

¿Qué es lo más elemental que debo saber sobre UML?

Linda María Fernández Olvera
UPIICSA-IPN
Ingeniería de Diseño
lfernandezol1500@alumno.ipn.mx

Abstract – Este ensayo explicará los componentes más usados del Lenguaje UML, su utilidad, el objetivo de cada diagrama y algunas características importantes que son importantes de saber para el uso correcto del lenguaje al momento de modelar y diseñar sistemas.

Índice de Términos – UML, diseño de sistemas, requerimientos, cliente,

I. INTRODUCCION

UML (Lenguaje de Modelado Unificado) es un lenguaje estándar para escribir diseños de software, se puede usar para visualizar, especificar, construir y documentar los sistemas de software. Permite a los desarrolladores de sistemas generar diseños, plasmarlos de manera gráfica y comunicar las ideas de manera sencilla.

El desarrollo del sistema solía ser una construcción de ideas al azar, los analistas evaluaban los requerimientos del cliente, los analizaban y entregaban al desarrollador esperando que al final resultase lo que el cliente deseaba. Hoy en día es necesario contar con un plan para el desarrollo de sistemas, ya que tanto el cliente tiene que comprender qué es lo que se hará con el sistema, así como debe tener la capacidad de señalar los cambios que necesite para satisfacer sus necesidades. Debido a que el trabajo de desarrollo es un esfuerzo orientado a equipos, por lo que cada uno de los miembros tiene que saber qué lugar toma su trabajo en la solución final.

Según la ley de Moore, conforme avance la tecnología también lo hará la complejidad de los sistemas informáticos que se desarrollan, por lo tanto es necesario emplear herramientas que nos ayuden a organizar el proceso de diseño, de tal forma que todos los integrantes del equipo de desarrollo lo comprendan.

El UML es la herramienta que permite la organización adecuada para el diseño de sistemas complejos, además de que reduce el periodo de desarrollo.

II. DESARROLLO

UML fue creado por Grady Booch, James Rumbaugh e Ivar Jacobson cuando trabajaban en empresas distintas a principios de los 90s cada uno diseñó una metodología propia para el análisis y diseño orientado a objetos, con el tiempo colaboraron entre sí y decidieron desarrollar UML en conjunto.

Conforme UML fue incursionando en la industria del software los corporativos se dieron cuenta de que esta metodología cumplía con sus propósitos, en 1997 se produjo la versión 1.0 del UML y se propuso como un lenguaje modelado estándar.

A. Diagramas UML

UML se compone por diversos elementos gráficos que se combinan para crear diagramas.

El objetivo de los diagramas es presentar diversas perspectivas de un sistema y conformar un modelo. Es importante considerar que el modelo UML describe lo que hará el sistema pero no dice como implementarlo.

A continuación, describiré los tipos de diagramas más comunes del lenguaje.

1. Diagrama de clases

Una clase es una categoría o grupo de cosas que tienen atributos y acciones. En la notación de UML una clase se compone de un rectángulo dividido en 3 áreas, la primera contiene el nombre de la clase, el área central los atributos y finalmente el área inferior las acciones. En la Figura 1 se muestra el símbolo de una clase. Las relaciones de las clases se expresan como líneas que conectan los rectángulos de clase.

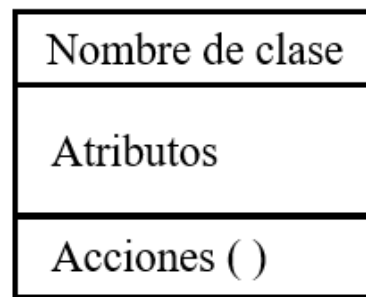


Figura 1. Símbolo UML de una clase.

¿Por qué es importante pensar en clases, atributos y acciones? El software moderno simula al mundo, después de décadas de desarrollo se infiere que es más sencillo desarrollar aplicaciones representando clases de objetos reales. También permiten al analista de requerimientos dirigirse a los clientes en su propia terminología y permiten que ellos comuniquen detalles de sus necesidades.

