

¿QUE ES UML?

UML, o **Lenguaje Unificado de Modelado** (Unified Modeling Language, en inglés), es un conjunto de herramientas visuales que se utilizan para representar cómo funciona un sistema de software. Su principal objetivo es ayudar a visualizar, entender y comunicar cómo diferentes partes de un programa interactúan entre sí.

Imagina que estás construyendo una casa. Antes de comenzar, haces planos detallados que muestran la distribución de las habitaciones, las conexiones entre ellas y los materiales que vas a usar. UML es como esos planos, pero para programas de software. Estos diagramas ayudan a los desarrolladores, diseñadores y otras partes involucradas a ver el "mapa" del sistema que se va a construir, de manera que todos tengan la misma idea clara de cómo funcionará.

¿EN QUE CONSISTEN LOS 11 DIAGRAMAS DISPONIBLES DE UML?

Diagramas estructurales

Estos diagramas se enfocan en mostrar cómo está compuesto el sistema, es decir, las piezas que lo forman y cómo se relacionan entre sí. Son como el "esqueleto" del sistema.

1. Diagrama de clases

Este es uno de los más importantes. Muestra los objetos o "clases" principales de un sistema y cómo están relacionados. Imagina que estás construyendo una ciudad, el diagrama de clases te diría cuáles son los edificios (clases), qué hacen (atributos y métodos), y cómo se conectan entre sí (relaciones). Por ejemplo, en una tienda online, podrías tener clases como "Producto", "Cliente", "Pedido", y ver cómo interactúan.

2. Diagrama de objetos

Similar al diagrama de clases, pero en lugar de enfocarse en tipos de objetos (las "clases"), muestra instancias reales de esos objetos. Es como ver ejemplos específicos de los "edificios" de tu ciudad: no solo "Producto", sino el producto "Camisa azul", con toda la información que lo describe. Es como una foto momentánea de un sistema funcionando.

3. Diagrama de componentes

Este diagrama muestra cómo las partes físicas del sistema, como archivos o bibliotecas de código, encajan entre sí. Piensa en un diagrama que muestra cómo se ensamblan las piezas de un automóvil. En un software, esto puede ser cómo los módulos o partes grandes del código se unen para hacer funcionar todo el sistema.

4. Diagrama de despliegue

Muestra dónde se ejecuta el software. Por ejemplo, si tienes una aplicación que corre en tu teléfono, el diagrama de despliegue te dirá que partes del sistema están en el teléfono, en un servidor o en la nube. Es como un mapa que te dice dónde está cada pieza del sistema.

5. Diagrama de paquetes

Agrupar varias clases o componentes en paquetes. Es útil cuando el sistema es muy grande, para que puedas organizar las clases en "categorías" o módulos más manejables. Es como organizar libros en estanterías: los libros son las clases, y las estanterías son los paquetes.

6. Diagrama de estructura compuesta

Este diagrama te muestra cómo los objetos o partes de un sistema trabajan juntos de manera interna. Es como abrir una máquina para ver todas sus piezas y entender cómo encajan y cooperan. Te ayuda a ver cómo diferentes partes pequeñas interactúan en un nivel más detallado.

Diagramas de comportamiento

Estos diagramas te muestran cómo se comporta el sistema o cómo las piezas interactúan cuando el sistema está en funcionamiento. Es como ver un sistema en movimiento, mostrando el flujo de las cosas.

7. Diagrama de casos de uso

Este diagrama se enfoca en las acciones que los usuarios pueden hacer con el sistema. Es como una lista de todas las cosas que puedes hacer en una aplicación. Por ejemplo, en una tienda online, los casos de uso serían "registrarse", "buscar productos", "realizar un pedido", etc. Ayuda a visualizar todas las posibles interacciones que el usuario tendrá con el sistema.

8. Diagrama de secuencia

Este diagrama muestra cómo los objetos del sistema interactúan en un orden específico. Es como un guion de una película: te dice quién habla primero, luego quién responde, y así sucesivamente. Te ayuda a entender el flujo de mensajes entre los objetos y en qué orden ocurren las cosas. Por ejemplo, al hacer una compra en línea, el cliente selecciona el producto, luego el sistema procesa el pedido y finalmente confirma la compra.

