

La figura 2 muestra los elementos sintácticos básicos del lenguaje UML.

Por su naturaleza se distinguen tres clases de símbolos:

1. Elementos estructurales:
Delinean los objetos reales o virtuales que se utilizan para crear un modelo del sistema real.
2. Elementos de comportamiento:
Delinean las transiciones o variaciones experimentado por elementos estructurales o por relaciones entre estos.
3. Conectores:
Delinean relaciones, sus calificaciones y condicionantes entre los elementos estructurales y de comportamiento.

Elementos estructurales

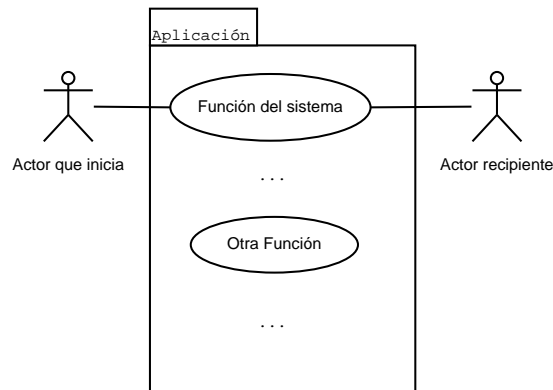


Figura 3: Caso de uso, esquema principal

La figura 3 muestra esquemáticamente un *caso de uso*. En el análisis de un dominio de problema se pregunta, de que forma será utilizado el sistema a realizar (black box) – es decir el punto de vista de todos actores externos. Actores son cualquier sistema externo, incluso seres humanos.

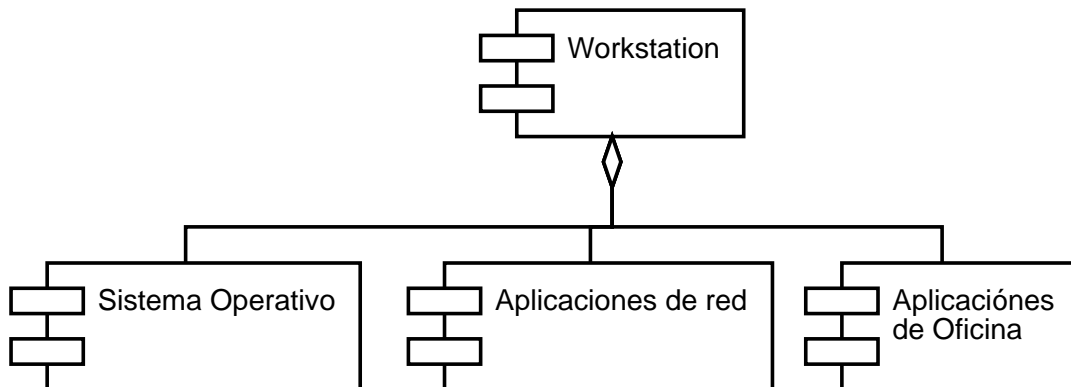


Figura 4: Diagrama de componentes, Estación de trabajo

Los diagramas de *componentes*, vea fig. 4, muestran la interrelación entre módulos o subsistemas de Software.

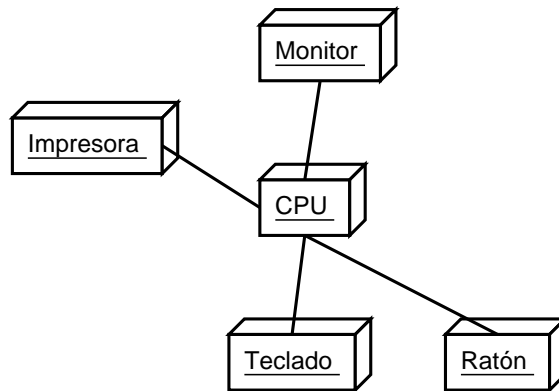


Figura 5: Diagrama de distribución, Estación de trabajo

Diagramas de *distribución*, como el de figura 5 muestran interconexión de componentes del Hardware. Su utilidad (entre otras) consiste en la localización de los portadores físicos de los componentes de Software.

Diagramas Clase e Interfaz son elementos estructurales que se discutirán en la segunda unidad.

