

Manual de Umbrello UML Modeller

Umbrello UML Modeller Autores

Traductor: Marcos Fouces Lago

revisión 1.2 (2003-10-15)

Copyright © 2001 Paul Hensgen

Copyright © 2002, 2003 Umbrello UML Modeller Autores

Umbrello UML Modeller ayuda en el proceso de desarrollo de software usando el estándar 'Unified Modelling Language' (UML) que le permitirá crear diagramas para diseñar y documentar sus sistemas.

Tabla de contenidos

[1. Introducción](#)

[2. Introducción a UML](#)

[Acerca de UML](#)

[Elementos de UML](#)

[Diagrama de casos de uso](#)

[Diagrama de clases](#)

[Diagramas de secuencia](#)

[Diagramas de colaboración](#)

[Diagrama de estado](#)

[Diagrama de actividad](#)

[Elementos de ayuda](#)

[Diagramas de componentes](#)

[Diagramas de implementación](#)

[Diagramas de relación de entidad](#)

[Conceptos del diagrama de relaciones de entidades extendido \(EER\)](#)

[3. Trabajando con Umbrello UML Modeller](#)

[Interfaz de usuario](#)

[Título](#)

[Ventana de documentación](#)

[Área de trabajo](#)

[Crear, cargar y guardar proyectos](#)

[Nuevo proyecto](#)

[Guardar modelo](#)

[Cargar modelo](#)

[Editar modelo](#)

[Añadir y eliminar diagramas](#)

[Crear diagramas](#)

[Eliminar diagramas](#)

[Renombrar diagramas](#)

[Editar diagramas](#)

[Insertar diagramas](#)

[Borrar elementos](#)

[Editar elementos](#)

[Editar clases](#)

[Asociaciones](#)

[Notas, textos y cajas](#)

[4. Importación y generación de código](#)

[Generación de código](#)

[Generación de código](#)

[Importación de código](#)

[5. Otras características](#)

[Otras características de Umbrello UML Modeller](#)

[Copia de objetos como imágenes PNG](#)

[Exportar como imagen](#)

[Impresión](#)

[Carpetas lógicas](#)

[6. Autores e historia](#)

[7. Copyright](#)

Capítulo 1. Introducción

Umbrello UML Modeller es una herramienta de diagramas " que ayuda en el proceso del desarrollo de software. Umbrello UML Modeller le facilitará la creación de un producto de alta calidad, especialmente durante fases de análisis y diseño del proyecto. UML también puede usarse para documentar sus diseños de software para ayudarle a usted y al resto de desarrolladores.

Tener una buena maqueta del software es la mejor forma de comunicarse con otros desarrolladores que participen en el proyecto, así como con sus clientes. Una buena maqueta de extremadamente importante para los proyectos de mediano o gran tamaño, pero también resulta útil para los más pequeños. Aunque trabaje en un pequeño proyecto personal, podrá beneficiarse de una buena maqueta porque esta le proporcionará una visión global que le ayudará en la creación de un mejor código.

UML es el lenguaje de diagramas que se utiliza para la descripción de tales maquetas. Es posible representar las ideas en UML utilizando diversos tipos de diagramas. Umbrello UML Modeller 1.2 soporta los siguientes tipos:

- Diagrama de clase
- Diagrama de secuencia
- Diagrama de colaboración
- Diagrama de caso de uso
- Diagrama de estado
- Diagrama de actividad
- Diagrama de colaboración
- Diagrama de secuencia

Puede encontrar más información sobre UML en la [página web de OMG](http://www.omg.org), <http://www.omg.org>, quien creó el estándar UML.

Esperamos que disfrute de Umbrello UML Modeller y que este le sirva en la creación de software de gran calidad. Umbrello UML Modeller es una herramienta libre, y lo único que le pedimos es que informe a los desarrolladores de Umbrello UML Modeller de fallos, problemas o sugerencias en la dirección de correo electrónico (uml-devel@lists.sourceforge.net).

Capítulo 2. Introducción a UML

Tabla de contenidos

[Acerca de UML](#)

[Elementos de UML](#)

[Diagrama de casos de uso](#)

[Diagrama de clases](#)

[Diagramas de secuencia](#)

[Diagramas de colaboración](#)

[Diagrama de estado](#)

[Diagrama de actividad](#)

[Elementos de ayuda](#)

[Diagramas de componentes](#)

[Diagramas de implementación](#)

[Diagramas de relación de entidad](#)

[Conceptos del diagrama de relaciones de entidades extendido \(EER\)](#)

Acerca de UML

Este capítulo proporciona una introducción rápida a las características básicas de UML. Tenga en cuenta que no se trata de un manual de referencia de UML sino de una breve introducción. Si desea más información sobre UML, o sobre el análisis y diseño del software en general, le recomendamos que lea cualquier de los libros publicados sobre el tema. También hay una buena cantidad de tutoriales en Internet, que puede utilizar como punto de partida.

El lenguaje unificado de diagrama o notación (UML) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está controlado por el grupo de administración de objetos (OMG) y es el estándar de descripción de esquemas de software.

UML está diseñado para su uso con software orientado a objetos, y tiene un uso limitado en otro tipo de cuestiones de programación.

UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa

alguna parte o punto de vista del sistema. Umbrello UML Modeller soporta los siguientes tipos de diagramas:

- [*Diagrama de casos de uso*](#) que muestra a los actores (otros usuarios del sistema), los casos de uso (las situaciones que se producen cuando utilizan el sistema) y sus relaciones.
- [*Diagrama de clases*](#) que muestra las clases y la relaciones entre ellas.
- [*Diagrama de secuencia*](#) muestra los objetos y sus múltiples relaciones entre ellos.
- [*Diagrama de colaboración*](#) que muestra objetos y sus relaciones, destacando los objetos que participan en el intercambio de mensajes.
- [*Diagrama de estado*](#) muestra estados, cambios de estado y eventos en un objeto o en parte del sistema.
- [*Diagrama de actividad*](#) que muestra actividades, así como los cambios de una a otra actividad junto con los eventos que ocurren en ciertas partes del sistema.
- [*Diagrama de componentes*](#) que muestra los componentes de mayor nivel de la programación (cosas como Kparts o Java Beans).
- [*Diagrama de implementación*](#) que muestra las instancias de los componentes y sus relaciones.
- [*Diagrama de relaciones de entidad*](#) que muestra los datos y las relaciones y restricciones entre ellos.

