

CENDESI eBusiness IT

<http://www.cendesi.com/site/es/>

Modelos y UML (Parte III) (Artículos CENDESI)

Artículos CENDESI

Conocimiento Libre



Análisis y Diseño de Software Orientado a Objetos con UML

Modelos de Software y Lenguaje UML (Parte III):

Diagramas UML

Hasta ahora hemos hecho en los dos anteriores artículos un acercamiento sobre el Lenguaje de Modelado de Sistemas UML; su importancia en la industria de software; las herramientas de modelado y cuáles son las ventajas de su aplicación con respecto a la documentación tradicional, en proyectos de desarrollo de sistemas. En este artículo haremos una introducción sobre los diferentes diagramas UML y cuál es su aplicación en la representación de sistemas de software.

Antes de introducir cada uno de los diagramas de UML, debemos recordar que como lenguaje que es, si no se usa con frecuencia, se puede olvidar un poco. La meta de estos diagramas y el lenguaje entero es captar las estructuras y arquitectura de un sistema de software, y diseñarlo con la mayor exactitud posible, con respecto a los requerimientos y/o resultados esperados.

Tipos de Diagramas de modelado UML

Existen diversos tipos de diagrama que pueden ser implementados en la representación de su proyecto de software. Cada uno de ellos contiene un nivel de información relacionado con el objeto de representación del mismo. Los principales diagramas de UML son:

Diagramas de Casos de Uso

El diagrama de casos de uso representa la forma en cómo un Actor opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan. Estas

operaciones son las que llamamos casos de uso. Los diagramas de casos de uso tienen la responsabilidad de documentar los macro requisitos del sistema.

Los símbolos principales de un caso de uso son el actor; el óvalo del caso de uso y las relaciones de asociación.

Diagrama de Actividades

Los diagramas de Actividades en UML son el equivalente de diagramas de flujo en la documentación clásica de software. Estos diagramas son de gran utilidad para analizar procesos sin requerir el uso de códigos de programación e igualmente para refinar procesos que comprenden el sistema.

Diagrama de Clases

Los diagramas de clases representan una visión estática del sistema a desarrollar; estos no describen comportamiento o interacción del mismo y solamente representan las clases de un sistema y cuál es la relación existente entre ellas.

En el desarrollo de un sistema pueden existir diversos diagramas de clases que representan las partes del mismo. No es necesario el uso de un solo diagrama de clases para representar un sistema, y generalmente suele no hacerse. La importancia de los diagramas de clases es mostrar desde varias perspectivas la estructura de un sistema visto desde las clases que lo componen. Para representar una visión de comportamiento de un sistema se utilizan los diagramas de interacción.

Diagramas de Interacción

Como se mencionó antes, estos diagramas representan la visión de comportamiento o interacción entre los diferentes componentes que conforman su sistema. Existen dos tipos de diagrama de interacción: Los de secuencia y los de colaboración, teniendo ambos la misma finalidad siendo su única diferencia es respecto a su construcción.

Los diagramas de secuencia usan la perspectiva de línea de tiempo para representar el comportamiento del sistema o componente de software. Esta línea de tiempo se lee en el

orden que va desde la esquina superior izquierda hasta la esquina inferior derecha del diagrama.

Los diagramas de colaboración no están ordenados en tiempo, por lo que su lectura está determinada por números que indican la secuencia de hechos.

Diagramas de Estado

Mientras que los diagramas de interacción muestran los objetos y los mensajes que se pasan entre ellos, un diagrama de estado muestra el estado cambiante de un solo objeto.

Diagrama de Componentes

Un diagrama de componentes representa las dependencias entre componentes software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. El diagrama de componente hace parte de la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos de ejecución. Esta vista proporciona la oportunidad de establecer correspondencias entre las clases y los componentes de implementación y nodos. La vista de implementación de un sistema se representa con los diagramas de componentes.