

Utilidades de los diagramas de clases

Entornos de desarrollo – Marcos Fernández Sellers

Generación automática de código

La **Generación Automática de Código** consiste en la creación de código fuente de manera automatizada utilizando herramientas CASE. El proceso pasa por establecer una correspondencia entre los elementos formales de los diagramas y las estructuras de un lenguaje de programación concreto. El diagrama de clases es un buen punto de partida, porque permite una traducción bastante directa de las clases representadas gráficamente a clases escritas en un lenguaje de programación específico, como Java o C++.

Normalmente las herramientas de generación de diagramas UML incluyen la facilidad de la generación, o actualización automática de código fuente, a partir de los diagramas creados.

Ingeniería Inversa

La **ingeniería inversa** se define como el proceso de analizar código, documentación y comportamiento de una aplicación para identificar sus componentes actuales y sus dependencias, y para extraer y crear una abstracción del sistema e información del diseño. El sistema en estudio no es alterado, sino que se produce un conocimiento adicional del mismo.

La ingeniería inversa puede ser de varios tipos:

- Ingeniería inversa de datos: Se aplica sobre algún código de bases datos (aplicación, código SQL, etc.) para obtener los modelos relacionales o sobre el modelo relacional para obtener el diagrama entidad-relación.
- Ingeniería inversa de lógica o de proceso: Cuando la ingeniería inversa se aplica sobre el código de un programa para averiguar su lógica (reingeniería), o sobre cualquier documento de diseño para obtener documentos de análisis o de requisitos.
- Ingeniería inversa de interfaces de usuario: Se aplica con objeto de mantener la lógica interna del programa para obtener los modelos y especificaciones que sirvieron de base para la construcción de la misma, con objeto de tomarlas como punto de partida en procesos de ingeniería directa que permitan modificar dicha interfaz.

Visual Paradigm

Es una de las herramientas más utilizadas para la elaboración de diagramas UML.

Visual Paradigm for UML (VP-UML): Incluye una versión para uso no comercial que se distribuye libremente sin más que registrarse para obtener un archivo de licencia. Incluye diferentes módulos para realizar desarrollo UML, diseñar bases de datos, realizar actividades de ingeniería inversa y diseñar con Agile. Es compatible con los IDE de Eclipse, Visual Studio .net, IntelliJDEA y NetBeans. Multiplataforma, incluye instaladores para Windows y Linux.

Aquí tienes el enlace a la página oficial de Visual Paradigm: <http://www.visual-paradigm.com>

Algunas características concretas:

- Incluye una versión para uso no comercial, aunque se debe aclarar que viene con funcionalidad limitada, que se distribuye bajo licencia LGPL. Es posible solicitar una licencia de prueba para treinta días que utilizaremos cuando veamos la parte de ingeniería directa e inversa y generación de código.
- Es multiplataforma.
- Compatible con UML 2.0.
- Admite la generación de informes en formatos PDF, HTML y otros.
- Incluye un módulo para integrarse con NetBeans.
- Permite realizar actividades de ingeniería inversa y directa. Esto junto con la consideración anterior permite generar código en un proyecto NetBeans directamente a partir del diseño de clases, ahorrándonos trabajo.

Descarga e instalación de Visual Paradigm

Obtenemos los archivos desde la [página de descargas de Visual Paradigm](#) (también disponible desde el Moodle).

Ofrece dos versiones

- Visual Paradigm for UML (VP-UML), versión de prueba de 30 días.
- Versión Community-Edition, para uso no comercial (gratuito).

La versión Community-Edition incluye algunas de las funcionalidades de la versión completa, entre las que no se encuentra la generación de código ni la ingeniería inversa, que se verán al final de la unidad por lo que **se recomienda empezar por la versión completa de prueba por 30 días**.

Ejecutaremos el archivo de instalación, que tendrá diferente extensión si es para Windows o para Linux. En nuestro caso suponemos que lo hacemos en un equipo con Ubuntu Desktop (en la máquina virtual que hemos estado utilizando). Se debe tener en cuenta que en el nombre se incluye la versión y la fecha en la que apareció, por lo que estos datos pueden cambiar con el tiempo.

```
usuario@equipo:~/VP/ chmod +x VP_Suite_Linux_16_3_20220422.sh
usuario@equipo:~/VP/sudo ./VP_Suite_Linux_16_3_20220422.sh
```

Iniciar Visual Paradigm

Una vez realizada la instalación tendremos una entrada en el menú Aplicaciones llamada Otras, si trabajamos con Linux o bien una entrada de menú en el botón Inicio para Visual Paradigm, si es que trabajamos en Windows. En cualquiera de los casos para abrir la herramienta buscamos la opción Visual Paradigm for UML, que se abrevia como VP-UML. Al hacer clic se abrirá el programa, y **debemos seleccionar la opción de 'Evaluation'**.

Iniciar VP-UML desde NetBeans

Para integrar VisualParadigm con Netbeans, debemos seguir el siguiente [tutorial](#). Habrá que establecer la ruta de instalación de Netbeans (desde Netbeans podemos comprobarla en *Help/About*).

Una vez integrado, abrimos o reiniciamos Netbeans, al que se ha incorporado una pequeña diferencia, y es que podemos añadir a un proyecto en desarrollo existente un proyecto VP-UML. ¿Cómo lo hacemos?

Estando en la ventana de Proyectos, si hacemos clic con el botón secundario sobre un proyecto vemos una serie de opciones, como compilar o construir, ahora, además, nos aparecerá un submenú de Visual Paradigm Enterprise.

Los proyectos de Visual Paradigm se podrán almacenar en el directorio por defecto, que se denomina vpproject y cuelga del directorio principal del proyecto NetBeans, o en otra ubicación. Nosotros nos quedaremos con la opción por defecto.

También podemos importar un proyecto VP-UML que tengamos ya creado seleccionándolo al crear el proyecto existente.

Una vez creado o importado el proyecto, tendremos una serie de botones en la zona superior derecha que nos permitirán crear los diferentes diagramas de UML, y que queden asociados al proyecto NetBeans.

[Videotutorial sobre integración de Netbeans con Visual Paradigm](#)

Actividad

Una vez realizada la integración de Netbeans con Visual Paradigm, vamos a probar las dos utilidades principales que hemos visto al principio: la generación automática de código, y la generación de diagramas a partir de código.

Para ello, vamos a seguir dos videotutoriales en la que nos aparecerán todos los pasos a realizar:

- [Generar diagramas UML a partir de código Java.](#)
- [Generar código desde un diagrama UML](#)

Deberemos realizar un documento PDF con las respectivas capturas de pantalla que demuestren que hemos sido capaces, en primer lugar, de integrar Netbeans y Visual Paradigm, y, en segundo lugar, que hemos completado ambos tutoriales y hemos probado ambas utilidades de los diagramas UML.