

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)



CoNcept

Lanzado TNTConcept versión 0.6 (12/07/2007)

Desde [Autentia](#) ponemos a vuestra disposición el software que hemos construido (100% gratuito y sin restricciones funcionales) para nuestra gestión interna, llamado TNTConcept (auTeNTia).

Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Acegi, Hibernate, Maven, Subversion, etc.) y disponible en licencia GPL, seguro que a muchos profesionales independientes y PYMES os ayudará a organizar mejor vuestra operativa.

Las cosas grandes empiezan siendo algo pequeño Saber más en: <http://tntconcept.sourceforge.net/>

 <p>Autor: Cristóbal González Almiron es consultor de desarrollo de proyectos informáticos.</p> <p>Su experiencia profesional se ha desarrollado en empresas como Compaq, HP, Mapfre, Endesa, Repsol, Universidad Autónoma de Madrid, en las áreas de Desarrollo de Software (Orientado a Objetos), tecnologías de Internet, Técnica de Sistemas de alta disponibilidad y formación a usuarios.</p> <p>Contacte con Cristóbal González criskerberos-tutoriales@yahoo.com</p>	<p>NUEVO CATÁLOGO DE SERVICIOS DE AUTENTIA (PDF 6,2MB)</p>  <p>www.adictosaltrabajo.com es el Web de difusión de conocimiento de www.autentia.com</p>  <p>autentia real business solutions</p> <p>Catálogo de cursos</p>
---	--

Descargar este documento en formato PDF [togetherArch.pdf](#)

[Firma en nuestro libro de Visitas <----->](#) [Asociarme al grupo AdictosAlTrabajo en eConozco](#)

Free UML 2.1 Design Tool

Visually develop applications with Roundtrip model to code, ERD & DB
www.visual-paradigm.com

Eclipse Tool for JSF

Evadial Studio Pro Download now!

Anuncios Google

Fecha de creación del tutorial: 2006-07-12

Creación y documentación de aplicaciones Java con Borland Together para Eclipse

[Creación y documentación de aplicaciones Java con Borland Together para Eclipse](#)

[Introducción](#)

[Instalación del programa](#)

[Iniciando el Together](#)

[Añadiendo casos de uso](#)

[Añadiendo un diagrama de secuencia](#)

[Diagramas de colaboración](#)

[Añadiendo diagramas de clases](#)

[Relaciones entre clases](#)

[Tipos de asociación](#)

[Diagramas de actividad](#)

[Añadiendo un diagrama de estado](#)

[Añadiendo diagramas de despliegue](#)

[Generación del código asociado](#)

Introducción

En nuestro trabajo diario de desarrollo en Java nos veremos obligados a hacer dos tareas, la propia de desarrollo y la tradicionalmente conocida como “empapelar”, es decir, generar la documentación del proyecto. Con las prisas de la entrega acabamos haciendo una documentación mediocre al final del proyecto. Lo ideal es generar la documentación del proyecto antes del desarrollo del proyecto, para que nos sirva de guía en nuestro desarrollo.

Con UML y las metodologías de análisis y diseño orientado a objetos podemos generar la documentación antes de comenzar, al menos la de análisis y diseño. Hay multitud de herramientas que nos permiten hacer esta

documentación. Pero las más modernas van un paso más allá: nos generan el esqueleto de la aplicación a partir del análisis, y sincronizan la documentación UML a medida que modificamos el proyecto.

Una de las herramientas líderes en este apartado es el Borland Together, que permite no sólo generar la documentación UML para un proyecto, sino que si disponemos del proyecto Java ya también la crea y la mantiene actualizada a medida que vamos cambiando el proyecto (ideal para reingenierías o entregas de documentación). Además podremos generar proyectos Java a partir de un proyecto UML (ideal para proyectos nuevos), con lo que todas las posibilidades están cubiertas.

En este tutorial vamos a aprender a crear un proyecto Java con su documentación en forma de diagramas UML.

Instalación del programa

Para instalar la versión de evaluación, se realiza en tres pasos:

1. Se obtiene la licencia de la página web de Borland. Por correo nos llegará un fichero TXT de registro.
2. Se instala el programa, con las opciones por defecto.
3. Se copia el fichero de registro a nuestra carpeta de usuario en c:\documents and settings\miusuario

La siguiente secuencia muestra la instalación. Se ha utilizado las opciones por defecto



Pulsamos “Next”



Pulsamos “Next”



“Pulsamos “Next”



Pulsamos “Next”



Pulsamos “Next”



Pulsamos "Install"



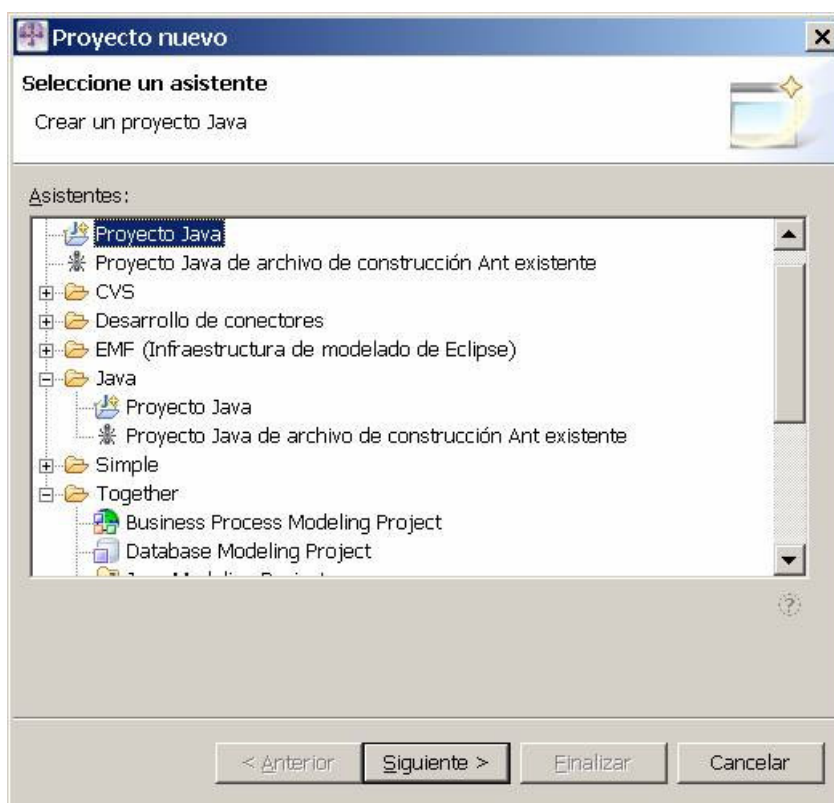
Tras el correspondiente copiado de ficheros, el sistema estará instalado.