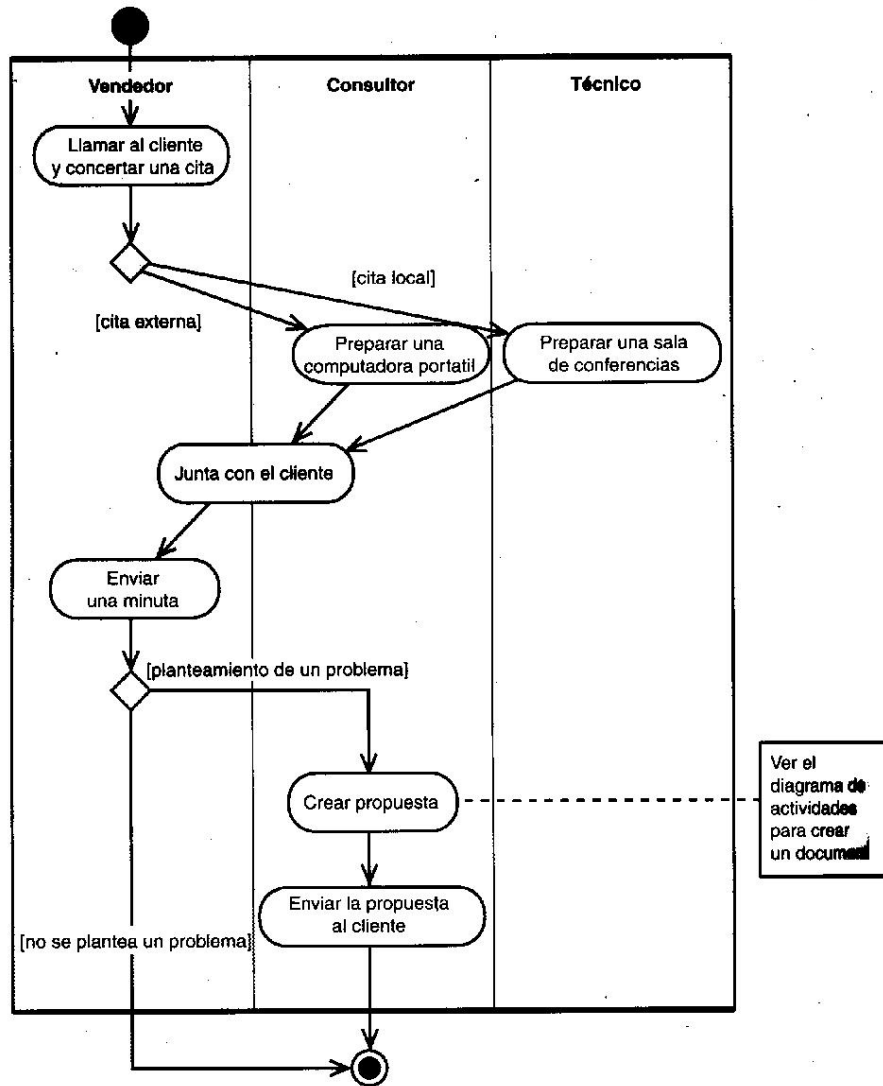


Figura 6: Diagrama de actividades, Proceso de venta

Elementos de comportamiento

El diagrama de *actividades*, vea fig. 6, corresponde al tradicional flujograma. La figura está tomada del libro aprendiendo UML en 24 horas, e introduce el concepto de *marcos de responsabilidad*.

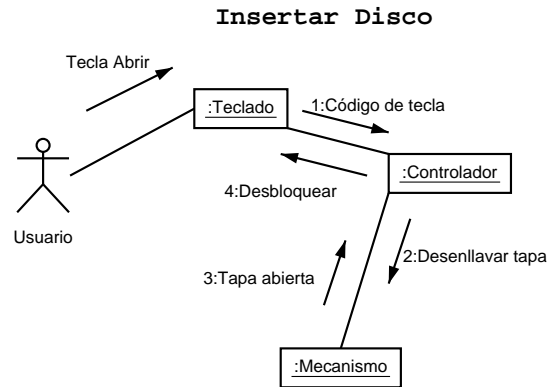


Figura 7: Diagrama de colaboración, Tocador de CD

Los diagramas de colaboraciones (fig. 7) y de secuencias (fig. 8) son equivalentes a través de su relación con los objetos que representan.

La *colaboración* muestra interacción entre objetos. La transición de un estado al siguiente se induce por *mensajes* representado con flechas numeradas secuencialmente. Vea fig.7.

