

UML y UP: Análisis y Diseño Orientado a Objetos

Módulo 1

Introducción a Enterprise Architect

¿Por qué Enterprise Architect?

Es la herramienta que vamos a utilizar durante el curso para realizar los diagramas.

Aunque hay otros en el mercado, elegimos este por ser **el programa más utilizado para documentar utilizando UML**. Es muy completo, bastante sencillo de usar y fácil de integrar con otras herramientas. Podemos exportar, por ejemplo, la documentación a un archivo de Word o generar código fuente a partir de un diagrama.



Conceptos Generales

Estereotipos

Un **estereotipo** es un **nuevo elemento de UML** que **extiende, a partir de uno existente, quedando definido con una semántica extendida y un nuevo icono gráfico, aunque este último es opcional**. En caso de no establecer un icono para el estereotipo, se visualiza de la forma **<<nombreEstereotipo>>**.

Los estereotipos pueden utilizarse para cualquier elemento de UML, como ser clases y/o relaciones. Las clases estereotipadas más comunes son las clases del tipo **<<Boundary>>**, **<<Control>>**

y **<<Entity>>**, y entre las relaciones la más conocida es **<<use>>**.

Valor etiquetado (Tagged Values)

El valor etiquetado se utiliza para guardar información adicional dentro de un elemento de UML. Está formado por una etiqueta (*tag*) y un valor (*value*), donde por ejemplo, se puede guardar el nombre de la persona que construyó una clase o la versión de la misma.

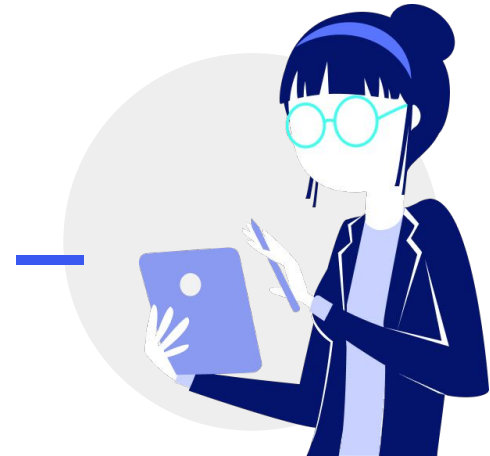
Ingeniería Directa

La **ingeniería directa** es el **proceso que permite generar código fuente en cualquier lenguaje a partir de los distintos diagramas de UML**.

Por ejemplo, a partir del Diagrama de Clases es posible generar el “esqueleto del sistema”, es decir el código fuente formado por clases y métodos, pero vacíos.

El ***Enterprise Architect*** permite **generar código en diversos lenguajes**, entre ellos, Java, VB.Net, C#, C++, Python, Delphi y PHP. Cuenta con un editor de código fuente incorporado lo que hace posible la navegación desde el modelo directamente al código fuente en el mismo entorno.

Las plantillas de generación de código permiten personalizar el código fuente generado de acuerdo con las especificaciones del proyecto.



Ingeniería Inversa

La **ingeniería inversa** es el **proceso que permite generar diagramas a partir de código fuente**.

Se utiliza generalmente para construir el diagrama de clases a partir de las clases de código fuente de cualquier lenguaje.

El *Enterprise Architect* permite **generar el diagrama de clases a partir del código fuente de diversos lenguajes** tales como Java, VB.Net, C#, C++, Python, Delphi y PHP.

El Lenguaje XMI

XMI significa **XML Metadata Interchange**, y es una forma de almacenar los diagramas de UML en un formato de texto basado en XML.

El objetivo de XMI es **permitir la portabilidad de un proyecto armado en UML desde cualquier tipo de aplicación**. Por ejemplo, es posible utilizar el *Enterprise Architect* y guardar el proyecto en formato .xml, para luego utilizar otra aplicación como *Rational* o *Poseidon UML* y poder abrir el proyecto sin inconvenientes.

Compilado, pruebas y ejecución

Con *Enterprise Architect* también es posible **compilar, probar, depurar, ejecutar y correr scripts de implementación, todo desde su entorno de desarrollo.**

Con la capacidad de generar clases de prueba NUnit y JUnit a partir de clases de origen utilizando Transformaciones MDA e integrar el proceso de prueba directamente en el IDE de Enterprise Architect, desde sus últimas versiones es posible integrar UML y el modelado en el proceso de compilación / prueba / ejecución / implementación.

Seguimiento del código en ejecución (Debug/Depuración)

Además, *Enterprise Architect* incluye **capacidades de depuración** para Java, .NET y Microsoft Native (C ++, C y VB).

Los depuradores integrados en *Enterprise Architect* están diseñados específicamente para **permitir que un desarrollador o tester capture la información de seguimiento mientras “recorren” el código de ejecución.**



**¡Sigamos
trabajando!**