Iniciando el Together

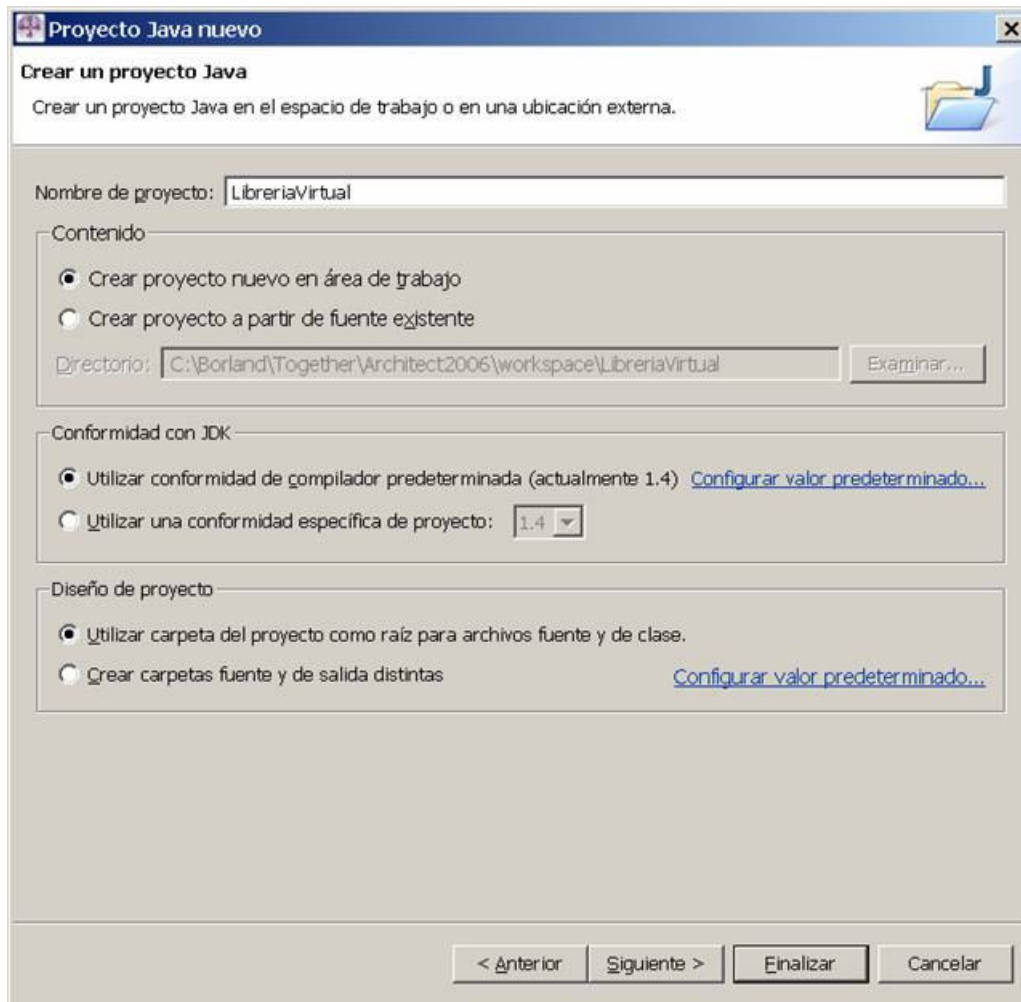
Arrancamos el Together y nos sale la típica ventana de Eclipse



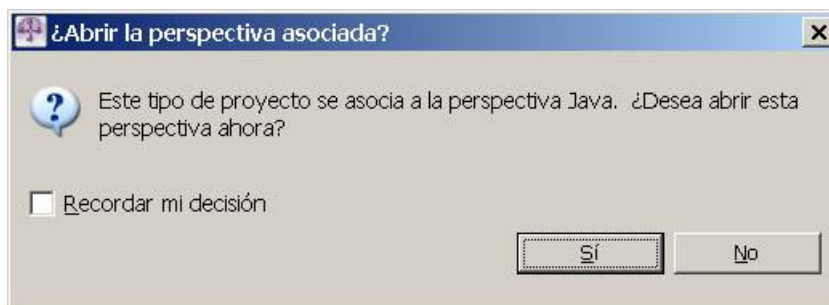
Ahora abrimos el menú “Archivo\nuevo proyecto...”



Dentro de los proyectos de Together podemos ver varios tipos de proyectos. Comenzamos por uno sencillo. Elegimos “Nuevo proyecto Java” y le ponemos un nombre al proyecto

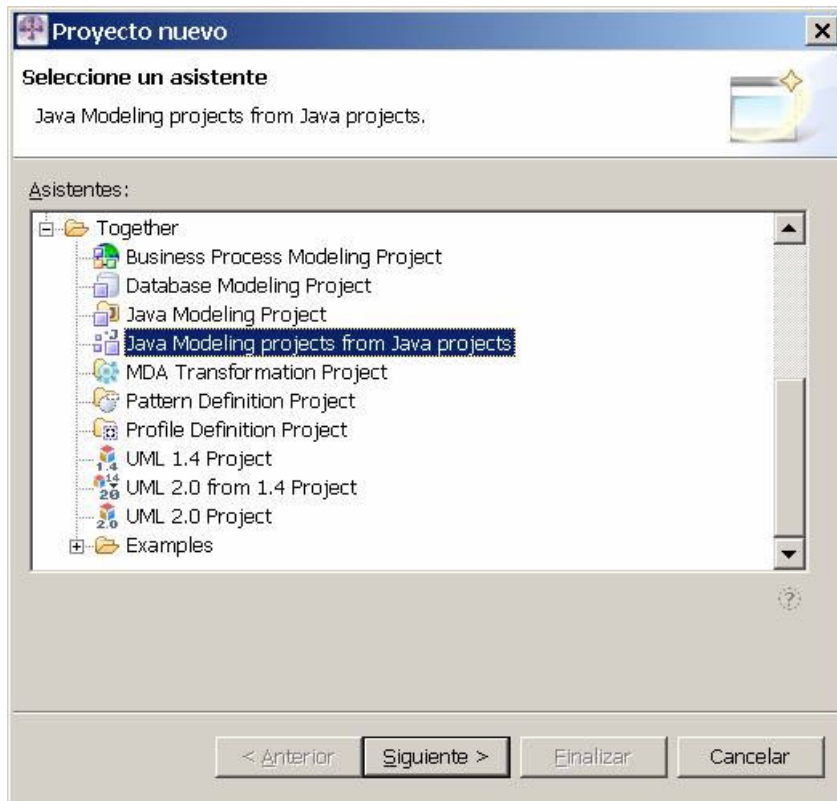


Le he dado un nombre al proyecto. Nos aparece la siguiente pantalla del Eclipse:

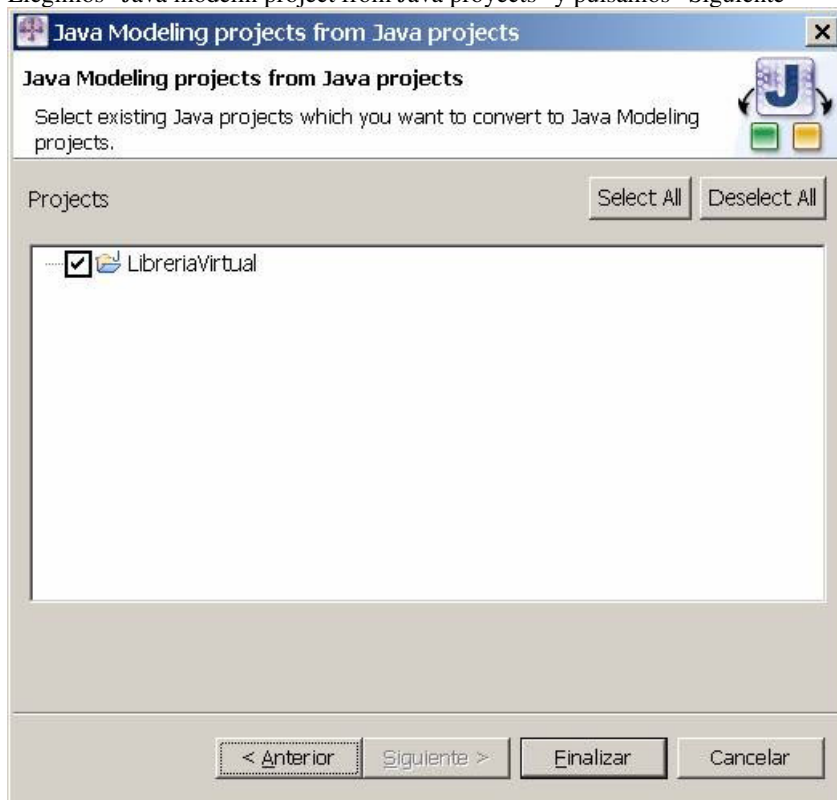


Pulsamos “Sí” para pasar a la perspectiva de Java.

Ahora seleccionamos nuestro proyecto y elegimos “Nuevo...\Proyecto...” con el botón derecho. Nos muestra un diálogo de proyectos.



Elegimos “Java modelin project from Java projects” y pulsamos “Siguiente”



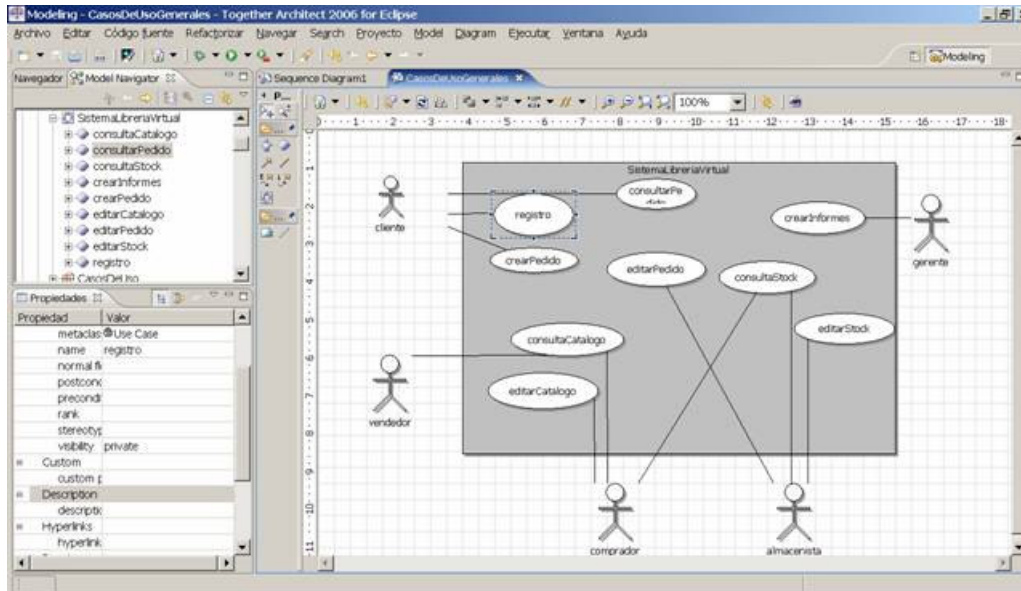
Elegimos nuestro proyecto de librería virtual y pulsamos “Finalizar”. Se nos ha añadido una perspectiva más a nuestro Eclipse, la de “Modeling”

Añadiendo casos de uso

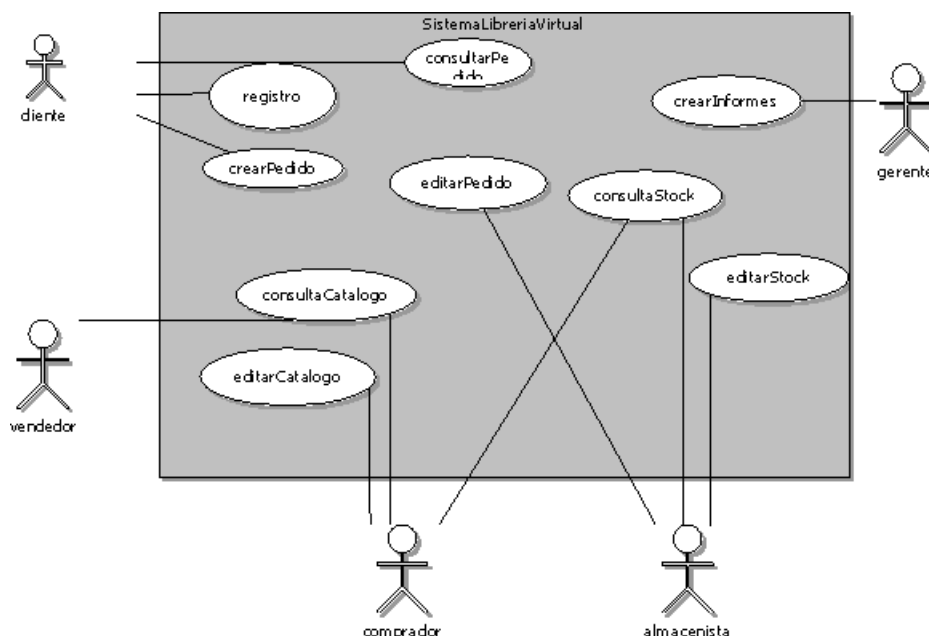
Para añadir un diagrama UML pulsamos con el botón derecho sobre el proyecto y elegimos la opción “New Diagram...”. Together nos da las siguientes opciones:



Elegimos nuevo “Use case”



