Manual del usuario y referencia

Manual del usuario y referencia de Altova UModel® 2017

Fecha de publicación: 2017

Contenido

1 Altova UModel® 2017 3

2 Introducción a UModel 6

3 Tutorial de UModel 8

3.1 Iniciar UModel ... 10
3.2 Casos de uso ... 14
3.3 Diagramas de clases ... 24
3.3.1 Crear clases derivadas ... 30
3.4 Diagramas de objetos ... 36
3.5 Diagramas de componentes ... 42
3.6 Diagramas de implementación ... 48
3.7 Ingeniería de ida y vuelta (modelo - código - modelo) ... 53
3.8 Ingeniería de ida y vuelta (código - modelo - código) ... 60

4 Interfaz del usuario 68

4.1 Estructura del modelo ... 69
4.2 Árbol de diagramas ... 74
4.3 Favoritos ... 76
4.4 Propiedades ... 78
4.5 Estilos ... 80
4.6 Jerarquía ... 83
4.7 Vista general ... 86
4.8 Documentación ... 87
4.9 Mensajes ... 88
4.10 Panel de diagramas ... 89
4.10.1 Propiedades de los diagramas ... 92
4.10.2 Cortar, copiar y pegar en los diagramas de UModel ... 94
4.11 Agregar/insertar elementos en los modelos ... 98
4.12 Crear hipervínculos entre los elementos de modelado ... 100

4.13 Ejemplos Bank 106

5 Interfaz de la línea de comandos 110

5.1 Archivo: Nuevo / Cargar / Opciones al guardar 114

6 Proyectos e ingeniería de código 118

6.1 Crear un proyecto de UModel desde cero 121
6.2 Importar código fuente a los proyectos 126
6.3 Importar binarios Java, C# y VB 131
6.4 Sincronizar el modelo y el código fuente 135
6.4.1 Consejos prácticos 137
6.4.2 Refactorización de código y sincronización 139
6.5 Requisitos para ingeniería directa 141
6.6 Correspondencia entre código Java y elementos de UModel 143
6.7 Correspondencia entre código C# y elementos de UModel 144
6.8 Correspondencia entre elementos de XML Schema y de UModel 145
6.9 Correspondencia entre código VB.NET y de elementos de UModel 146
6.10 Correspondencia entre elementos de BD y de UModel 147
6.11 Incluir otros proyectos de UModel 148
6.12 Combinar proyectos de UModel 150
6.12.1 Fusión de proyectos a 2 bandas 150
6.12.2 Fusión de proyectos a 3 bandas 151
6.12.3 Ejemplo de fusión manual a 3 bandas 153
6.13 Compartir paquetes y diagramas 156
6.14 Plantillas UML 160
6.14.1 Firmas de plantilla 161
6.14.2 Enlace de plantilla 162
6.14.3 Usar plantillas en operaciones y propiedades 163
6.15 Configuración del proyecto 164
6.16 Mejorar el rendimiento 165