2. Diagrama de objetos

El objeto es una instancia de una clase, una entidad con valores específicos definidos de los atributos y acciones. En este diagrama el símbolo del objeto es un rectángulo, similar al de una clase, pero el nombre está subrayado. El nombre de la instancia específica se encuentra a la izquierda de los dos puntos (:) y el nombre de la clase a la derecha como se muestra en la Figura 2.

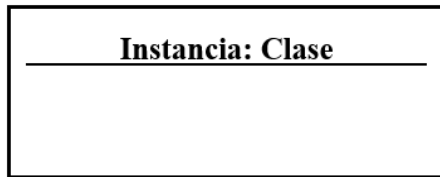


Figura 2. Símbolo UML del objeto

3. Diagrama de casos de uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Esta valiosa herramienta es una técnica de aciertos y errores para definir los requerimientos del sistema desde la visión del usuario. Es muy útil si la finalidad es crear un sistema que pueda ser empleado por usuarios en general.

En la figura 3 se muestra como el elemento que corresponde al usuario se le conoce como actor. La elipse representa el caso de uso. El actor puede ser una persona u otro sistema.

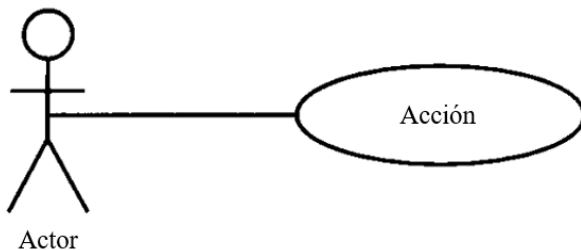


Figura 3. Símbolo UML de diagrama de casos de uso

4. Diagrama de estados

Los objetos todo el tiempo se encuentran en estados particulares, en UML el símbolo de la parte superior representa el estado inicial y el de la parte inferior el estado final como se muestra en la figura 4. Los estados parten de uno inicial y se conducen por flechas que apuntan el siguiente estado, de esta manera muestran el orden.

Estos estados representan el lugar, la manera o las acciones que está realizando el objeto. Por ejemplo, una persona puede ser recién nacida, infante, adolescente, joven o adulta. Los alimentos pueden estar refrigerados, cocinados, crudos, vencidos. Un reproductor puede estar en pausa, avanzar de canción, detenerse o reproducir música.

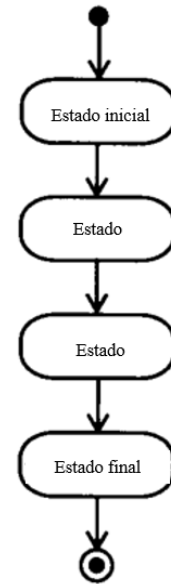


Figura 4. Diagrama de estados UML.

5. Diagrama de secuencias

Los diagramas anteriores como el de clases u objetos representan información estática. Sin embargo, en un sistema funcional los objetos interactúan entre sí, y las interacciones suceden con el tiempo.

En el diagrama de secuencias UML se muestra la mecánica de la interacción con relación a los tiempos.

Entre los componentes de un objeto se presentan estados y acciones de las clases. El diagrama como se muestra en la Figura 5, captura las interacciones que se realizan a través del tiempo.

