

UML

Sitio: Agencia de Habilidades para el Futuro
Curso: Modelado y Diseño de Software 1º D
Libro: UML

Imprimido por: Eduardo Moreno
Día: domingo, 23 de marzo de 2025, 10:33

Tabla de contenidos

1. Introducción

1.1. ¿Por qué es necesario un lenguaje común?

1.2. Definición de UML

2. Conociendo UML

3. Modelo Objeto

3.1. Ventajas de trabajar con objetos



Introducción

En los principios de la computación, los programadores/as no realizaban análisis muy profundos sobre el problema por resolver. Con frecuencia, comenzaban a escribir el programa desde el principio, y el código necesario se escribía directamente.

Hoy en día, es necesario contar con un **plan** bien analizado.

Un plan para el cliente

Un cliente tiene que comprender qué es lo que hará un equipo de desarrolladores/as; además tiene que ser capaz de señalar cambios si no se han captado claramente sus necesidades (o si cambia de opinión durante el proceso).

Un plan para el equipo de desarrollo

El desarrollo es un esfuerzo orientado a equipos, por lo que cada uno de sus miembros tiene que saber qué lugar toma su trabajo en la solución final (así como saber cuál es la solución en general).

Todo el equipo debe desarrollar su parte teniendo en cuenta que la suma de los engranajes deben resolver el problema original.

Ante el aumento de las nuevas tecnologías, los sistemas informáticos crecen en complejidad para el desarrollo. En ellos se encuentran diversas piezas de hardware y software que se comunican a grandes distancias mediante una red, la misma que está vinculada a bases de datos que, a su vez, contienen enormes cantidades de información.



La clave está en organizar el proceso de diseño de tal forma que las personas involucradas en el desarrollo del sistema lo comprendan.

En el **índice** de la derecha podrás recorrer los contenidos revisando los contenidos todas las veces que lo necesites.



Un lenguaje común

Como ya sabemos, el desarrollo de software pedido por un cliente requiere de un plan. Este plan tiene que ser comprendido por todas las personas involucradas en el proyecto, que tienen diferentes habilidades y capacidades y aportan desde su especialización. Por eso es fundamental que dicho plan esté expresado en un lenguaje común a todo el equipo, que evite equivocaciones e interpretaciones erróneas.

Ese lenguaje es el **UML** (Lenguaje Unificado de Modelado), que permite la organización y la comunicación de todos los involucrados en el desarrollo software. UML permite trabajar a partir de diagramas para representar la información más importante del plan.

Para recordar

Tal como dijimos en la semana 1, el arquitecto/a no podría crear una estructura como es un edificio de oficinas sin crear primero un **anteproyecto o plano detallado**.

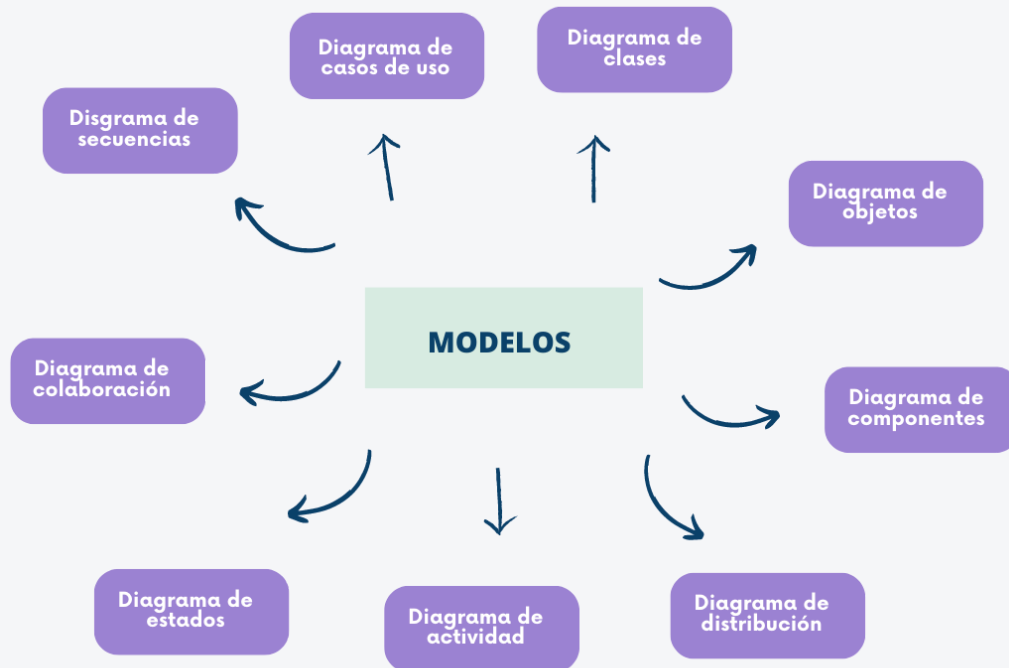
La idea es que así como un arquitecto le muestra un anteproyecto a la persona que lo contrató, se deberá **mostrar el diseño del software al cliente**.