7 Crear relaciones entre los modelos 168

7.1 Ver las relaciones entre los modelos 171
7.2 Asociaciones, realizaciones y dependencias 172

8 Generar documentación UML 176

8.1 Con una hoja de estilos SPS predeterminada ... 183

8.2 Con hojas de estilos predefinidas por el usuario ... 185

9 Diagramas UML 188

9.1 Diagramas de comportamiento ... 189

9.1.1 Diagrama de actividades ... 189

Insertar elementos ... 190

Crear bifurcaciones y convergencias ... 192

Elementos ... 195

9.1.2 Diagrama de máquina de estados ... 205

Insertar elementos ... 206

Crear estados, actividades y transiciones ... 207

Estados compuestos ... 213

Generar código a partir de diagramas de máquina de estados ... 216

Trabajar con código de máquina de estados ... 219

Elementos ... 224

9.1.3 Diagrama de máquina de estados de protocolos ... 227

Insertar elementos ... 228

Elementos ... 229

9.1.4 Diagrama de casos de uso ... 231

9.1.5 Diagrama de comunicación ... 232

Insertar elementos ... 233

9.1.6 Diagrama global de interacción ... 236

Insertar elementos ... 237

9.1.7 Diagrama de secuencia ... 240

Insertar elementos ... 241

Líneas de vida ... 243

Fragmentos combinados ... 244

Usos de interacción ... 248

Puertas ... 248

Invariantes de estado ... 249

Mensajes ... 249

Generar diagramas de secuencia a partir de código fuente ... 254

Generar varios diagramas de secuencia a partir de propiedades ... 257

Generar código a partir de diagramas de secuencia ... 259

Agregar código a diagramas de secuencia ... 263

9.1.8 Diagrama de ciclo de vida ... 266
Insertar elementos ... 267
Línea de vida ... 267
Marca de graduación ... 270
Evento/estímulo ... 271
Restricción de duración ... 271
Restricción de tiempo ... 272
Mensaje ... 273
9.2 Diagramas de estructura ... 275
9.2.1 Diagrama de clases ... 275
Personalizar diagramas de clases ... 275
Invalidar operaciones de clases base e implementar operaciones de interfaz ... 282
Crear métodos getter y setter ... 283
Notaciones de forma esférica (Ball and socket) ... 285
Agregar excepciones emitidas a los métodos de una clase ... 286
Generar diagramas de clases ... 287
9.2.2 Diagrama de estructura de un compuesto ... 290
Insertar elementos ... 290
9.2.3 Diagrama de componentes ... 292
9.2.4 Diagrama de implementación ... 292
9.2.5 Diagrama de objetos ... 293
9.2.6 Diagrama de paquetes ... 293
Insertar elementos ... 295
Generar diagramas de paquetes al importar código o binarios ... 296
9.2.7 Diagrama de perfil y estereotipos ... 298
Agregar estereotipos y definir valores etiquetados ... 300
Estereotipos y enumeraciones ... 304
Estilos de estereotipo definidos por el usuario ... 306
Asignar iconos de estereotipo personalizados ... 307
9.3 Otros diagramas ... 312
9.3.1 Diagramas de esquema XML ... 312
Importar esquemas XML ... 313
Insertar elementos ... 320
Crear y generar un esquema XML ... 323

10 XMI: intercambio de metadatos XML 328

11 Trabajo en equipo con UModel 332

11.1 Crear y editar subproyectos ... 334

12 Control de código fuente 342

12.1 Sistemas de control de código fuente compatibles ... 344

12.2 Comandos de control de código fuente ... 346

12.2.1 Abrir desde el control de código fuente ... 346
12.2.2 Habilitar control de código fuente ... 349
12.2.3 Obtener la versión más reciente ... 349
12.2.4 Obtener ... 350
12.2.5 Obtener carpetas ... 351
12.2.6 Desproteger ... 352
12.2.7 Proteger ... 354
12.2.8 Anular desprotección ... 355
12.2.9 Agregar al control de código fuente ... 356
12.2.10 Quitar del control de código fuente ... 358
12.2.11 Compartir desde el control de código fuente ... 359
12.2.12 Mostrar historial ... 360
12.2.13 Mostrar diferencias ... 362
12.2.14 Mostrar propiedades ... 363
12.2.15 Actualizar estado ... 364
12.2.16 Administrador del control de código fuente ... 364
12.2.17 Cambiar control de código fuente ... 364

12.3 Control de código fuente con Git ... 366

12.3.1 Habilitar Git con el complemento de control de código fuente ... 367
12.3.2 Agregar un proyecto al control de código fuente de Git ... 367
12.3.3 Clonar un proyecto desde el control de código fuente de Git ... 368

13 Iconos en los diagramas de UModel 372

13.1 Diagramas de actividades ... 373

13.2 Diagramas de clases ... 375

13.3 Diagramas de comunicación ... 376

13.4 Diagramas de estructura de un compuesto ... 377

13.5 Diagramas de componentes ... 378

13.6 Diagramas de implementación ... 379

13.7 Diagramas global de interacción ... 380

13.8 Diagramas de objetos ... 381
13.9 Diagramas de paquetes ... 382
13.10 Diagramas de perfil ... 383
13.11 Diagramas de máquina de estados de protocolos ... 384
13.12 Diagramas de secuencia ... 385
13.13 Diagramas de máquina de estados ... 386
13.14 Diagramas de ciclo de vida ... 387
13.15 Diagramas de casos de uso ... 388
13.16 Diagramas de esquema XML ... 389

14 Referencia del usuario 392

14.1 Menú Archivo ... 393
14.2 Menú Edición ... 399
14.3 Menú Proyecto ... 402
14.4 Menú Diseño ... 411
14.5 Menú Vista ... 413
14.6 Menú Herramientas ... 414
14.6.1 Herramientas definidas por el usuario ... 414
14.6.2 Personalizar ... 414
Comandos ... 414
Barras de herramientas ... 416
Herramientas ... 416
Teclado ... 419
Menú ... 420
Opciones ... 421
14.6.3 Restaurar barras de herramientas y ventanas ... 421
14.6.4 Opciones ... 421
14.7 Menú Ventanas ... 428
14.8 Menú Ayuda ... 429

