

TEMA V

UML (MODELIZACIÓN)

1. INTRODUCCIÓN

La orientación a objetos supuso un cambio en la forma de trabajar frente al enfoque estructurado. Con este nuevo paradigma (modelo o patrón) el proceso se centra en simular los elementos de la realidad asociada al problema de la forma más cercana posible. La abstracción que permite representar estos elementos se denomina objeto, y tiene las siguientes características:

Está formado por un conjunto de atributos, que son los datos que le caracterizan y un conjunto de operaciones que definen su comportamiento. Las operaciones asociadas a un objeto actúan sobre sus atributos para modificar su estado. Cuando se indica a un objeto que ejecute una operación determinada se dice que se le pasa un mensaje.

Las aplicaciones orientadas a objetos están formadas por un conjunto de objetos que interaccionan enviándose mensajes para producir resultados. Los objetos similares se abstraen en clases, se dice que un objeto es una instancia de una clase.

Cuando se ejecuta un programa orientado a objetos ocurren tres sucesos:

1. Primero, los objetos se crean a medida que se necesitan.
2. Segundo. Los mensajes se mueven de un objeto a otro (o del usuario a un objeto) a medida que el programa procesa información o responde a la entrada del usuario.
3. Tercero, cuando los objetos ya no se necesitan, se borran y se libera la memoria.

2. UML (Unified Modeling Language o Lenguaje Unificado de Modelado)

Es un conjunto de herramientas que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Se ha convertido en el estándar de la

industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

UML permite a los desarrolladores visualizar el producto de su trabajo en esquemas o diagramas estandarizados denominados modelos (representación gráfica de una realidad), sirve para organizar y comunicar de forma clara los elementos que involucran un todo.

Es como un esquema teórico de un sistema o de una realidad compleja que se elabora para facilitar su comprensión y el estudio de su comportamiento.

Pretende proporcionar una forma estándar de representar el diseño de un sistema.

¿Por qué es útil modelar?

Permite utilizar un lenguaje común que facilita la comunicación entre el equipo de desarrollo.

Podemos documentar todos los “objetos” de un proceso de desarrollo: información, requisitos, arquitectura, pruebas, versiones, ...

Hay estructuras que trascienden lo representable en un lenguaje de programación, como las que hacen referencia a la arquitectura del sistema, utilizando estas tecnologías podemos incluso indicar qué módulos de software vamos a desarrollar y sus relaciones, o en qué nodos hardware se ejecutarán cuando trabajamos con sistemas distribuidos.

Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose modelos precisos, no ambiguos y completo

Además, UML puede conectarse a lenguajes de programación mediante ingeniería directa o inversa

Elementos de un diagrama UML

UML define un sistema como una colección de modelos. Los modelos se implementan en diagramas, representación gráfica de una colección de elementos.

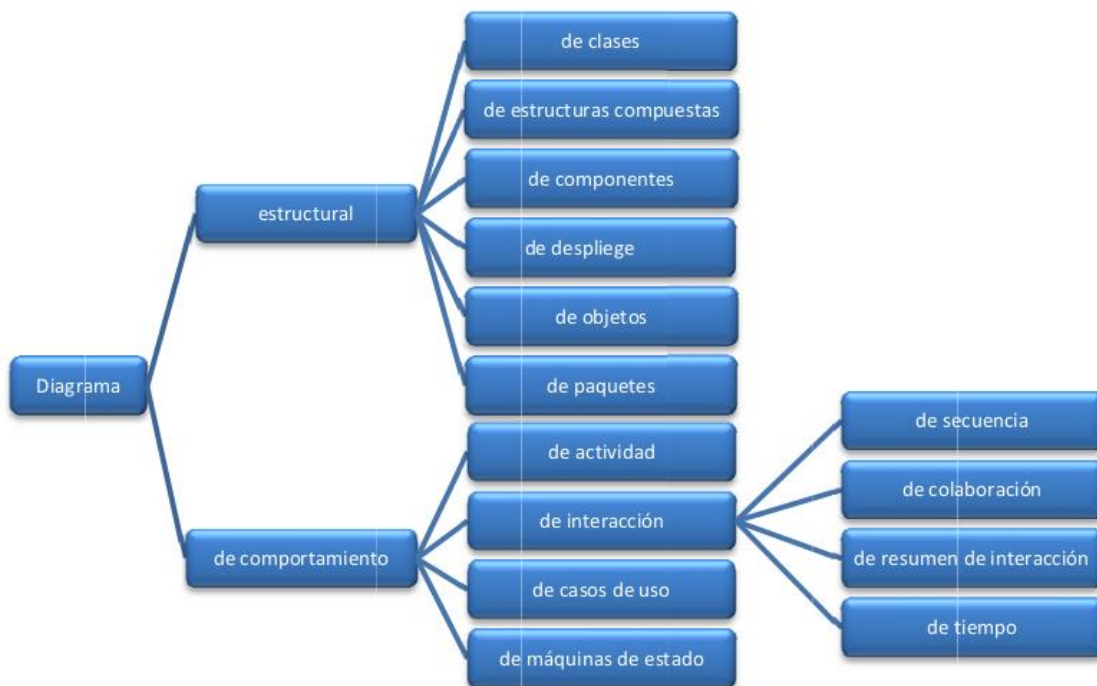
Un diagrama se compone de 4 tipos de elementos:

1. Estructuras: nodos del grafo y definen el tipo de diagrama
2. Relaciones: arcos del grafo que se establecen entre las estructuras
3. Notas: cuadros que nos permiten escribir comentarios que nos ayudarán a comprender el concepto representado

4. Agrupaciones: se utilizan cuándo representamos sistemas grandes para facilitar su desarrollo por bloques

Tipos de diagramas

1. Estructurales: Representan la visión estática del sistema. Especifican clases y objetos y como se distribuyen físicamente en el sistema.
2. De comportamiento: muestran la conducta en tiempo de ejecución del sistema, tanto desde el punto de vista del sistema completo como de las instancias u objetos que lo integran.



No son necesarios todos estos tipos, se puede modelar con unos cuantos de ellos. Veremos los más utilizados: **de casos de uso, de clases y de secuencia**

3. HERRAMIENTAS UML

La herramienta más simple que se puede utilizar para generar diagramas es lápiz y papel. Sin embargo, podemos acceder a herramientas CASE que facilitan el desarrollo de los diagramas UML. Estas herramientas suelen contar con un entorno de ventanas tipo wysiwyg (What You

See Is What You Get), permiten documentar los diagramas e integrarse con otros entornos de desarrollo incluyendo la generación automática de código y procedimientos de ingeniería inversa.