Elementos de UML

Diagrama de casos de uso

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de *casos de uso* y los actores participantes en el proceso.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen *qué* es lo que debe hacer el sistema, pero no *cómo*.

Umbrello UML Modeller mostrar un diagrama de casos de uso

Caso de uso

Un *caso de uso* describe, —desde el punto de vista de los actores—, un grupo de actividades de un sistema que produce un resultado concreto y tangible.

Los casos de uso son descriptores de las interacciones típicas entre los usuarios de un sistema y ese mismo sistema. Representan el interfaz externo del sistema y especifican qué requisitos de funcionamiento debe tener este (recuerde, únicamente el qué, nunca el cómo).

Cuando se trabaja con casos de uso, es importante tener presentes algunas sencillas reglas:

- Cada caso de uso está relacionado como mínimo con un actor
- Cada caso de uso es un iniciador (es decir, un actor)
- Cada caso de uso lleva a un resultado relevante (un resultado con «valor intrínseco»)

Los casos de uso pueden tener relaciones con otros casos de uso.

Los tres tipos de relaciones más comunes entre casos de uso son:

- `<<include>>` que especifica una situación en la que un caso de uso tiene lugar *dentro* de otro caso de uso
- `<<extends>>` que especifica que en ciertas situaciones, o en algún punto (llamado punto de extensión) un caso de uso será extendido por otro.
- *Generalización* que especifica que un caso de uso hereda las características del «super» caso de uso, y puede volver a especificar algunas o todas ellas de una forma muy similar a las herencias entre clases.

Actor

Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos.

Los actores no representan a personas *físicas* o a sistemas, sino su *rol*. Esto significa que cuando una persona interactúa con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores. Por ejemplo, una persona que proporciona servicios de atención telefónica a clientes y realiza pedidos para los clientes estaría representada por un actor «equipo de soporte» y por otro actor «representante de ventas».

Descripción de casos de uso

Las descripciones de casos de uso son reseñas textuales del caso de uso. Normalmente tienen el formato de una nota o un

documento relacionado de alguna manera con el caso de uso, y explica los procesos o actividades que tienen lugar en el caso de uso.

Diagrama de clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

Umbrello UML Modeller mostrando un diagrama de clases

Clase

Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase (instancias de esa clase) tienen el mismo comportamiento y el mismo conjunto de atributos (cada objeto tiene el suyo propio). En ocasiones se utiliza el término «tipo» en lugar de clase, pero recuerde que no son lo mismo, y que el término tipo tiene un significado más general.

En UML, las clases están representadas por rectángulos, con el nombre de la clase, y también pueden mostrar atributos y operaciones de la clase en otros dos «compartimentos» dentro del rectángulo.

Representación visual de una clase en UML

Atributos

En UML, los atributos se muestran al menos con su nombre, y también pueden mostrar su tipo, valor inicial y otras propiedades. Los atributos también pueden ser mostrados visualmente:

- + Indica atributos *públicos*
- # Indica atributos *protegidos*
- - Indica atributos *privados*

Operaciones

Las operaciones (métodos) también se muestran al menos con su nombre, y pueden mostrar sus parámetros y valores de retorno.

Las operaciones, al igual que los atributos, se pueden mostrar visualmente:

- + Indica operaciones *públicas*
- # Indica operaciones *protegidas*
- - Indica operaciones *privadas*

Plantillas

Las clases pueden tener plantillas, un valor usado para una clase no especificada o un tipo. El tipo de plantilla se especifica cuando se inicia una clase (es decir cuando se crea un objeto). Las plantillas existen en C++ y se introducirán en Java 1.5 con el nombre de Genéricos.

Asociaciones de clases

Las clases se puede relaciones (estar asociadas) con otras de diferentes maneras:

Generalización

La herencia es uno de los conceptos fundamentales de la programación orientada a objetos, en la que una clase «recoge» todos los atributos y operaciones de la clase de la que es heredera, y puede alterar/modificar algunos de ellos, así como añadir más atributos y operaciones propias.

En UML, una asociación de *generalización* entre dos clases, coloca a estas en una jerarquía que representa el concepto de herencia de una clase derivada de la clase base. En UML, las generalizaciones se representan por medio de una línea que conecta las dos clases, con una flecha en el lado de la clase base.

Representación visual de una generalización en UML

Asociaciones

Una asociación representa una relación entre clases, y aporta la semántica común y la estructura de muchos tipos de «conexiones» entre objetos.

Las asociaciones son los mecanismos que permite a los objetos comunicarse entre sí. Describe la conexión entre diferentes clases (la conexión entre los objetos reales se denomina conexión de objetos o *enlace*).

Las asociaciones pueden tener una papel que especifica el propósito de la asociación y pueden ser unidireccionales o bidireccionales (indicando si los dos objetos participantes en la relación pueden intercambiar mensajes entre sí, o es únicamente uno de ellos el que recibe información del otro). Cada extremo de

la asociación también tiene un valor de multiplicidad, que indica cuántos objetos de ese lado de la asociación están relacionados con un objeto del extremo contrario.

En UML, las asociaciones se representan por medio de líneas que conectan las clases participantes en la relación, y también pueden mostrar el papel y la multiplicidad de cada uno de los participantes. La multiplicidad se muestra como un rango [mín...máx] de valores no negativos, con un asterisco (*) representando el infinito en el lado máximo.

Representación visual de una asociación en UML

Acumulación

Las acumulaciones son tipos especiales de asociaciones en las que las dos clases participantes no tienen un estado igual, pero constituyen una relación «completa». Una acumulación describe cómo se compone la clase que asume el rol completo de otras clases que se encargan de las partes. En las acumulaciones, la clase que actúa como completa, tiene una multiplicidad de uno.

En UML, las acumulaciones están representadas por una asociación que muestra un rombo en uno de los lados de la clase completa.

Representación visual de una relación de acumulación en UML

Composición

Las composiciones son asociaciones que representan acumulaciones *muy fuertes*. Esto significa que las composiciones también forman relaciones completas, pero dichas relaciones son tan fuertes que las partes no pueden existir por sí mismas. Únicamente existen como parte del conjunto, y si este es destruido las partes también lo son.

En UML, las composiciones están representadas por un rombo sólido al lado del conjunto.

Otros componentes de los diagrama de clases

Los diagramas de clases pueden contener más componentes aparte de clases.

Interfaces

Las interfaces son clases abstractas, esto es, instancias que no pueden ser creadas directamente a partir de ellas. Pueden contener operaciones, pero no atributos. Las clases pueden heredarse de las interfaces pudiendo así realizarse instancias a partir de estos diagramas.