15 Generador de código 436

15.1 Códigos de error ... 437

16 Información sobre licencias 440

16.1 Distribución electrónica de software ... 441
16.2 Activación del software y medición de licencias ... 442

16.3 Derechos de propiedad intelectual 444
16.4 Contrato de licencia para el usuario final 445

Índice

Chapter 1

Altova UModel® 2017

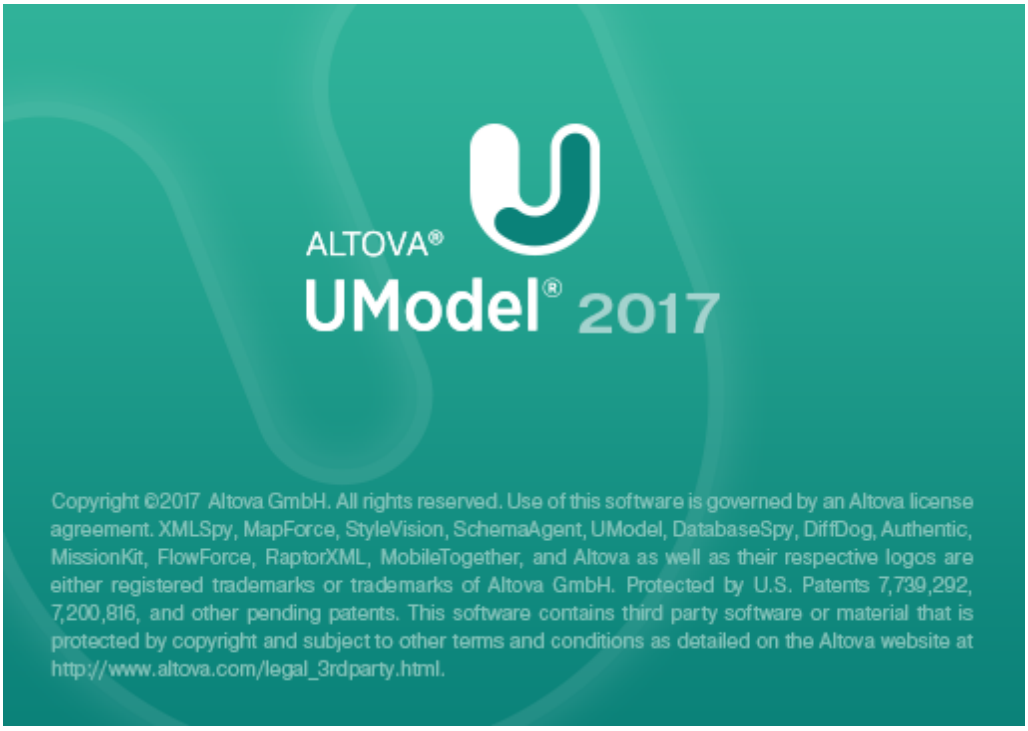
1 Altova UModel® 2017

UModel® 2017 Basic Edition es una asequible herramienta de modelado UML con una potente interfaz gráfica y avanzadas funciones que le facilitarán el trabajo con UML. Además incluye funciones para trabajar con los aspectos más prácticos de la especificación 2.4.
UModel es una aplicación de 32/64 bits para Windows compatible con Windows XP/Vista, Windows 7/8/10 y Windows Server 2003/2008/2012/2016. Las ediciones Enterprise y Professional son compatibles con plataformas de 64 bits.

Principales características de UModel® 2017:

- Compatible con los 14 tipos de diagrama de modelado UML 2.4
- Compatible con procesos de arquitectura dirigida por modelos (MDA), que permite realizar conversiones entre diferentes lenguajes de programación (solo en la edición Enterprise Edition)
- Opción para importar y exportar bases de datos SQL en UModel (solo en las ediciones Enterprise y Professional)
- Funciones para trabajar en equipo y editar proyectos de forma simultánea
- Fusión de proyectos a 3 bandas
- Máquinas de estado de protocolos
- Compatible con diagramas SysML (solo en las ediciones Enterprise y Professional)
- Generación de diagramas de secuencia a partir del código fuente
- Generación de código a partir de diagramas de máquina de estados
- API y complemento de UModel
- Entorno de scripting y editor de formularios integrados (solo en las ediciones Enterprise y Professional)
- Integración en Eclipse
- Compatible con sistemas de control de versiones
- Compatible con diagramas de XML Schema
- Compatible con BPMN (Business Process Modeling Notation) 1.0 y 2.0
- Opción para usar varias capas en el diagrama UML
- Opción para importar binarios Java, C# y Visual Basic
- Opción para crear hipervínculos entre diagramas y elementos de modelado
- Color de sintaxis en los diagramas
- Estilos en cascada
- Número ilimitado de operaciones Deshacer/Rehacer
- Sofisticada generación de código Java, C# y Visual Basic a partir de los modelos
- Ingeniería inversa de código fuente Java, C# y Visual Basic ya existente
- Ingeniería de ida y vuelta para combinar el código con el modelo
- Importación y exportación de modelos con XMI 2.4 para UML 2.0, 2.1, & 2.1.2, 2.2, 2.3, 2.4
- Generación de documentación de proyectos de UModel

Todas estas características le ayudarán a incrementar la productividad y optimizar resultados con UML.



Última actualización: 03/04/2017

Chapter 2

Introducción a UModel

2 Introducción a UModel

Sitio web de Altova: Herramienta de modelado UML Altova UModel

UML es un lenguaje de modelado completo pero no abarca una metodología para procesos de desarrollo, generación de código ni ingeniería de ida y vuelta. UModel está diseñado para ofrecer una total flexibilidad durante el proceso de modelado:

- Los diagramas de UModel se pueden crear en el orden que se quiera y en cualquier momento. No es necesario seguir un orden determinado durante el modelado.
- La actualización/combinación del código o del modelo se puede hacer por proyectos, por paquetes o por clases. Para poder realizar ingeniería de ida y vuelta en UModel no hace falta tener pseudocódigo ni que el código generado incluya comentarios.
- En UModel, la generación de código se realiza sobre plantillas SPL y, por tanto, se puede personalizar. Las personalizaciones son detectadas automáticamente durante la generación de código.
- La función de generación de código y de ingeniería inversa es compatible con estos lenguajes

Lenguaje	Versión
C#	1.2, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0
Java	1.4, 5.0 (1.5), 6, 7, 8
Visual Basic .NET	7.1, 8.0, 9.0

- Cada proyecto puede ser compatible con Java, C# y VB simultáneamente.
- UModel es compatible con plantillas y genéricos UML.
- Intercambio de metadatos XML (XMI 2.1 para UML 2.0, 2.1.2, 2.2, 2.3 y XMI 2.4 para UML 2.4.1)
- Cuando se añaden propiedades u operaciones, UModel ofrece ayudantes de entrada contextuales donde el usuario puede elegir tipos, nivel de protección y otras propiedades normalmente disponibles en entornos IDE como XMLSpy, Visual Studio o Eclipse.
- Color de sintaxis en diagramas para poder trabajar de forma más intuitiva.
- Los elementos de modelado y sus propiedades (fuente, color, borde, etc.) se pueden personalizar de forma jerárquica por proyectos, por nodos/líneas, por familias de elementos o por elementos.
- Puede definir actores personalizables en diagramas de casos de uso para ilustrar terminales y otros símbolos.
- Puede buscar elementos de modelado por su nombre en la pestaña Diagramas, en el panel Estructura del modelo y en las ventanas Mensajes y Documentación.
- Puede buscar/resaltar clases, asociaciones de objetos, dependencias, generalizaciones, etc. con ayuda del menú contextual de los diagramas.
- Las operaciones de Deshacer/Rehacer no solo abarcan cambios en el contenido sino también cambios de estilo realizados en los elementos del modelo.

Nota: el objetivo de la presente documentación no es describir ni explicar el lenguaje UML. Su objetivo es explicar al usuario cómo utilizar la herramienta de modelado Altova UModel para modelar código y completar procesos de ingeniería de ida y vuelta correctamente.

Chapter 3

Tutorial de UModel

3 Tutorial de UModel

Este tutorial describe paso a paso cómo crear un proyecto de modelado en UModel.

El tutorial se ocupa principalmente del proceso de ingeniería directa, es decir, de cómo usar UModel para crear diagramas UML y generar código. Después describe el proceso de ingeniería inversa, tanto desde el punto de vista del código como del modelo.