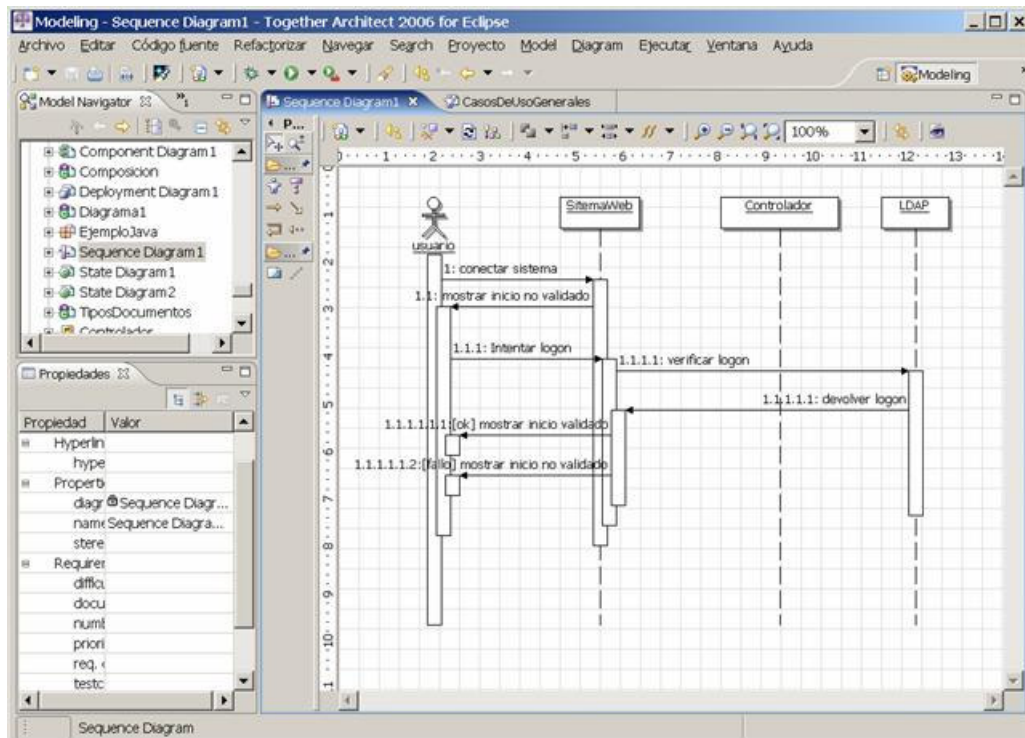
Le hemos añadido unos actores, un sistema y una serie de casos de uso. A la izquierda del diagrama tenemos los pequeños botones que sirven para añadirlos. El icono del hombrecito representa al actor, la caja gris de fondo al sistema y las elipses a los casos de uso. En detalle se puede ver:



Cada elemento que añadimos al diagrama genera un elemento en el panel de la izquierda. Así nos aparecerá un elemento comprador, un almacenista, etc.

Añadiendo un diagrama de secuencia

Pulsamos en “New Diagram\sequence” y rellenamos el diagrama

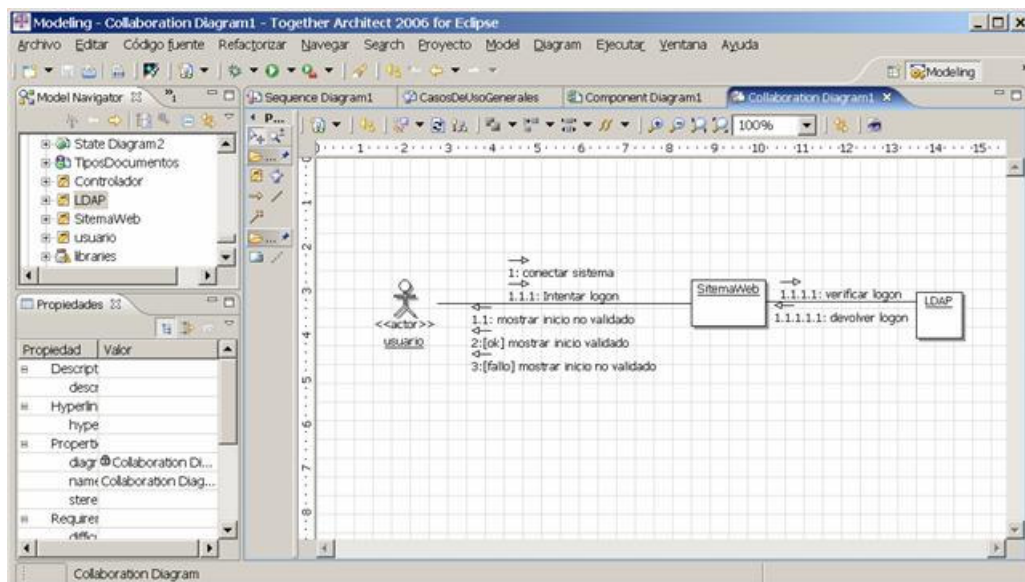


Hemos añadido un actor, y tres objetos más, el primero representa al sistema y los otros a dos controladores adicionales. En esta versión de Together no se permite cambiar los iconos de los objetos, por lo que serán siempre cuadrados.

Añadiendo las flechas de la secuencia, Together nos calculará automáticamente las cajas rectangulares que representan a los procesos. También añadirá los números de orden de las secuencias, que son números largos del tipo 1.1.2.1.

Diagramas de colaboración

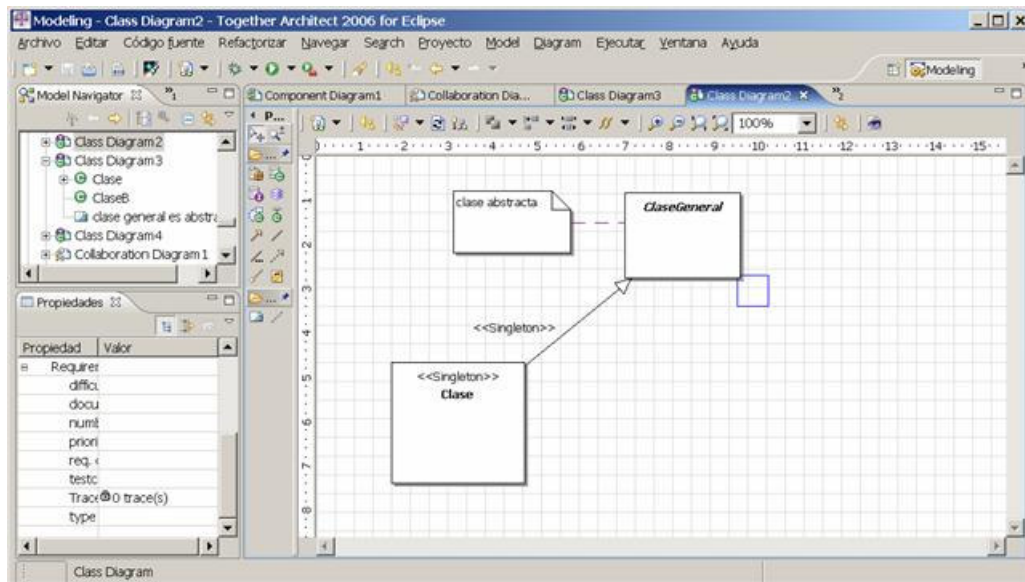
Una vez que tenemos el diagrama de secuencia, es fácil crear el de colaboración. Basta pulsar en “New Diagram\collaboration”



Al añadir el actor usuario, el sistema web y el servidor LDAP que habíamos definido en el diagrama de secuencia anterior, Together calculará automáticamente las flechas del diagrama. Esto es muy cómodo, pues basta con colocar adecuadamente los objetos para que Together ajuste las líneas y sean visibles.