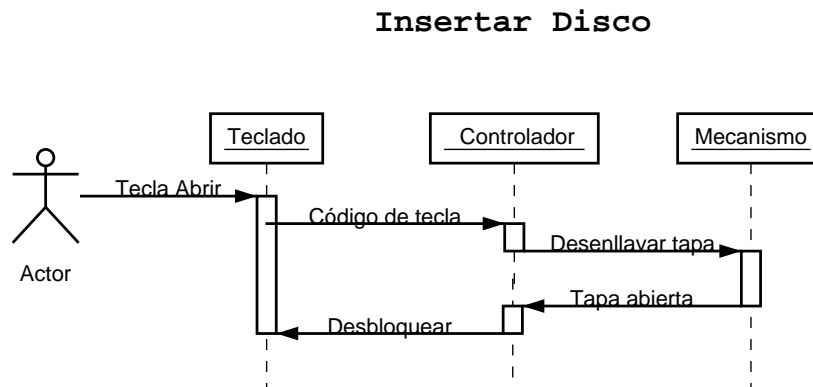


Figura 8: Diagrama de secuencias, Tocador CD

El diagrama correspondiente se mira en la figura 8. La *secuencia* de las transiciones de estado de los diferentes objetos y su interacción están delineados al *paso del tiempo*, que transcurre desde arriba hacia abajo en el diagrama.

Tocador de CD

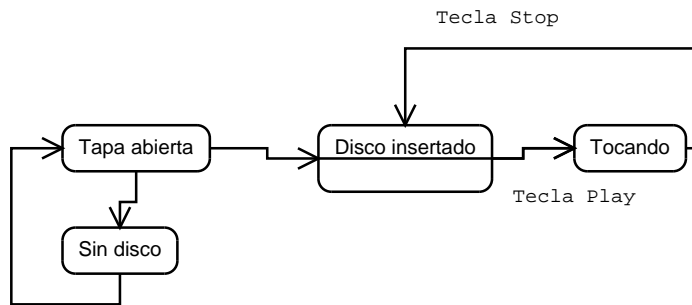


Figura 9: Diagrama de estado, Tocador CD

Los diagramas de *estados* son equivalentes a los conocidos de la teoría de Bool y de autómatas, aunque extienden su semántica. Estados de objetos son rectángulos redondeados, son interconectados con flechas de *mensajes*, indicando la transición entre ellos, vea fig. 9.

Conectores

Los elementos conforman la base del contenido transmitido, pero las relaciones entre los objetos son esenciales para comprender lo expresado por un diagrama.

La *asociación* agrupa elementos al mismo nivel, como en el ejemplo de la figura 5. La asociación se denota con una línea *solida*.

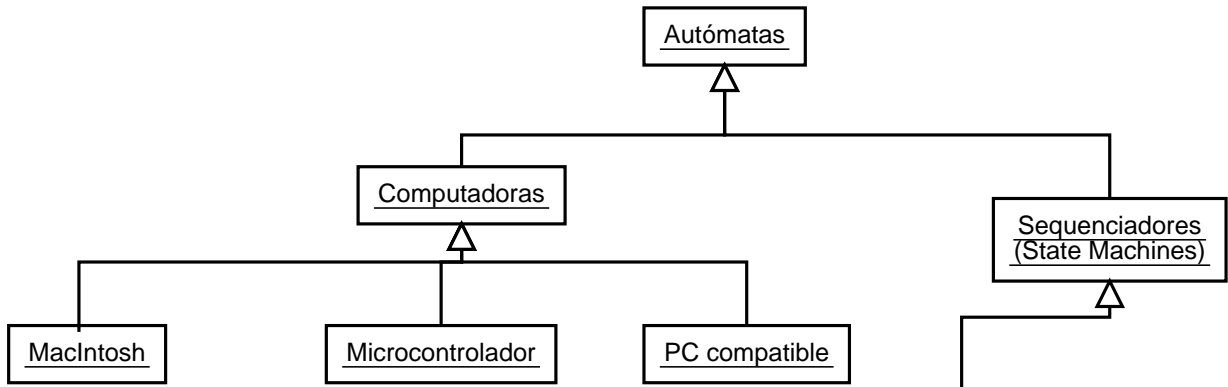


Figura 10: Generalización, Jerarquía de autómatas

La *generalización* crea jerarquías. Unos objetos se subsumen bajo una clase o categoría, mediante una flecha con *línea sólida y punta triangular vacía*, vea fig. 10.

Tocador de CD

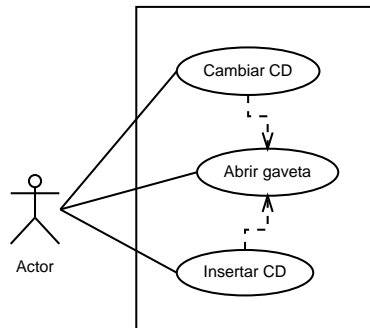


Figura 11: Dependencias, Inclusión de casos de uso

La *dependencia* muestra, que una realización de una función, secuencia o elemento depende de otro elemento. Se dibuja con una *línea discontinua con punta de flecha*. Fig. 11 muestra el uso de la dependencia para “importar” o incluir un caso de uso reutilizable.