Estos son los diagramas UML que abarca el tutorial, que además explica cómo manipular sus elementos de modelado:

- Casos de uso
- Diagramas de clase
- Diagramas de objetos
- Diagramas de componentes
- Diagramas de implementación

El apartado sobre ingeniería de ida y vuelta (modelo - código - modelo) explica:

- Cómo generar código desde UModel
- Cómo agregar una operación nueva al código externo
- Cómo combinar el código externo con el modelo de UModel otra vez

El apartado sobre ingeniería de ida y vuelta (código - modelo - código) explica:

- Cómo importar código generado con XMLSpy desde un directorio o archivo de proyecto
- Cómo agregar una clase nueva al modelo generado en UModel
- Cómo combinar el proyecto actualizado con el código externo

Todos los archivos utilizados en este tutorial están disponibles en la carpeta `C:\Documents and Settings\All Users\Application Data\Altova`. Cuando un usuario inicia la aplicación por primera vez, el archivo `Examples` del usuario se copia en la carpeta `C:\Documents and Settings\<usuario>\Mis Documentos\Altova\UModel2017\UModelExamples\`. Por tanto, es importante que no mueva, edite ni elimine los archivos de ejemplo del directorio `...\All Users\...` inicial.

BankView-start.ump
Este es el archivo de proyecto de UModel que incluye el estado inicial del ejemplo que se utiliza en el tutorial. Este archivo contiene varios diagramas y varias clases, objetos y otros elementos del modelo. Durante el tutorial se añadirán nuevos, paquetes, diagramas y otros elementos que le ayudarán a entender lo fácil que es modelar aplicaciones en UModel. Recuerde que la función de revisión de sintaxis emite un informe con errores y advertencias sobre el archivo y el tutorial explica cómo solucionar estos problemas.

BankView-finish.ump
Este es el archivo de proyecto de UModel que incluye el estado final del ejemplo que utiliza el tutorial, tras seguir paso a paso todas las instrucciones. Este archivo de proyecto es el que se utiliza para generar código y sincronizarlo con UModel.

OrgChart.zip
Este archivo está en la carpeta `C:\Documents and Settings\All Users\Application Data\Altova` y se usa para el proceso de ingeniería de ida y vuelta. Desempaquételo en la carpeta `...\UModelExamples` antes de empezar.

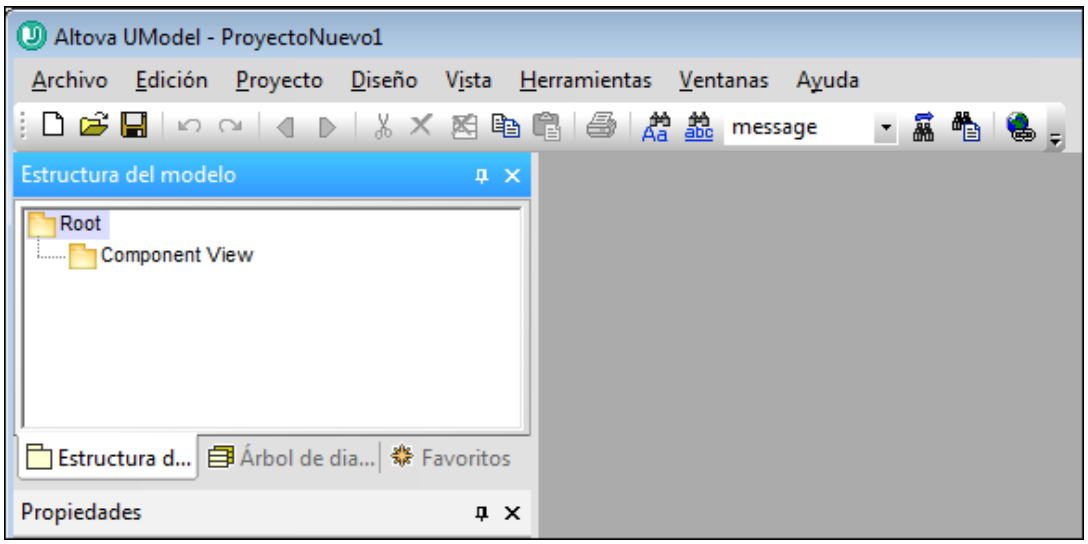
En el mismo directorio hay otros archivos de ejemplo para los lenguajes de programación **Java** y **C#** (es decir, **Bank_Java.ump**, **Bank_CSharp.ump** y **Bank_MultiLanguage.ump**). Estos archivos de proyecto también incluyen diagramas de secuencia que se describen en el resto de la documentación.

En la sección sobre proyectos y generación de código del tutorial encontrará información sobre cómo crear un proyecto desde cero y generar código.