Tipo de datos

Los tipo de datos son primitivas incluidas en algunos lenguajes de programación. Algunos ejemplos son: bool y float. No pueden tener relación con clases, pero las clases sí pueden relacionarse con ellos.

Enumeraciones

Las enumeraciones son simples listas de valores. Un ejemplo típico de esto sería una enumeración de los días de la semana. Al igual que los tipos de datos, no pueden relacionarse con las clases, pero las clases sí pueden hacerlo con ellos.

Paquetes

Los paquetes, en lenguajes de programación, representan un espacio de nombres en un diagrama se emplean para representar partes del sistema que contienen más de una clase, incluso cientos de ellas.

Diagramas de secuencia

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos. En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

Umbrello UML Modeller mostrando un diagrama de secuencia

Los mensajes pueden ser o bien síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado

hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada. Los mensajes síncronos tienen una caja vertical en un lateral del objeto invocante que muestra el flujo del control del programa.

Diagramas de colaboración

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.

Umbrello UML Modeller mostrando un diagrama de colaboración

Diagrama de estado

Los diagramas de estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto.

Los diagramas de estado ven a los objetos como *máquinas de estado* o autómatas finitos que pueden estar en un conjunto de estados finitos y que pueden cambiar su estado a través de un estímulo perteneciente a un conjunto finito. Por ejemplo, un objeto de tipo *NetServer* puede tener durante su vida uno de los siguientes estados:

- Listo
- Escuchando
- Trabajando
- Detenido

y los eventos que pueden producir que el objeto cambie de estado son

- Se crea el objeto
- El objeto recibe un mensaje de escucha
- Un cliente solicita una conexión a través de la red
- Un cliente finaliza una solicitud
- La solicitud se ejecuta y se termina
- El objeto recibe un mensaje de detención

- etc

Umbrello UML Modeller mostrando un diagrama de estado

Estado

Los estados son los ladrillos de los diagramas de estado. Un estado pertenece a exactamente una clase y representa un resumen de los valores y atributos que puede tener la clase. Un estado UML describe el estado interno de un objeto de una clase particular

Tenga en cuenta que no todos los cambios en los atributos de un objeto deben estar representados por estados, sino únicamente aquellos cambios que pueden afectar significativamente a la forma de funcionamiento del objeto

Hay dos tipos especiales de estados: inicio y fin. Son especiales en el sentido de que no hay ningún evento que pueda devolver a un objeto a su estado de inicio, y de la misma forma no hay ningún evento que pueda sacar a un objeto de su estado de fin.

Diagrama de actividad

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, que únicamente (o mayormente) contienen actividades.

Umbrello UML Modeller mostrando un diagrama de actividad

Los diagramas de actividad son similares a los diagramas de flujo procesales, con la diferencia de que todas las actividades están claramente unidas a objetos.

Los diagramas de actividad siempre están asociados a una *clase*, a una *operación* o a un *caso de uso*.

Los diagramas de actividad soportan actividades tanto secuenciales como paralelas. La ejecución paralela se representa por medio de iconos de fork/espera, y en el caso de las actividades paralelas, no importa en qué orden sean invocadas (pueden ser ejecutadas simultáneamente o una detrás de otra).

Actividad

Una actividad es un único paso de un proceso. Una actividad es un estado del sistema que actividad interna y, al menos, una transición saliente. Las actividades también pueden tener más de una transición saliente, si tienen diferentes condiciones.

Las actividades pueden formar jerarquías, lo que significa que una actividad puede estar formada de varias actividades «de detalle», en cuyo caso las transiciones entrantes y salientes deberían coincidir con las del diagrama de detalle.

Elementos de ayuda

Existen unos pocos elementos en UML que no tiene un valor semántico real en la maqueta, pero que ayudan a clarificar partes del programa. Estos elementos son

- Línea de texto
- Notas de texto y enlaces
- Cajas

Las líneas de texto son útiles para añadir información textual a un diagrama. Es texto es libre y no tiene ningún significado para la maqueta.

Las notas son útiles para añadir información más detallada de un objeto o una situación específica. Tienen la gran ventaja de que se pueden anclar a los elementos UML para mostrar que una nota «pertenece» a un objeto o situación específicos.

Las cajas son rectángulos repartidos libremente que pueden usarse para juntar objetos haciendo los diagramas más legibles. No tienen significado lógico en la maqueta.

Diagramas de componentes

Los diagramas de componentes muestran los componentes del software (ya sea las tecnologías que lo forman como Kparts, componentes CORBA, Java Beans o simplemente secciones del sistema claramente distintas) y los artilugios de que está compuesto como los archivos de código fuente, las librerías o las tablas de una base de datos.

Los componentes pueden tener interfaces (es decir clases abstractas con operaciones) que permiten asociaciones entre componentes.

Diagramas de implementación

Los diagramas de implementación muestran las instancias existentes al ejecutarse así como sus relaciones. También se representan los nodos que identifican recursos físicos, típicamente un ordenador así como interfaces y objetos (instancias de las clases).

Diagramas de relación de entidad

Los diagramas de relaciones de entidad (diagramas ER) muestran el diseño conceptual de las aplicaciones de bases de datos. Representan varias entidades (conceptos) en el sistema de información y las relaciones y restricciones existentes entre ellas. Una extensión de los diagramas de relaciones de entidad llamado «diagramas de relaciones de entidad extendida» o «diagramas de relaciones de entidad mejoradas» (EER), se utiliza para incorporar las técnicas de diseño orientadas a objetos en los diagramas ER.

Umbrello mostrando un diagrama de relaciones de entidad

Entidad

Una *Entidad* es cualquier concepto del mundo real con una existencia independiente. Puede ser un objeto con una existencia física (ejemplo, máquina, robot) o puede ser un objeto con una existencia conceptual (p. ej.: Curso de universidad). Cada entidad tiene un conjunto de atributos que describen las propiedades de la entidad.

Nota: No existen notaciones estándar para representar los diagramas ER. Los diferentes textos sobre este asunto utilizan diferentes notaciones. Los conceptos y notaciones para los diagramas EER utilizados en Umbrello provienen del siguiente libro: Elmasri R. y Navathe S. (2004). *Fundamentals of Database Systems* 4ª ed. Addison Wesley (Fundamentos de los sistemas de bases de datos)

En un diagrama ER, las entidades se representan como rectángulos, con el nombre de la clase, y también pueden mostrar atributos y operaciones de la clase en otros dos «compartimentos» dentro del rectángulo.