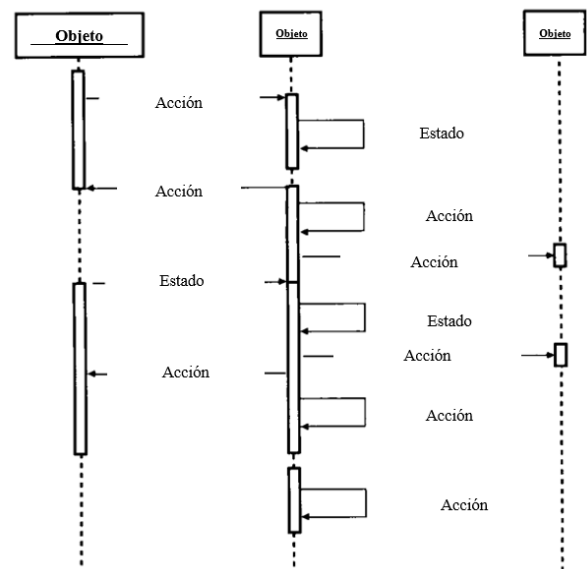


Figura 5. Diagrama de secuencias UML

6. Diagrama de actividades

Las actividades que ocurren dentro de los casos de uso o dentro del comportamiento o acciones de un objeto normalmente se dan en secuencia. La Figura 6 muestra como el diagrama de actividades UML representa los pasos de las secuencias.

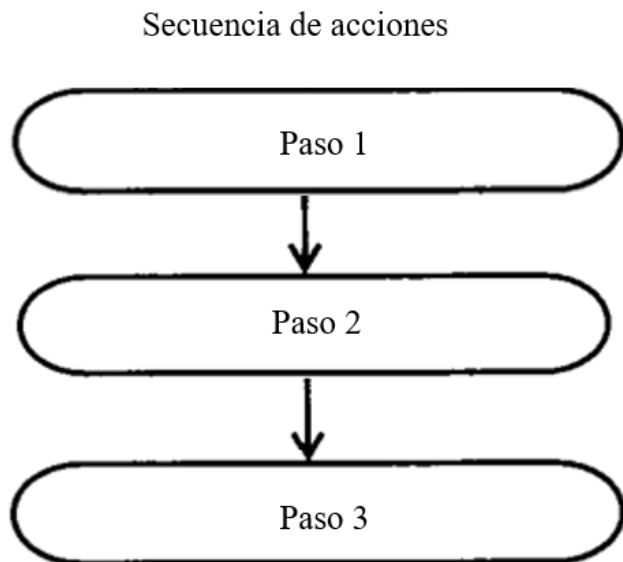


Figura 6. Diagrama de actividades UML.

7. Diagrama de colaboraciones

Los elementos de un sistema trabajan en conjunto para cumplir con los objetivos del sistema. En la Figura 7 se muestra como las clases se relacionan, la acción con la que se relacionan y el orden o secuencia de las acciones.

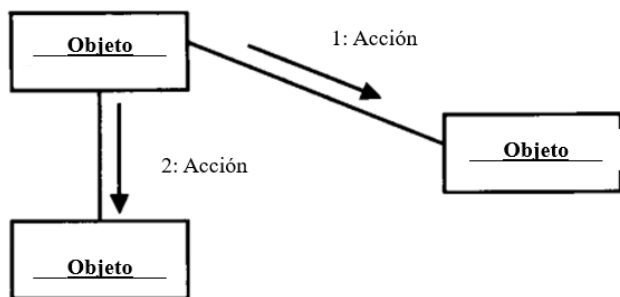


Figura 7. Diagrama de colaboraciones UML

8. Diagrama de componentes

El desarrollo de software se realiza mediante componentes, que representan una parte modular de un sistema. El comportamiento se define en términos de interfaces obligatorias y proporcionadas.

En la Figura 8 se muestra como se ilustra el componente o módulo de un sistema.

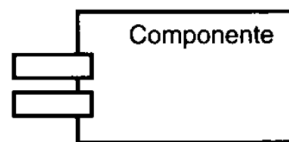


Figura 8. Diagrama de componentes UML.

9. Diagrama de distribución

Este diagrama muestra la arquitectura física de un sistema informático. Puede representar los equipos y dispositivos, sus interconexiones y el software que se encontrará en cada máquina. Cada computadora se representa como un cubo y las interacciones son líneas que conectan el cubo.

En la Figura 9 se muestra una conexión entre 3 computadoras y sus características.

