

Tipos de diagramas UML

¿Qué es UML?

El **Lenguaje Unificado de Modelado (UML)** fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML **describen los límites, la estructura y el comportamiento del sistema y los objetos** que contiene.

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML guarda una relación directa con el análisis y el diseño orientados a objetos.

El **UML se perfecciona continuamente**. UML 2.0 extiende las especificaciones de UML para cubrir más aspectos de desarrollo, incluido Agile. La meta era reestructurar y perfeccionar UML de forma que la facilidad de uso, la implementación y la adaptación se simplificaran. Actualmente la versión es la 2.5.

Estas son algunas de las actualizaciones de los diagramas UML:

- Mayor integración entre modelos estructurales y de comportamiento.
- Capacidad de definir jerarquía y desglosar un sistema de software en componentes y subcomponentes.
- UML 2.0 eleva el número de diagramas a 13.

Tipos de diagramas UML

En la versión 2 de UML se agregaron 3 tipos de diagramas (el de estructura compuesta, el de interacción y el de tiempo) a los 11 ya existentes en la versión 1. También desaparece el Diagrama de colaboración que no forma parte de UML 2.x, por lo que serían 13 en total.

Los diferentes tipos de diagramas se dividen en dos grupos, los que describen la estructura del sistema y los que describen su comportamiento. Dentro de los de comportamiento hay a su vez un subgrupo con los de interacción.

1) Diagramas de estructura (Qué componentes hay)

- **Diagrama de clases:** Describe los diferentes tipos de objetos en un sistema y las relaciones existentes entre ellos. Dentro de las clases muestra las propiedades y operaciones, así como las restricciones de las conexiones entre objetos.
- **Diagrama de objetos:** (También llamado Diagrama de instancias) Foto de los objetos en un sistema en un momento del tiempo.
- **Diagrama de paquetes:** Muestra la estructura y dependencia entre paquetes (librerías), los cuales permiten agrupar elementos (no solamente clases) para la descripción de grandes sistemas.

- **Diagrama de despliegue:** Muestra la relación entre componentes o subsistemas software y el hardware donde se despliega o instala.
- **Diagrama de estructura compuesta:** Descompone jerárquicamente una clase mostrando su estructura interna.
- **Diagrama de componentes:** Muestra la jerarquía y relaciones entre componentes de un sistema software.

2) Diagramas de comportamiento (Cómo se comportan)

- **Diagrama de casos de uso:** Permite capturar los requerimientos funcionales de un sistema.
- **Diagrama de estado:** Permite mostrar el comportamiento de un objeto a lo largo de su vida.
- **Diagrama de actividad:** Describe la lógica de un procedimiento, un proceso de negocio o workflow.
- **Diagramas de interacción** (Subgrupo dentro de los diagramas de comportamiento):
Describen cómo los grupos de objetos colaboran para producir un comportamiento. Son los siguientes:
 - **Diagrama de secuencia:** Muestra los mensajes que son pasados entre objetos en un escenario.
 - **Diagrama de comunicación:** Muestra las interacciones entre los participantes haciendo énfasis en la secuencia de mensajes.
 - **Diagrama de (visión de conjunto o resumen de) interacción:** Se trata de mostrar de forma conjunta diagramas de actividad y diagramas de secuencia.
 - **Diagrama de tiempo:** Pone el foco en las restricciones temporales de un objeto o un conjunto de objetos.