Representación visual de una entidad en un diagrama ER

Atributos de la entidad

En los diagramas ER, los atributos de la entidad se muestra con su nombre en un compartimento diferente de la entidad a la que pertenecen.

Restricciones

Las restricciones en los diagramas ER especifican las restricciones de los datos en el esquema de información.

Existen cuatro tipos de restricciones soportadas por Umbrello:

- *Clave primaria:* El conjunto de atributos declarados como *clave primaria* es única para la entidad. Solo puede haber una clave primaria en una entidad y ninguno de los atributos que la componen puede ser NULL.
- *Clave única:* El conjunto de atributos declarados como *única* son únicos para la entidad. Pueden haber muchas restricciones únicas en una entidad. Los atributos que lo componen pueden tener el valor NULL. Las claves únicas y

primarias identifican de forma única una fila de una tabla (entidad)

- *Clave externa:* Una clave externa es una restricción referencia entre dos tablas. La clave externa identifica una columna o un conjunto de columnas en un tabla (referenciada) que referencia una columna o conjunto de columnas en otra tabla (referenciada). Las columnas en la tabla referenciada deben formar una clave primaria o una clave única.
- *Restricción de comprobación:* Una restricción de comprobación (también conocida como restricción de comprobación de tabla) es una condición que define los datos válidos cuando se añaden o actualizan datos en una tabla de la base de datos relacional. Se aplicará una restricción a cada fila de la tabla. La restricción debe ser un predicado. Puede referirse a una o varias columnas de la tabla.

Ejemplo: precio ≥ 0

Conceptos del diagrama de relaciones de entidades extendido (EER)

Especialización

La especialización es una manera de formar nuevas entidades utilizando entidades que ya se hayan definido. Las entidades nuevas, conocidas como entidades derivadas, asumir (o heredar) atributos de las entidades que ya existían, y que se refieren a las entidades base. Se pretende ayudar a reutilizar datos con pequeñas o ninguna modificaciones.

En Umbrello, se puede especificar la especialización de separación y de solapamiento

Especialización disjunta

Una especialización disjunta especifica que las subclases de una especialización deben ser disjuntas, es decir, una entidad puede ser miembro, como máximo, de una de las derivadas en la especialización.

Representación visual de una especialización disjunta en un diagrama ER

Especialización de solapamiento

Cuando las entidades derivadas no son obligatoriamente disjuntas, el conjunto de entidades se denomina una «especialización por solapamiento», lo que significa que una

entidad puede pertenecer a más de una entidad derivada de la especialización.

Representación visual de una especialización de solapamiento en un diagrama ER

Categoría

Una entidad derivada se considera una *Categoría* cuando representa una colección de objetos que es un subconjunto de la unión de varios tipos de entidades. Una categoría se modela cuando se necesita una relación única superclase/subclase con más de una superclase, donde la superclase representa diferentes tipos de entidades (similar a la herencia múltiple en Programación Orientada a Objetos).

Representación visual de una categoría en un diagrama ER

Capítulo 3. Trabajando con Umbrello UML Modeller

Tabla de contenidos

Interfaz de usuario

Título

Ventana de documentación

Área de trabajo

Crear, cargar y guardar proyectos

Nuevo proyecto

Guardar modelo

Cargar modelo

Editar modelo

Añadir y eliminar diagramas

Crear diagramas

Eliminar diagramas

Renombrar diagramas

Editar diagramas

Insertar diagramas

Borrar elementos

Editar elementos

Editar clases

Asociaciones

Notas, textos y cajas

Este capítulo presentará el interfaz de usuario de Umbrello UML Modeller y le informará de todo lo que necesita saber para iniciar su primer esquema. Todas las acciones de Umbrello UML Modeller

son accesibles a través del menú y de las barras de herramientas, pero Umbrello UML Modeller también hace un constante uso del botón derecho del ratón. Puede utilizar el botón derecho en la práctica totalidad de los elementos del área de trabajo o de la vista en árbol de Umbrello UML Modeller para abrir un menú con las funciones más útiles aplicables al elemento en particular sobre el que está trabajando. Algunos usuarios encuentran el manejo de estos menús un tanto confuso inicialmente, porque están más acostumbrados a trabajar con el menú o las barras de herramientas, pero una vez que se acostumbre al botón derecho, verá que puede acelerar enormemente su trabajo.

Interfaz de usuario

La ventana principal de Umbrello UML Modeller está dividida en tres áreas que le ayudarán a mantener una visión general de todo el sistema y a acceder rápidamente a los diferentes diagramas mientras trabaja en su proyecto.

Esas áreas reciben el nombre de:

- Vista en árbol
- Área de trabajo
- Ventana de documentación

Interfaz de usuario de Umbrello UML Modeller

Título

La vista en árbol está situada en la parte superior izquierda de la ventana, muestra todos los diagramas, clases, actores y casos de uso de los que está compuesto su esquema. Le permite tener una perspectiva de los elementos que componen su esquema. La vista en árbol también le proporciona una forma rápida de pasar de un diagrama a otro de su esquema así como de introducir elementos de su esquema en el diagrama actual.

Si está trabajando en un modelo con bastantes clases y diagramas, la vista en árbol le puede ayudar a controlarlo todo organizando los elementos de su esquema en carpetas. Puede crear nuevas carpetas seleccionando la opción adecuada en el menú contextual (botón derecho

Ventana de documentación

La ventana de documentación es esa ventana pequeña situada al fondo a la izquierda de Umbrello UML Modeller, le permite previsualizar rápidamente la documentación para el objeto seleccionado. Esta ventana es bastante pequeña debido a que su propósito es darle una rápida noción del elemento en cuestión sin acaparar mucho espacio en la pantalla. Si desea ver la

documentación más detalladamente puede abrir las propiedades del objeto.

Área de trabajo

El área de trabajo es la ventana principal de Umbrello UML Modeller y donde todo se lleva a cabo la parte importante del trabajo. Aquí es donde editará y verá los diagramas de su esquema. El área de trabajo muestra el diagrama sobre el que se está trabajando en cada momento, sólo es posible mostrar uno cada vez.

Crear, cargar y guardar proyectos

Lo primero que necesita para empezar a hacer cosas interesantes en Umbrello UML Modeller es crear un esquema sobre el que trabajar. Cuando se inicia Umbrello UML Modeller, siempre se carga el último esquema sobre el que se ha estado trabajando o crea uno vacío (según la configuración establecida en preferencias). Esto le permitirá empezar a trabajar inmediatamente.