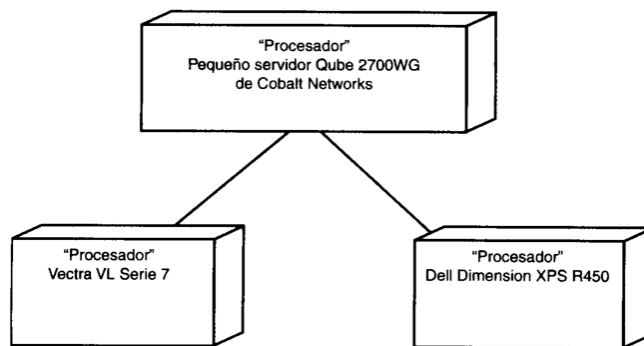


Figura 9. Diagrama de distribución UML de tres computadoras.

B. Otras características

1. Paquetes

Los paquetes son agrupaciones de ciertas clases, componentes o elementos que se necesitan organizar en grupos, ya sea porque son parte un subsistema o funcionan en conjunto. En la figura 10 se muestra un paquete que agrupo 3 clases.

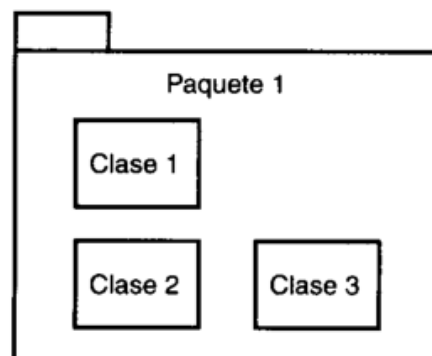


Figura 10. Paquete UML

2. Notas

Un rectángulo con la esquina doblada que representa una explicación para una parte de diagrama. Dentro del rectángulo se coloca la explicación como se muestra en la Figura 11.

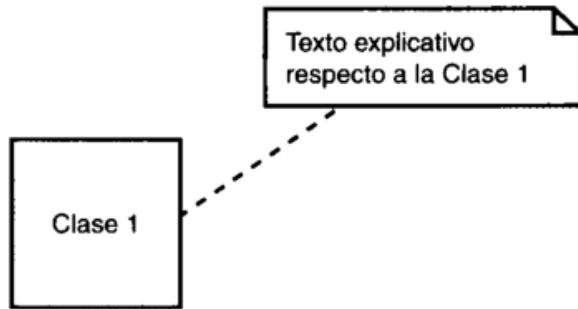


Figura 11. Nota

3. Estereotipos

Los estereotipos le permiten tomar elementos propios del UML y convertirlos en otros. Se representa como un nombre entre dos pares de paréntesis angulares.

En la Figura 12, se muestra una interfaz que es una clase que realiza operaciones, pero no tiene atributos. En lugar de inventar un nuevo elemento se podrá utilizar el símbolo de la clase pero ahora con <<Interfaz>> situada en el nombre de la clase.

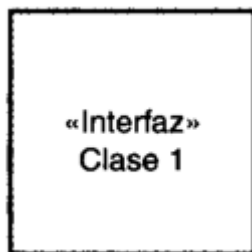


Figura 12. Estereotipo de Interfaz

III. CONCLUSIÓN

Los diagramas UML permiten examinar un sistema desde distintos tipos de vista. Un modelo UML no necesita todos los tipos de diagramas, solo los necesarios para cubrir las necesidades del cliente. Este lenguaje permite incorporar todas las perspectivas y organizarlas para facilitar el entendimiento para todos los integrantes del equipo de desarrollo.

Es importante recalcar que UML indica que hará el sistema, pero no cómo lo hará. Solamente diseña el funcionamiento y relaciones que tendrá entre los objetos.

REFERENCES

- [1] Schmuller Joseph, "Aprendiendo UML en 24 horas" Prentice Hall pp. 5-16.