9. Diagrama de comunicación

Similar al diagrama de secuencia, pero aquí se enfoca más en cómo los objetos se comunican entre sí, en lugar de enfocarse en el tiempo. Es como una red de

conversaciones: quién habla con quién y qué mensajes se envían, pero sin preocuparnos tanto por el orden.

10. Diagrama de actividades

Este diagrama muestra el flujo de trabajo de un proceso. Es como un mapa que te guía paso a paso a través de un conjunto de acciones. Por ejemplo, si estás registrando un nuevo usuario en una aplicación, el diagrama de actividades te mostraría las acciones: "ingresar nombre", "crear contraseña", "verificar email", y así sucesivamente.

11. Diagrama de estados

Este diagrama muestra cómo un objeto o sistema cambia de estado dependiendo de lo que ocurra. Es útil para ver cómo un sistema responde a diferentes eventos. Por ejemplo, un pedido en una tienda online podría estar en el estado "pendiente", luego "en proceso", y finalmente "enviado". El diagrama de estados te dice cómo cambia el sistema en función de diferentes acciones o situaciones.

GLOSARIO UML

- **Actividad:** Es una acción o tarea que realiza el sistema. En UML, las actividades se representan en el diagrama de actividades, donde se muestra el flujo de trabajo de un proceso.
- **Asociación:** Es una relación entre dos o más clases u objetos en un diagrama de clases. La asociación indica que hay una conexión o vínculo entre ellos.
- **Caso de uso:** Representa una interacción entre un usuario y el sistema. Se describe una función que el sistema puede realizar, como "hacer una reserva" o "registrarse". Se muestra en el diagrama de casos de uso.
- **Clase:** Un grupo de objetos con características similares, representada en un diagrama de clases. Por ejemplo, la clase "Vehículo" puede incluir diferentes objetos como "Auto" o "Moto".
- **Componente:** Es una parte física del sistema, como un archivo o módulo. Se muestra en un diagrama de componentes.
- **Comunicación:** En UML, el diagrama de comunicación muestra cómo los objetos se envían mensajes entre sí.
- **Composición:** Es una forma fuerte de asociación entre dos clases donde una clase "contiene" otra y depende completamente de ella. Si la clase "contenedora" se elimina, también se elimina la clase contenida.
- **Despliegue:** Se refiere a cómo las partes del sistema se instalan o ejecutan en hardware. El diagrama de despliegue muestra la distribución de componentes en diferentes nodos.

- **Estado:** Representa una condición en la que un objeto puede estar en un momento dado. El diagrama de estados muestra cómo un objeto cambia de estado en respuesta a eventos.
- **Instancia:** Es un objeto concreto o específico que pertenece a una clase. En UML, una instancia se representa en un diagrama de objetos, que muestra ejemplos específicos de clases.
- **Interfaz:** Es una definición de métodos que una clase debe implementar. Se utiliza para garantizar que diferentes clases puedan trabajar juntas de manera coherente.
- **Mensaje:** Es la información que pasa entre objetos en un diagrama de secuencia o comunicación. Los mensajes representan la interacción entre diferentes partes del sistema.
- **Objeto:** Una instancia específica de una clase. Por ejemplo, si "Coche" es una clase, un objeto sería "Mi coche rojo". Los diagramas de objetos representan estos ejemplos específicos.
- **Paquete:** Un contenedor que agrupa clases o componentes relacionados. Los diagramas de paquetes ayudan a organizar las clases y simplificar la visualización de sistemas complejos.
- **Relación:** Define cómo dos o más elementos en UML están conectados entre sí. Las relaciones pueden ser asociaciones, dependencias, generalizaciones, entre otras.
- **Secuencia:** El orden en que los mensajes se envían entre objetos para realizar una tarea. Los diagramas de secuencia muestran este flujo de interacción.
- **Sistema:** Conjunto de elementos interconectados que trabajan juntos para llevar a cabo funciones específicas.
- **UML (Unified Modeling Language):** Lenguaje de modelado estándar que se usa para visualizar, especificar, construir y documentar los componentes de un sistema de software.