Nuevo proyecto

Si en algún momento necesita crear un nuevo esquema, hágalo seleccionando (**NuevoArchivo** o pinchando sobre el icono Nuevo en la barra de herramientas. Si está trabajando en un esquema que ha sido modificado, Umbrello UML Modeller le preguntará si quiere o no guardar sus cambios antes de cargar el nuevo esquema.

Guardar modelo

Puede guardar su esquema en cualquier momento seleccionando **Guardar** desde el menú **Archivo** o pulsando sobre el botón Guardar en la barra de herramientas de la aplicación. Si necesita guardar su modelo con un nombre diferente puede usar la opción **Guardar como** desde el menú **Archivo**.

Umbrello UML Modeller también le permite ir guardando su trabajo automáticamente cada cierto tiempo. Puede indicar si desea o no esta opción así como el intervalo de tiempo en **preferencias**.

Cargar modelo

Para cargar un esquema ya existente puede seleccionar la opción **Abrir** desde el menú **Archivo** o pulsar sobre el icono Abrir de la barra de herramientas. Puede acceder a los esquemas usados recientemente en el submenú **Abrir reciente** en el menú **Archivo**.

Umbrello UML Modeller sólo puede trabajar en un sólo esquema al mismo tiempo, esto hace que si intenta cargar un nuevo esquema en el programa cuando ha realizado alguna modificación sobre el que está trabajando, Umbrello UML Modeller le preguntará si quiere guardar sus cambios antes de cerrarlo y abrir el nuevo. Puede iniciar dos o más instancias de

Umbrello UML Modeller al mismo tiempo, también puede copiar y pegar entre dos instancias.

Editar modelo

En Umbrello UML Modeller existen básicamente dos formas de editar los elementos de su modelo.

- Editar los elementos del esquema directamente a través de la vista en árbol.
- Editarlos a través de un diagrama.

Usando el menú contextual de los distintos elementos de la vista en árbol podrá añadir, eliminar y modificar casi todos los elementos de su esquema. Pinchando con el botón derecho del ratón sobre las carpetas en la vista en árbol también podrá crear diferentes tipos de diagramas dependiendo si la carpeta en cuestión es *vista de caso de uso* o *vista lógica*, actores, casos de uso, clases, etc...

Una vez que ha añadido algún elemento a su modelo, podrá editarlo mediante su diálogo de propiedades al que podrá acceder seleccionando la opción *Propiedades* en el menú contextual que aparece al pinchar con el botón derecho en los objetos de la vista en árbol.

También puede editar su esquema creando o modificando sus elementos por medio de los diagramas. En la siguiente sección tiene más detalles de como hacer esto.

Añadir y eliminar diagramas

Su esquema UML está formado por un conjunto de elementos de UML y las asociaciones entre ellos. Dado que no es posible ver el esquema directamente, se usan *diagramas* para ello.

Crear diagramas

Para crear un nuevo diagrama en su esquema seleccione el tipo de diagrama que necesita en **Nuevo** del menú **Diagrama** y asígnele un nombre. Se creará el diagrama y podrá verlo en la vista de árbol.

Recuerde que Umbrello UML Modeller emplea a menudo menús contextuales, también puede pulsar botón derecho sobre una carpeta de la vista en árbol y seleccionar el tipo de diagrama que desee desde la opción **Nuevo** del menú contextual. Observe que puede crear diagramas de caso de uso sólo desde las carpetas de vista de casos de uso y los otros tipos de diagramas sólo pueden ser creados en las carpetas de vista lógica.

Eliminar diagramas

Si desea eliminar un diagrama de su esquema, puede hacerlo activándolo y seleccionando **Borrar** desde el menú **Diagrama**. También puede eliminarlo seleccionado **Borrar** desde el menú contextual del diagrama en la vista en árbol.

Dado que el borrado de diagramas es algo delicado porque se puede perder mucho trabajo si se hace accidentalmente, Umbrello UML Modeller le pedirá confirmación antes de eliminar cualquier diagrama. Una vez que un diagrama ha sido borrado y el archivo ha sido guardado, ya no será posible recuperarlo.

Renombrar diagramas

Si quiere cambiar el nombre de un diagrama ya existente, puede hacerlo seleccionando Renombrar desde el menú que le aparece al botón derecho sobre la vista de árbol.

Otra forma de renombrar un diagrama es hacerlo a través de su diálogo de propiedades que puede ver seleccionando 'Propiedades' desde el menú contextual o haciendo doble click sobre la vista de árbol.

Editar diagramas

Cuando esté trabajando en un diagrama, Umbrello UML Modeller tratará de ayudarle indicándole algunas sencillas reglas sobre qué elementos son válidos en los distintos tipos de diagramas así como las relaciones que pueden existir entre ellos. Si usted es un experto en UML seguramente ni se dé cuenta, pero si está empezando le ayudará a crear correctamente diagramas que cumplan el estándar.

Una vez que ha creado sus diagramas, es hora de editarlos. Algún novato observador habrá observado la diferencia entre editar un diagrama y editar el *esquema*. Como ya sabrá, los diagramas son *vistas* de su esquema. Por ejemplo, si crea una clase editando un diagrama de clases estará editando el diagrama y el modelo, pero si cambia el color o otra opción visual en un diagrama de clases, estará editando sólo el diagrama sin modificar el esquema.

Insertar diagramas

Una de las primeras cosas que hará cuando edite un nuevo diagrama es insertar elementos en ellos (clases, actores, casos de uso, etc.). Existen básicamente dos formas de hacer esto:

- Arrastrando elementos ya existentes en su esquemas desde la vista en árbol.
- Crear nuevos elementos en su modelo y añadirlos a su diagrama al mismo tiempo usando alguna de las herramientas de edición de la barra de herramientas.

Para introducir elementos ya existentes en su esquema, simplemente arrástrelos desde la vista en árbol y suéltelos en el lugar del diagrama donde quiere situarlos. Siempre podrá mover elementos en los diagramas empleando la herramienta 'Seleccionar'.

la segunda forma de añadir elementos a su diagrama es usando las herramientas de edición de la barra de herramientas

principal (observe que esto también añadirá los elementos a su esquema).

La barra de herramientas principal está situada, por omisión, en la parte derecha de la ventana de la aplicación. Desde Umbrello UML Modeller 1.2 se sitúa en la parte superior de la ventana. Puede situarla en cualquier otra parte o anclarla en otro punto. Las herramientas que contiene (es decir, los botones que ve sobre ella) cambiarán según el modelo de diagrama sobre el que esté trabajando en cada momento. El botón de la herramienta que está seleccionada en cada momento aparece activado. Podrá pasar a la herramienta de selección pulsando **Esc**.