Añadiendo diagramas de clases

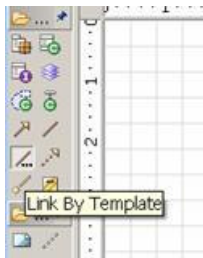
Pulsamos en “New diagram\class” y comenzamos a editar un diagrama de clases



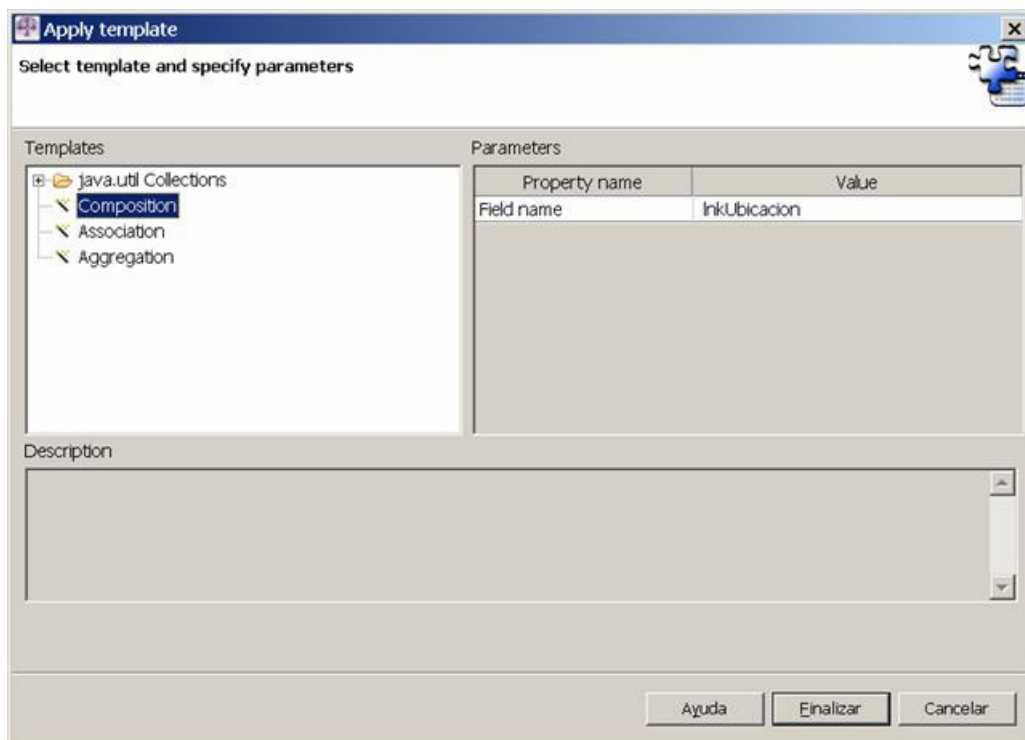
Relaciones entre clases

Together nos permite añadir relaciones entre clases de una manera sencilla.

Para crear estas asociaciones hay que elegir el icono “link by template”

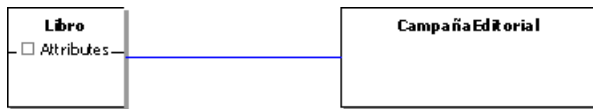


Esto nos abrirá el cuadro de diálogo que nos permite elegir los tipos de asociación.

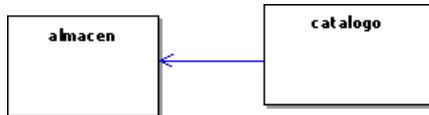


Tipos de asociación

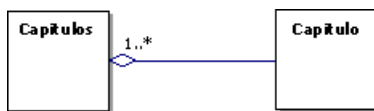
Una **asociación** se dibuja con una raya que une las clases. Los elementos para su dibujo como siempre están en los pequeños botones del panel de la izquierda.



Una **asociación unidireccional** se pinta mediante una línea dirigida. Para ello debemos pintar una línea normal y cambiar la propiedad de la línea “directed” a true.

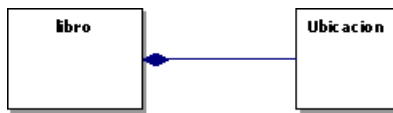


Una **agregación** se representa por una línea con un rombo blanco

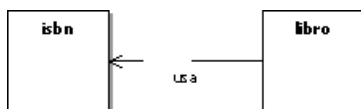


Las opciones de cardinalidad se fijan en las propiedades del enlace.