Cada diagrama del UML muestra una parte distinta del mismo modelo del programa software.



¿Qué es UML?

Observemos el siguiente esquema que muestra los diagramas de UML que te permiten diseñar el sistema o producto software.



UML son **diagramas** que permiten ver el sistema desde distintos puntos de vista. Hay distintos tipos de diagramas.

El **diagrama de clases** muestra al sistema o producto software en el aspecto estático, es decir la radiografía del sistema en un momento dado.

El **diagrama de secuencia** muestra al sistema en el aspecto dinámico, es decir la relación de los objetos y la forma de comunicarse a través del tiempo.

Todos estos diagramas de la imagen los iremos viendo y utilizando a lo largo de la cursada.



Lenguaje unificado de modelado (UML)

En la semana anterior recordamos el modelo E_R de base de datos como práctica. También aprendiste qué es modelar cuando vas a desarrollar un producto software.



¿Ya descargaste el material?

Hacé clic en el botón para acceder.

En esta semana, vas a conocer las primeras herramientas de UML, para diagramar la información que tu proyecto necesita.

Comencemos con la definición

El **UML** (Lenguaje Unificado de Modelado) es una de herramienta que permite a los desarrolladores/as de software (en este caso son vos y tu equipo) generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas (en este caso al cliente o compañeros/as de desarrollo).

Si recordás lo visto en la semana anterior, el **desarrollo de software** es la creación de un programa para un cliente; ste último es quien tiene el problema de información que debe ser resuelto.

Volvamos al caso de Manuel: necesita un software para gestionar su agencia de viajes, para que toda la información sobre tours, clientes, pagos y deudas esté ordenada y pueda manejarse de manera eficiente.

Dado que el desarrollo de software es una actividad humana, hay muchas posibilidades de cometer errores en cualquier etapa del proceso, por ejemplo, se pudo haber malentendido al cliente, es decir, probablemente se produjo un documento que el cliente no pudo comprender. Tal vez, ese documento tampoco fue comprendido por otros/as desarrolladores/as del equipo, pudiendo generarse un programa difícil de utilizar, y eso significa que no logramos una solución al problema original del cliente. Por eso es tan importante utilizar un lenguaje claro, completo y fácil de comunicar... ¡UML!

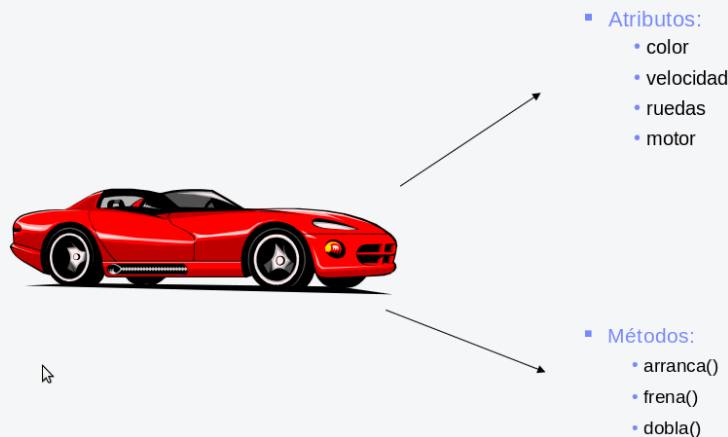




Pero, ¿qué es un objeto?

Lo primero que debemos saber respecto del UML son los distintos aspectos de la programación orientada a objetos. Pero, ¿qué es un objeto?

Si nos paramos a observar el mundo que nos rodea, podemos apreciar que casi todo está formado por **objetos**. Existen coches, edificios, sillas, mesas, semáforos, ascensores e incluso personas o animales. Todos ellos pueden ser considerados objetos, con una serie de características y comportamientos.



Por ejemplo, existen coches de diferentes marcas, colores, etc. y pueden acelerar, frenar, girar, entre otras tareas.



Un **objeto** es cualquier entidad o elemento que podemos ver o conocer. Y cada objeto pertenece a una clase y tiene atributos y métodos.

Los **programas** son el resultado de la búsqueda y obtención de una solución para un problema del mundo real.



Si redactamos los programas utilizando los mismos términos de nuestro mundo real, es decir, utilizando objetos, y no los términos del sistema o computadora donde se vaya a ejecutar, conseguiremos que estos sean más fáciles de leer y, por lo tanto, más fáciles de modificar.



Vamos a conocer las ventajas de trabajar con objetos.