Una vez que ha seleccionado una herramienta de edición en la barra de herramientas de trabajo (por ejemplo la herramienta para insertar clases) el puntero del ratón adopta forma de cruz, ahora puede insertar elementos en su esquema haciendo click en su diagrama. Observe que los elementos en UML deben tener un *nombre único* así que si tiene una clase llamada "claseA" en un diagrama y utiliza la herramienta insertar para introducir otra clase en otro diagrama, no podrá llamarla "claseA". Dado que se supone que se trata de elementos diferentes, sus nombres también deberán serlo. Si lo que quiere hacer es añadir el *mismo* elemento en su diagrama la herramienta insertar clase no es adecuada para esto, lo que debe hacer es arrastar y soltar la clase desde la vista en árbol.

Borrar elementos

Podrá borrar cualquier elemento seleccionando la opción **Borrar** desde su menú contextual.

De nuevo hay una *gran* diferencia entre eliminar un objeto de un diagrama y eliminarlo de todo el esquema. Si borra un objeto de un diagrama, únicamente lo eliminará de ese diagrama concreto, seguirá formando parte de su esquema y si otros diagramas lo usan seguirá estando ahí. En cambio, si borra el elemento desde la vista en árbol sí que lo eliminará completamente de su *esquema*, con lo que desaparecerá de todos los diagramas donde aparecía.

Editar elementos

Puede editar la mayoría de los elementos de UML de sus esquemas y diagramas abriendo su diálogo de propiedades y seleccionando las opciones pertinentes. Para editar las propiedades de un objeto, seleccione **Propiedades** desde su menú contextual (haciendo botón derecho). Cada elemento tiene un diálogo de varias páginas donde configurar las opciones correspondientes a estos elementos. En algunos elementos, como los actores, sólo disponen de un par de opciones como el nombre del objeto y su documentación mientras que para otros como las clases puede editar sus atributos y operaciones, seleccionar lo que quiere que se vea en el diagrama, incluso los colores.

Para la mayoría de los elementos UML, si usa la herramienta selección (la flecha) puede abrir el diálogo de propiedades haciendo doble click sobre ellos. La excepción a esto son las asociaciones en cuyo caso un doble click crearía un punto de anclaje, para ello tiene que emplear el menú contextual (haciendo botón derecho) e ir al diálogo de propiedades.

Observe que también puede seleccionar la opción de propiedades desde el menú contextual de los elementos en la vista en árbol. Esto también le permite editar las propiedades de los diagramas como seleccionar si la rejilla debe o no verse.

Editar clases

Aunque la edición de propiedades de todos los objetos ya haya sido tratada en la sección anterior, las clases merecen una explicación adicional debido a su mayor complejidad y a que poseen más opciones que la mayoría de los elementos de UML.

En el diálogo de propiedades de una clase es posible configurar cualquier parámetro, desde el color que emplea hasta sus atributos y operaciones.

Preferencias generales de clase

La pestaña de preferencias generales del diálogo de propiedades se explica por sí mismo. Desde ahí podrá cambiar el nombre de la clase, la visibilidad, la documentación, etc.. Esta pestaña siempre está disponible.

Configuración de los atributos de las clases.

En la configuración de atributos podrá añadir, editar o borrar atributos (variables) de una clase. Puede mover atributos arriba y abajo en la lista pulsando la flecha del lateral de la ventana. Esta pestaña siempre está disponible.

Configuración de operaciones de clase

De modo similar a la página de configuración de atributos, aquí podrá añadir, editar y borrar operaciones de su clase. Cuando añada o edite una operación, deberá insertar la información básica en el diálogo *Propiedades de operaciones*. Si lo que desea es añadir parámetros a su operación deberá pulsar sobre **Nuevo parámetro** para que se muestre el diálogo *Propiedades del parámetro*. Esta página siempre está disponible.

Configuración de la plantilla de clase

Desde ahí podrá añadir plantillas de clase, es decir clases no específicas o tipos de datos. En Java 1.5 recibirán el nombre de genéricas.

Página de asociaciones de clase

La página de **Asociaciones de clase** muestra todas las asociaciones de dicha clase en el diagrama actual. Si hace doble clic sobre una asociación podrá ver sus propiedades y, según su tipo, modificar algunos parámetros como el nombre del rol o incluir multiplicidad. Si la asociación no permite que estas opciones sean modificadas, el diálogo de propiedades de asociación será de solo lectura y solo podrá modificar la documentación asociada con esta asociación.

Esta página sólo estará disponible si abre las propiedades de clase desde un diagrama. Si selecciona las propiedades de clase en un menú contextual desde la vista en árbol, esta página no estará disponible.

Página de visualización de clase

En la página de **opción de visualización** podrá definir los elementos que deben mostrarse en un diagrama. Una clase podrá mostrarse simplemente como un rectángulo con su nombre escrito en él (especialmente útil si tiene varias clases o si, de momento, no está interesado en los detalles de cada clase) o bien mostrar completamente todos los paquetes, estereotipos, atributos y operaciones.

Dependiendo de la cantidad de información que desee ver, podrá seleccionar las correspondientes opciones en esa página. Los cambios que realice ahí sólo será *opciones de visualización* para el diagrama. Esto quiere decir que “esconder” la operación de una clase sólo hará que no se muestren en el diagrama, pero la operación seguirá ahí formando parte de su esquema. Esta operación sólo estará disponible si selecciona las propiedades de clase desde un diagrama. Si abre las propiedades de clase desde la vista en árbol esta página no estará disponible ya que esta opción de visualización carece de sentido en este contexto.

Página para configurar los colores de la clase

En la página **color de objetos** podrá configurar los colores que desee para las líneas y el relleno de los objetos. Obviamente, esta opción sólo tiene sentido para las clases que se muestran en los diagramas y no aparece cuando abre el diálogo de propiedades de clase desde la vista en árbol.

Asociaciones

Las asociaciones relacionan dos objetos UML entre sí. Normalmente, las asociaciones se definen entre dos clases, sin embargo algunos tipos de asociación también pueden darse entre casos de uso y actores.

Para crear una asociación seleccione la herramienta adecuada en la barra de herramientas de trabajo (asociación genérica,

generalización, agregación, etc.) y pinche primero sobre el primer elemento de la asociación y luego sobre el segundo. Observe que lo que hacemos son dos clicks, *no* arrastrar el click de un objeto a otro.