Una **composición** se representa por una línea con un rombo negro

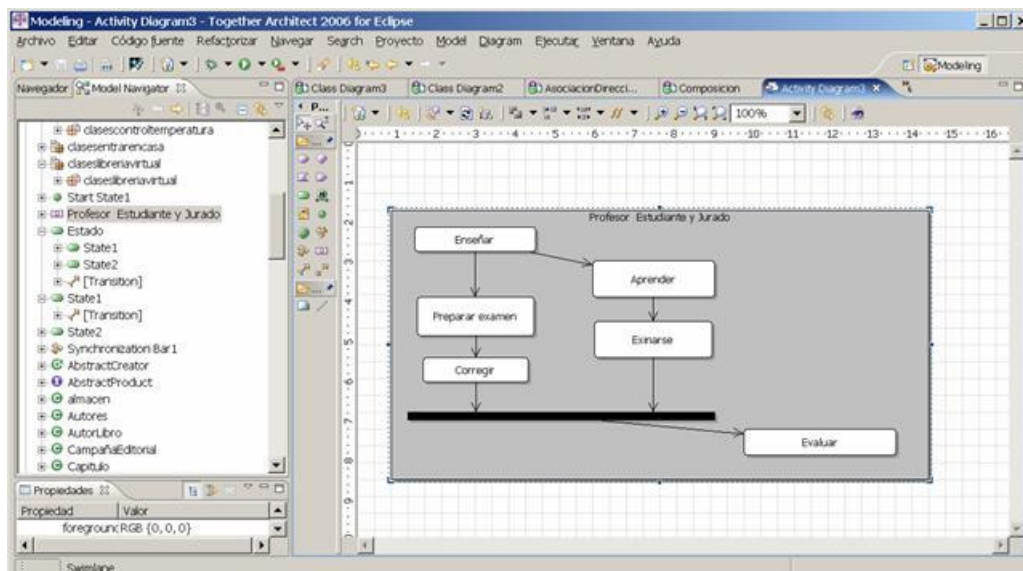


Una **asociación de dependencia** se representa por una flecha discontinua



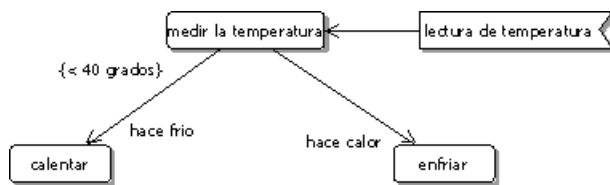
Diagramas de actividad

Pulsamos en “New diagram/activity”

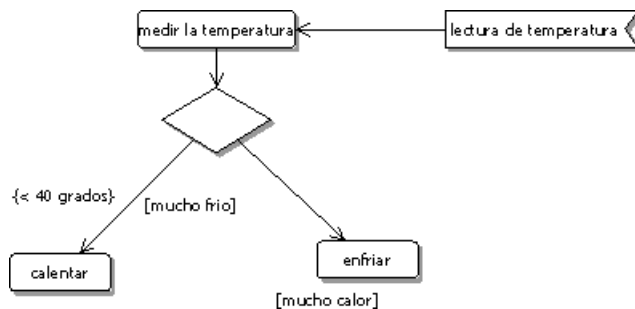


Hemos añadido un ejemplo con un conjunto de actividades, con una línea de sincronización y una actividad final. Como siempre, los elementos del diagrama están en el panel izquierdo.

Otro ejemplo de diagrama de actividad más completo sería:

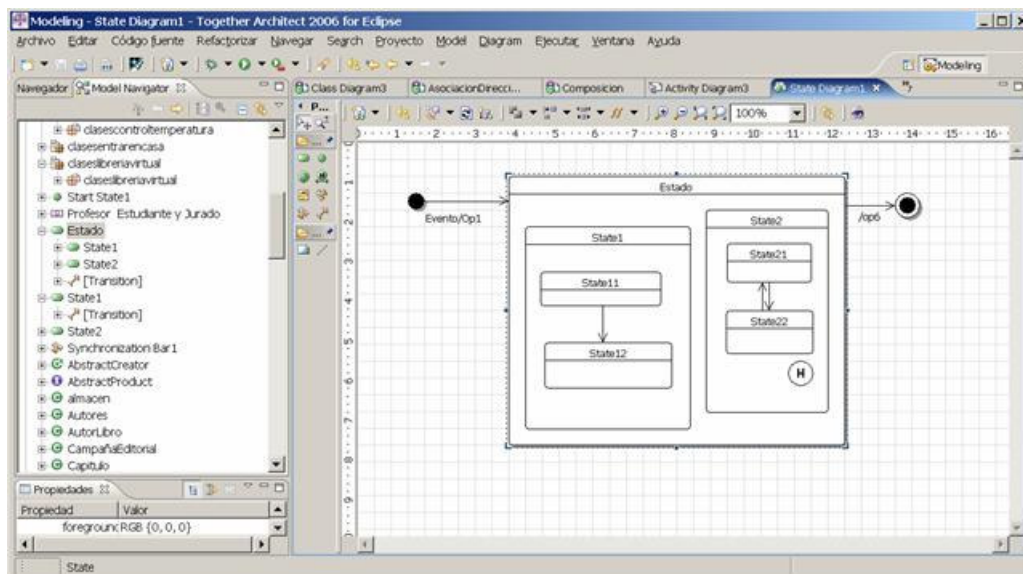


O una representación alternativa



Añadiendo un diagrama de estado

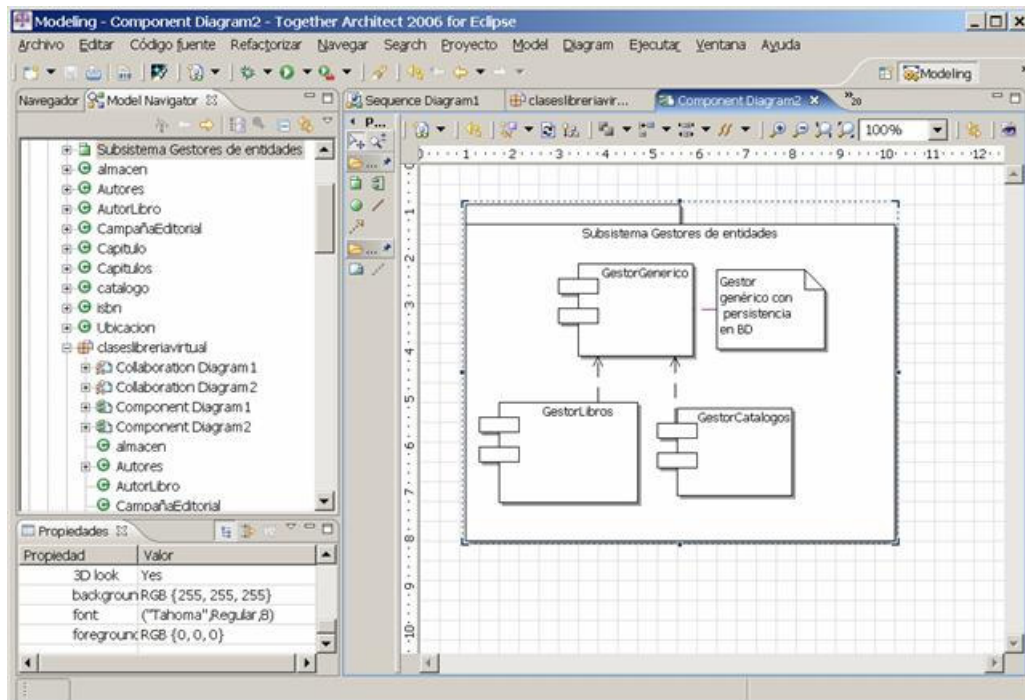
Pulsamos en “new diagram\state” y creamos un diagrama de estados.



Hemos añadido en el ejemplo un estado compuesto de dos estados “ortogonales”, es decir, cuyos estados son independientes entre sí. También hemos indicado que el segundo estado tiene una opción “historia”, que indica que recordará su estado entre instanciaciones diferentes.

