



P1 T5 DIAGRAMAS CASOS DE USO

CARLOS VILLOLDO, VÍCTOR GONZÁLEZ, MARIO JIMÉNEZ

ÍNDICE

DEFINICIÓN Y USO	3
COMPONENTES PRINCIPALES	4
HERRAMIENTAS DE DISEÑO	5
WEBGRAFÍA	5

1. DEFINICIÓN Y USO

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, el cual está compuesto por la clase (con atributos, métodos y visibilidad) y las relaciones (de herencia, composición, agregación, asociación y uso).

Los diagramas UML son diagramas de clase que representan ideas en formato gráfico/visual de forma eficaz, con el objetivo de hacer que las personas comprendan lo que se quiere transmitir. Representan el diseño, arquitectura y la implementación de sistemas software implicados.

Existen dos tipos de diagramas UML: diagramas de estructura y diagramas de comportamiento. Los diagramas de casos de uso se definen dentro de los diagramas de comportamiento, ya que permiten capturar los requerimientos funcionales de un sistema.

Los diagramas de comportamiento son los diagramas que expresan las secuencias de los estados por los que pasa un objeto a lo largo de su vida en respuesta a eventos.

El diagrama de casos de uso, entonces, es un diagrama de comportamiento que representa sistemas y procesos de programación orientada a objetos. En este diagrama, todos los objetos involucrados se estructuran y se relacionan entre sí.

Representa la forma en la que un cliente (actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en cómo los elementos interactúan (casos de uso).

El caso de uso como tal es una operación específica que se realiza tras una orden de algún agente externo, ya sea a petición de un actor o bien desde la llamada a otro caso de uso.

Los diagramas de casos de uso, entonces, sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios u otros sistemas.

Entonces, un diagrama de casos de uso debe:

- Demostrar las funcionalidades que satisfacen las necesidades del usuario.
- Apoyar el proceso de verificación y validación. Esto significa que el diagrama de casos de uso debe poderse utilizar para revisar el software y validar que todas las funcionalidades requeridas para los usuarios están incluidas en este.
- Ser de fácil lectura e interpretación para todos (no sólo para los desarrolladores de software sino también para clientes y usuarios finales).

2. COMPONENTES PRINCIPALES

Los componentes principales del diagrama “Casos de Uso” son:

Actor: Representa a una persona o grupo de personas que desempeñan un papel en la interacción con el software.

Caso de uso: Representa una funcionalidad (Conocida en inglés como Feature) que cumple uno o varios requisitos. Al colocarles nombre es recomendable usar verbos infinitivos con un complemento.

Relación: La interacción entre dos casos de uso o de un actor con un caso de uso se representa por medio de una relación, utilizando una línea o una flecha como se muestra en la figura.

Relaciones entre actor y caso de uso

La asociación entre un actor y un caso de uso se denomina relación de comunicación. Bajo esta relación, un actor puede ser:

Activo: Cuando el actor inicia o desencadena la ejecución del caso de uso. Si se está usando flecha, esta debe apuntar al caso de uso. Si la relación no tiene flecha, se considera que el actor es activo como predeterminado.

Pasivo: Cuando el caso de uso es iniciado por el software y no por el actor. La flecha debe apuntar al actor. Las notificaciones y alarmas que emiten a menudo sistemas de información son un buen ejemplo de este tipo de relación.

3. HERRAMIENTAS DE DISEÑO

Claramente los diagramas UML no se hacen con un programa como Paint, sino con las herramientas UML las que ayudan a utilizar el lenguaje de modelado.

Encontrar la herramienta adecuada no es fácil, ya que no todas ofrecen las mismas funciones.

Algunas de ellas necesitan poca memoria pero ofrecen pocas funciones, otras pueden trazar un diagrama y exportarlos a diferentes lenguajes de programación o importar un modelo desde un código existente, pero muchos de estos programas no ofrecen ninguna función para intercambiar información sobre el proyecto dentro de un equipo.

Algunos ejemplos de buenas herramientas serían las siguientes:

-Moqups: Es útil al crear software nuevos ya que ofrecen asistencia durante el desarrollo. Ofrece ayuda para estar organizados e informados. Ofrece plantillas para diagramas de clases, diagramas de secuencia y diagramas de casos de uso.

Además accede a varios conectores de diagramas UML para lograr una mejor eficiencia. Se puede hacer colaboración en línea para facilitar el trabajo en tiempo real.

-Creately: Es fácil de visualizar el diagramas UML y se puede hacer el trabajo en tiempo real con otros. Se pueden recibir comentarios de sus clientes o compañeros y realizar seguimiento de los cambios de forma sencilla.

Tienen conectores y formas inteligentes, barras de herramientas de contexto y otras funcionalidades avanzadas para dibujar diagramas UML fácilmente.

-GitMind: Se utiliza en navegadores, este ofrece sus propias plantillas, esto significa que no será necesario ir añadiendo todos los elementos a mano pero hacerlo manualmente no tiene dificultad.

Todo se almacena en la nube pudiendo así acceder al diagrama desde cualquier dispositivo pero no ofrece una versión de descarga.

Algo en contra es que solo se puede importar archivos XMind.

4. WEBGRAFÍA

- <https://www.ibm.com/docs/es/elm/6.0.3?topic=requirements-defining-use-cases>
- <https://www.ceac.es/blog/elaborar-diagramas-de-comportamiento-en-entornos-de-desarrollo>
- <http://www.pmoinformatica.com/2021/02/elementos-diagrama-casos-de-uso.html>
- https://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso
- <http://www.pmoinformatica.com/2021/02/diagrama-de-casos-de-uso-definicion-elementos-ejemplos-como-hacer.html>
- <https://www.youtube.com/watch?v=orvAkFFWo5o>
- <https://geekflare.com/es/about-uml-diagram-and-tools/>