Si intenta crear una asociación que no se ajuste a las especificaciones de UML, Umbrello UML Modeller no aceptará crearla y le mostrará un mensaje de error. Un ejemplo de esto sería si existiera una generalización desde la clase A hasta la B y tratase de crear otra desde B hasta A.

Pulsando con el botón derecho del ratón sobre una asociación verá un menú contextual con las acciones que pueden realizarse. Si quiere borrar una asociación, simplemente seleccione la opción **Borrar** en ese menú contextual. También puede seleccionar la opción **propiedades** y, según el tipo de asociación, editar los atributos que posea.

Puntos de anclaje

por omisión, las asociaciones se representan mediante una línea recta que conecta los dos objetos de un diagrama.

Podrá añadir puntos de anclaje haciendo doble click en cualquier parte de la línea de asociación. Estos puntos de anclaje se representan mediante un punto azul cada vez que se selecciona la línea, podrá moverlo a su antojo hasta dar la forma deseada a la (ex)recta de asociación.

Si desea eliminar puntos de anclaje, haga doble click sobre ellos. Observe que el único modo de editar las propiedades de una asociación es a través del menú contextual. Si trata de hacer doble click sobre él como haría con cualquier otro objeto UML, se insertará un punto de anclaje.

Notas, textos y cajas

Las notas, textos y cajas pueden aparecer en cualquier tipo de diagrama, carecen de valor semántico pero son útiles para añadir comentarios o explicaciones adicionales para facilitar la comprensión del diagrama.

Para añadir una nota o texto debe seleccionar la herramienta correspondiente de la barra de herramientas y pinchar sobre el punto del diagrama donde desee añadirlo. Puede editar el texto a través de su menú contextual o, en el caso de las notas, haciendo doble click sobre él.

Anclajes

Los anclajes sirven para unir una nota con otro elemento UML. P. ej. si suele usar una determinada nota para explicar o realizar algún comentario sobre una clase o una asociación en concreto, puede usar los anclajes para dejar claro que la nota “pertenece” a esa clase o asociación y no a otras.

Para anclar una nota con otro elemento UML, utilice la herramienta anclaje de la barra de herramientas de trabajo. Primero deberá pinchar sobre la nota y luego sobre el elemento UML al que quiere asociar la nota.

Capítulo 4. Importación y generación de código

Tabla de contenidos

Generación de código

Generación de código

Importación de código

Umbrello; es una herramienta de modelado UML, y como tal, su propósito principal es el de ayudar en el *análisis* y *diseño* de sus sistemas. Sin embargo, para realizar la transición entre el diseño y la *implementación*, Umbrello UML Modeller permite la generación de código fuente en diferentes lenguajes de programación como ayuda inicial. Además, si desea utilizar UML en un proyecto C++ ya iniciado, Umbrello UML Modeller puede ayudarle a crear una maqueta de su sistema a partir del código fuente realizando un análisis del código e importando las clases encontradas en él.

Generación de código

Umbrello UML Modeller puede generar código fuente en varios lenguajes de programación, a partir de la maqueta UML para ayudar a comenzar la implementación de su proyecto. El código generado consta de declaraciones de clases con sus métodos y atributos, de forma que usted pueda “rellenar los espacios en blanco” proporcionando la funcionalidad de las operaciones de sus clases.

Umbrello UML Modeller 1.2 viene provisto con soporte para la generación de código en ActionScript, Ada, C++, CORBA IDL, Java™, JavaScript, PHP, Perl, Python, SQL y XMLSchema.

Generación de código

Para generar código con Umbrello UML Modeller, en primer lugar hay que crear o cargar una maqueta que contenga al menos una clase. Una vez que este listo para comenzar a escribir el código, seleccione el **Asistente de generación de código** en el menú **Código** para iniciar un asistente que le guiará a través del proceso de generación de código.

El primer paso es seleccionar las clases para las que desea generar código fuente. De forma predeterminada se seleccionan todas las clases de la maqueta, y usted puede eliminar aquellas para las que no desea generar código eliminándolas de la lista de la izquierda.

El siguiente paso del asistente le permite modificar los parámetros que el generador de código utilizará en el proceso. Las opciones disponibles son las siguientes:

Opciones de generación de código en Umbrello UML Modeller

Opciones de generación

Comentado del Código

La opción **Escribir comentarios de documentación aún si está vacía** instruye al Generador de código a escribir comentarios del tipo `/** blah */` aún cuando el bloque de comentarios se encuentre vacío. Si usted ya agregó documentación a sus clases, métodos o atributos en su Maqueta, el Generador de código escribirá estos comentarios como documentación Doxygen sin importar lo que haya definido aquí, pero si selecciona esta opción Umbrello UML Modeller escribirá comentarios para todas sus clases, métodos y atributos aún cuando no exista documentación en la Maqueta, en cuyo caso usted deberá documentar sus clases más tarde editando directamente el código fuente.

Escribir comentarios para las secciones incluso si la sección está vacía producirá que Umbrello UML Modeller escriba comentarios en el código fuente para delimitar las diferentes secciones de una clase. Por ejemplo “Métodos públicos” o “Atributos ” antes de la sección correspondiente. Si selecciona esta opción Umbrello UML Modeller escribirá los comentarios para todas las secciones de las clases aún si la sección está vacía. Por ejemplo, escribirá un comentario diciendo “Métodos protegidos” aún si no existen métodos protegidos en su clase.

Carpetas

Escribir todos los archivos generados a la carpeta. Aquí debe seleccionar la carpeta donde desea que Umbrello UML Modeller guarde los archivos fuente generados.

La opción **Incluir archivos de cabecera de la carpeta** le permite insertar un encabezado al comienzo de cada archivo generado. Los archivos de encabezado pueden contener información sobre el copyright o la licencia, y contener variables evaluadas en tiempo de generación. Puede consultar a modo de ejemplo los archivos de encabezado distribuidos con Umbrello UML Modeller para ver cómo se usan estas variables para reemplazar su nombre o la fecha y hora del momento de generación.