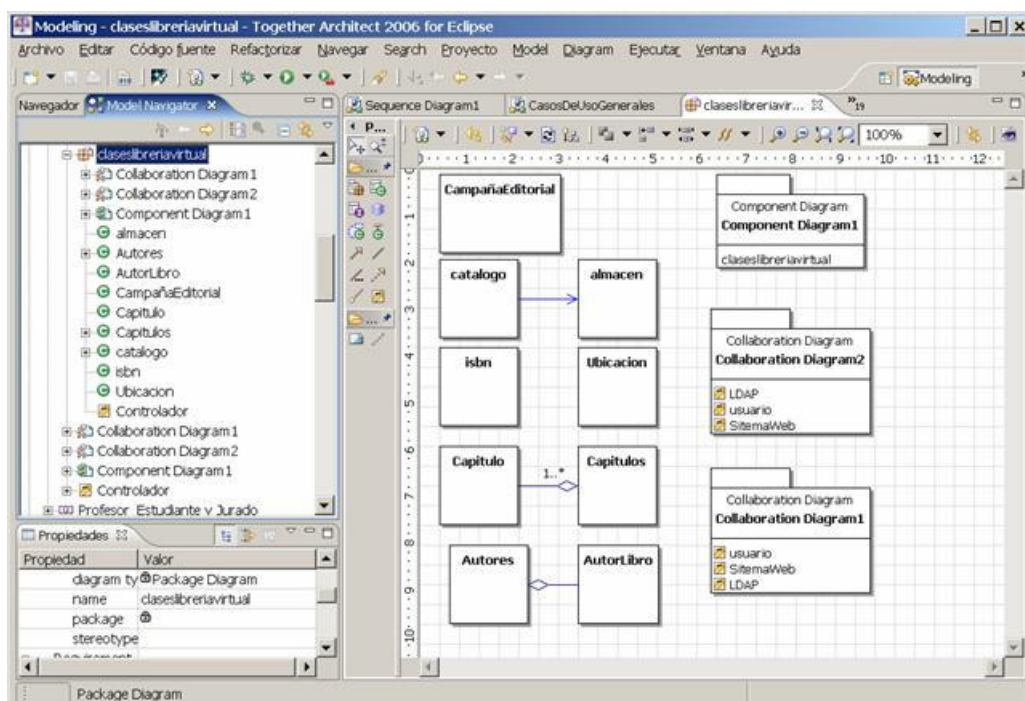
Añadiendo diagramas de componentes

Pulsando en “New diagram\components” podemos crear nuevos diagramas de componentes



Añadiendo diagramas de paquetes

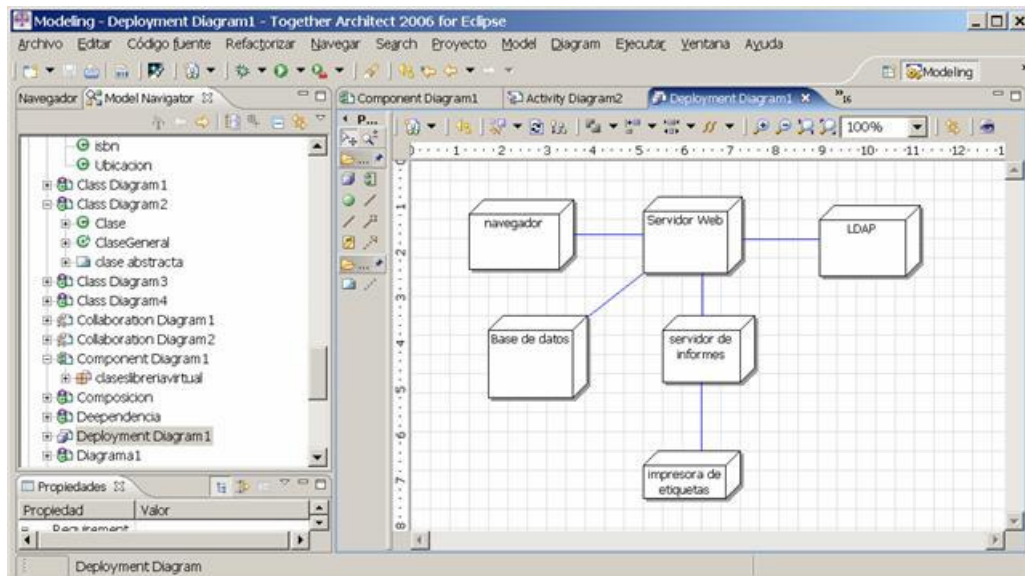
Together genera automáticamente los diagramas para cada paquete. Basta pulsar sobre el paquete para que muestre su diagrama asociado



Al añadir una clase o un diagrama al paquete, Together actualiza el diagrama para reflejar los cambios.

Añadiendo diagramas de despliegue

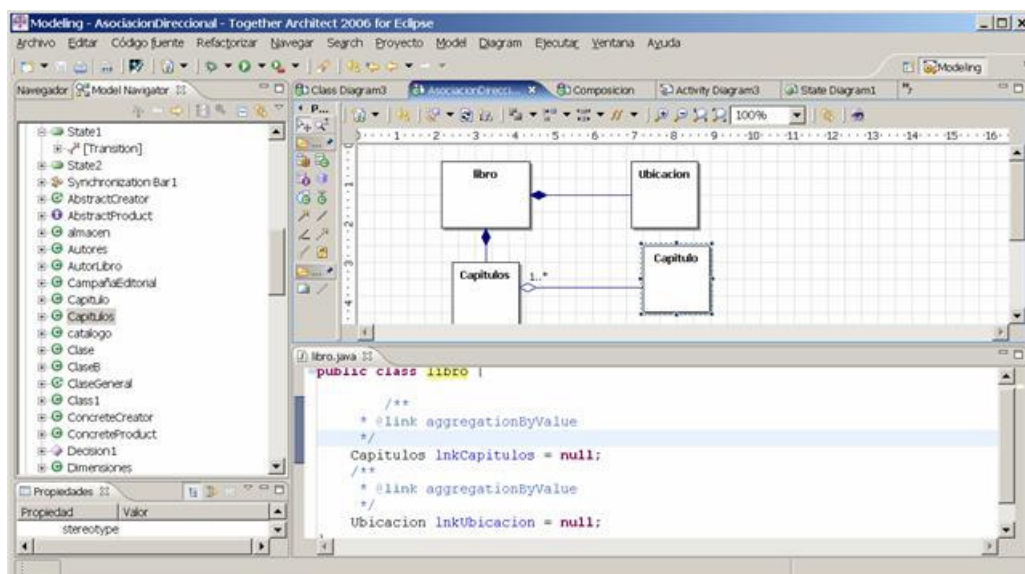
Para añadir un diagrama de despliegue pulsamos sobre "New diagram\deployment". Añadimos al diagrama los componentes que se deben desplegar.



Como siempre, los diferentes elementos que podemos añadir están situados a la izquierda del diagrama.