Política de sobreescritura

Esta opción le dice a Umbrello UML Modeller qué hacer si el archivo que desea crear ya existe en la carpeta de destino. Umbrello UML Modeller *no puede modificar archivos fuente*

existentes, por lo que debe elegir entre sobreescribir el existente, saltar la generación de ese archivo en particular, o dejar que Umbrello UML Modeller elija un nombre diferente para el archivo. Si usted elige la opción de utilizar un nombre diferente, Umbrello UML Modeller agregará un sufijo al nombre del archivo.
Lenguaje

Umbrello UML Modeller generará por omisión código fuente en el lenguaje que usted haya seleccionado como Lenguaje activo, pero con el Asistente para la generación de código usted tiene la opción de cambiar esto a otro idioma.

Asistente para la generación de código

El tercero y último paso del asistente le mostrará el estado del proceso de generación de código. Usted necesitará hacer click en el botón de Generación de código para que las clases sean escritas para usted.

Recuerde que las opciones que seleccione en el asistente de generación de código sólo son válidas para esa generación en concreto, la siguiente vez que ejecute el asistente deberá volver a seleccionar todas las opciones aunque sean las mismas. Puede evitar esto definiendo en la sección **Generación de código** del menú **Preferencias** -> **Configurar modelador de UML Umbrello UML Modeller...**

Si ha establecido las opciones de la generación de código y quiere generar directamente algún código sin el asistente, seleccione **Generar todo el código** desde el menú **Código**. Esto generará código para todas las clases de su esquema con la configuración actual (incluyendo la carpeta de salida y las políticas de sobreescritura, así que tenga cuidado).

Importación de código

Umbrello UML Modeller puede importar código fuente de sus proyectos actuales para ayudarle a crear los esquemas de sus sistemas. Umbrello UML Modeller 1.2 sólo puede hacerlo con C++ aunque en el futuro estará disponible para más lenguajes.

Para importar clases en su esquema seleccione la entrada **Importar clases** desde el menú **Código**. Una vez ahí, seleccione los archivos que contengan las declaraciones de clases de C++ y pulse aceptar. Se importarán las clases y pasarán a ser parte de su esquema en la vista en árbol. Recuerde que Umbrello UML Modeller no creará ningún tipo de diagrama para mostrar sus clases, sólo las importará a su esquema para que pueda usarlas luego en cualquier diagrama.

Menú para la importación de código fuente en Umbrello UML Modeller

Capítulo 5. Otras características

Tabla de contenidos

[Otras características de Umbrello UML Modeller](#)

[Copia de objetos como imágenes PNG](#)

[Exportar como imagen](#)

[Impresión](#)

[Carpetas lógicas](#)

Otras características de Umbrello UML Modeller

Este capítulo explica brevemente otras características que ofrece Umbrello UML Modeller.

Copia de objetos como imágenes PNG

Además de ofrecer la funcionalidad normal de copiar, cortar y pegar que se espera en la administración de los objetos de los diferentes diagramas, Umbrello UML Modeller puede copiar los objetos como imágenes PNG de forma que pueda insertarlos en cualquier otro tipo de documento. No es necesario hacer nada especial para utilizar esta característica, basta con que seleccione un objeto de un diagrama (clase, actor, etc.) y lo copie (**Ctrl-C** o mediante el menú), abra un documento de KWord (o cualquier otro programa que admita el pegado de imágenes) y seleccione la opción de [pegar](#). Esta es una gran característica para exportar partes del diagrama como simples imágenes.

Exportar como imagen

También puede exportar como imagen un diagrama completo. Lo único que tiene que hacer es seleccionar el diagrama que desea exportar y a continuación la opción [Exportar como imagen...](#) del menú [Diagrama](#).

Impresión

Umbrello UML Modeller le permite imprimir diagramas individuales. Pulse el botón Imprimir en la barra de herramientas de aplicación o seleccione la opción [Imprimir](#) en el menú [Archivo](#) para obtener un diálogo de impresión estándar de KDE donde podrá imprimir los diagramas.

Carpetas lógicas

Para organizar mejor la maqueta, especialmente en los proyectos grandes, es posible crear carpetas lógicas en la vista de árbol. Basta con que seleccione la opción [Nueva->Carpetas](#) en el menú contextual de la carpeta predeterminada en la vista de árbol. Las carpetas se pueden anidar, puede mover objetos de una a otras por el procedimiento normal de arrastrar y soltar.

Organización de una maqueta con carpetas lógicas en Umbrello UML Modeller

Capítulo 6. Autores e historia

Este proyecto fue iniciado por Paul Hensgen como uno de sus proyectos universitarios. El nombre original de la aplicación era *UML Modeller*. Paul se encargó de todo el desarrollo hasta finales de 2001, cuando el programa llegó a la versión 1.0.

La versión 1.0 ya ofrecía muchas funcionalidades, pero una vez que el proyecto fue revisado en la universidad de Paul, otros desarrolladores pudieron unirse al equipo y comenzar a enviar valiosas contribuciones a UML Modeller, como el cambio de un archivo binario a uno , sXMLoporte para más tipos de diagramas UML, generación de código e importación de código por nombrar algunas.

Paul tuvo que retirarse del equipo de desarrollo en verano de 2002 pero, como software libre que es, el programa continúa mejorando y evolucionando, y es mantenido por un grupo de desarrolladores de diferentes lugares del mundo. Además, en septiembre de 2002, el proyecto cambió el nombre de UML Modeller a Umbrello UML Modeller. Existen varias razones para el cambio del nombre, siendo la más importante que sólo “uml” (como era conocido) —resultaba un nombre muy genérico— y causaba problemas con algunas distribuciones. La otra razón importante es que los desarrolladores piensan que *Umbrello* es un nombre más impactante.

El desarrollo de Umbrello UML Modeller así como las discusiones sobre el futuro del programa son abiertas y tiene lugar en Internet. Si desea contribuir con el proyecto, no dude en ponerse en contacto con los desarrolladores. Hay muchas formas para ayudar en el desarrollo de Umbrello UML Modeller

- Informando de fallos o sugerencias
- Arreglando fallos o añadiendo funcionalidad
- Escribiendo buena documentación o traduciendo a otros idiomas
- Y, por supuesto... programando con nosotros

Como puede ver, hay muchas formas en las que puede contribuir. Todas ellas son muy importantes y todo el mundo es bienvenido. Puede contactar con los desarrolladores de Umbrello UML Modeller en (uml-devel@lists.sourceforge.net).

Capítulo 7. Copyright

Copyright 2001, Paul Hensgen

Copyright 2002, 2003 Los desarrolladores de Umbrello UML Modeller . Vea <http://uml.sf.net/developers.php> para más información.

Esta documentación está sujeta a los términos de la Licencia de Documentación Libre GNU.

Este programa está sujeto a los términos de la Licencia Pública General GNU.