Generación del código asociado

Together va generando el código a medida que avanzamos en el diagrama. Basta hacer doble clic en uno de los objetos para que en el panel inferior se vea su código fuente asociado.



```
public class libro {

    /**
     * @label usa
     */

    /**isbn Dependency_Link*/

    /**
     * @link aggregationByValue
     */
    Capitulos lnkCapitulos = null;
    /**
     * @link aggregationByValue
     */
    Ubicacion lnkUbicacion = null;
}
```

Como vemos Together utiliza anotaciones en comentarios para añadirle significado a los diferentes tipos de relaciones. Suponemos que en posteriores versiones de Together se modificará esto para hacer uso de las nuevas anotaciones de Java 5.0.

Además si modificamos el código fuente, Together actualiza los diagramas. En efecto, editamos el código fuente directamente y le añadimos un par de atributos.

```
public class libro {

    /**
     * @label usa
     */

    /**#isbn Dependency_Link*/

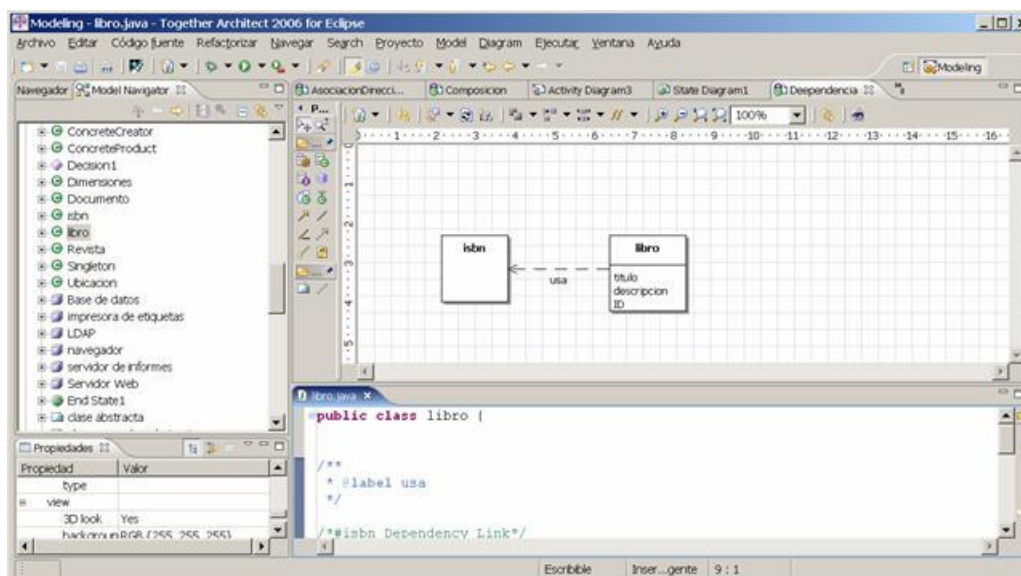
    /**
     * @link aggregationByValue
     */
    Capítulos lnkCapitulos = null;
    /**
     * @link aggregationByValue
     */
    Ubicacion lnkUbicacion = null;

    String titulo;

    String descripcion;

    private String ID;
}
```

Ahora vemos los diagramas para la clase libro:



Aquí vemos que todos los atributos que añadimos a la clase se representarán en el diagrama.

Conclusión

Together es una de las mejores herramientas de modelado UML y de generación de código fuente. Es además bastante intuitivo de manejar, dentro de la complejidad de este tipo de herramientas, y permite generar automáticamente parte del código fuente necesario.

Together nos permite iniciar un proyecto directamente mediante modelado o importar un proyecto java ya existente y generar su modelo automáticamente, lo que nos permite generar rápidamente la documentación UML de nuestro proyecto.

En artículos posteriores profundizaremos sobre algunas características de esta herramienta.

Sobre el autor



Cristóbal González Almirón es consultor de desarrollo de proyectos informáticos. Actualmente desarrolla su labor profesional para la Comunidad de Madrid, a través de su Agencia de Informática y Comunicaciones, en proyectos de implantación de gestión documental.

Su experiencia profesional se ha desarrollado en empresas como Compaq, HP, Mapfre, Endesa, Repsol, Universidad Autónoma de Madrid, en las áreas de Desarrollo de Software (Orientado a Objetos), tecnologías de Internet, Técnica de Sistemas de alta disponibilidad y formación a usuarios.

Si quieres contactar con él, envía un correo a criskerberos-tutoriales@yahoo.com.



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 2.5 License](https://creativecommons.org/licenses/by-nc-nd/2.5/).
[Puedes opinar sobre este tutorial aquí](#)



Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?

¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

info@autentia.com

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos

Autentia = Soporte a Desarrollo & Formación

J2EE, EJBs, Struts...

[Autentia S.L.](#) Somos expertos en:
J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
 y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	<input type="text"/>
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto

Descripción

[UML 2.0 con Borland Together CE](#)

Os mostramos como instalar la versión CE de Borland Together que ya introduce novedades de la especificación UML 2.0. También compartimos pensamientos sobre como afrontar

	correctamente un análisis de un sistema o una reingeniería de su documentación.
Instalación de Together para Eclipse	Os mostramos como instalar la versión de evaluación de Together sobre Eclipse. Estas dos herramientas constituyen una excelente base para el trabajo profesional y serio en entornos Java.
Optimización Java con Eclipse Profiler Plugin	Alejandro Pérez nos enseña como analizar el rendimiento de nuestras aplicaciones con Eclipse Profiler Plugin.
PMD, Eclipse y NetBeans	Tutorial que describe la instalación y uso de PMD en los entornos de desarrollo Eclipse y NetBeans
Modelado UML con Visual Paradigm	Os mostramos como instalar y utilizar la versión gratuita de Visual Paradigm for UML. Cabe destacar que permite extraer elementos de diseño desde textos de análisis.
Herramientas Gratuitas UML	Os mostramos como obtener algunas herramientas gratuitas UML, ArgoUML y Poseidon.
Manejo de Repositorios CVS desde Eclipse	En este tutorial os enseñamos a manejar el repositorio CVS desde la plataforma Eclipse
Soporte XML en Eclipse con X-MEN	Alejandro Perez nos enseña como potenciar el entorno eclipse para facilitarnos el trabajo con ficheros xml, gracias al pluggin X-MEN
Plugin PHPEclipse en Debian	Alejandro Pérez nos muestra como instalar el plugin de Eclipse: PHPEclipse, en GNU / Linux (Debian)
Introducción al UML	Este es el primer artículo sobre el diseño de proyectos orientados a objeto con UML, donde se describe los primeros diagramas a utilizar


Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)

	¿Buscas un hospedaje de calidad por sólo 2€ al mes?
---	--

www.AdictosAlTrabajo.com Optimizado 800X600