3.1 Iniciar UModel

Tras instalar UModel en el equipo:

1. Inicie UModel haciendo doble clic en el icono de UModel del Escritorio o desde el menú **Inicio | Todos los programas**.
   UModel se inicia con el proyecto predeterminado **`ProyectoNuevo1`**, que aparece en la interfaz gráfica.

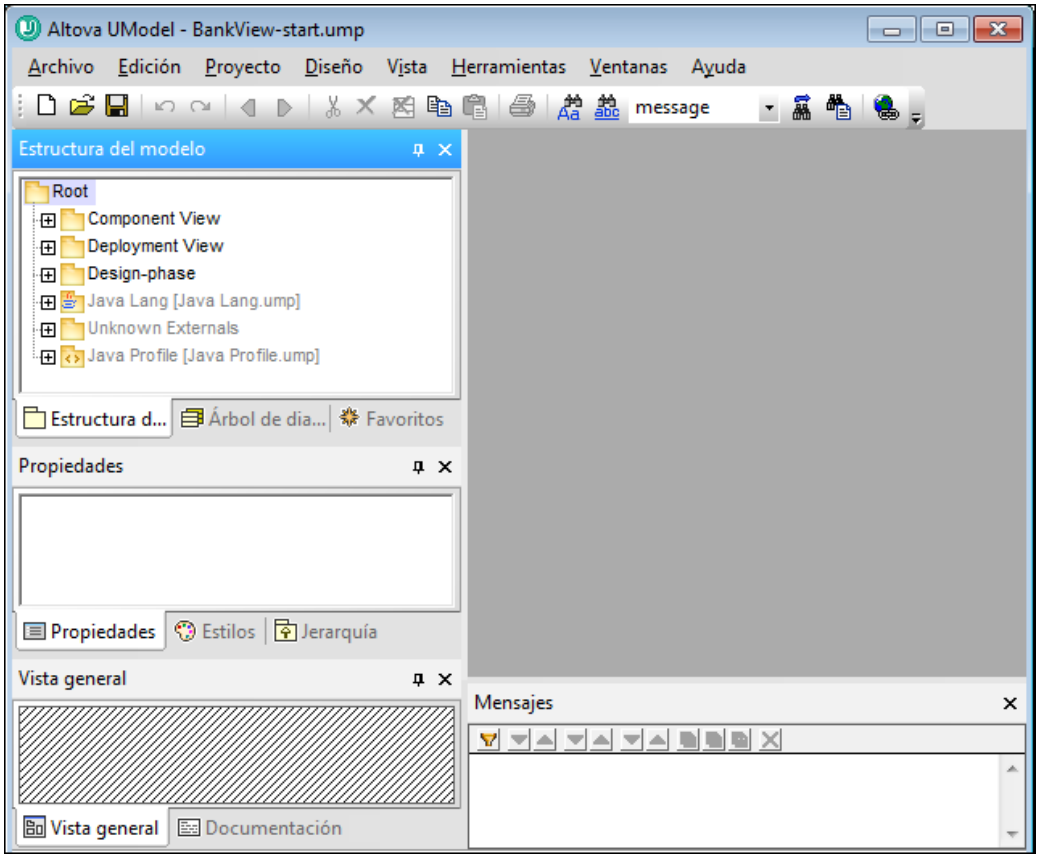


   Observe que la interfaz gráfica se divide en dos grandes secciones: el panel izquierdo formado por tres pestañas y el panel de diagramas de la derecha (que por ahora está vacío).

   En la pestaña Estructura del modelo puede ver dos paquetes predeterminados: `Root` y `Component View`. Estos dos paquetes no se pueden eliminar ni renombrar.

Para abrir el proyecto BankView-start:

1. Seleccione la opción de menú **Archivo | Abrir** y navegue hasta la carpeta `...\UModelExamples\Tutorial` de UModel. Recuerde que también puede abrir archivos `*.ump` desde una URL (haciendo clic en el botón **Cambiar a URL**).
2. Abra el archivo de proyecto **BankView-start.ump**.
   El archivo de proyecto se carga en UModel. Bajo el paquete `Root` aparecen ahora varios paquetes predefinidos. Observe que por ahora la ventana principal sigue vacía.



El panel superior izquierdo ofrece varias vistas del proyecto de modelado:

- La pestaña *Estructura del modelo* muestra todos los elementos de modelado del proyecto de UModel. Los elementos se pueden manipular en esta pestaña directamente, usando las teclas de edición estándar y operaciones de arrastrar y colocar.
- La pestaña *Árbol de diagramas* ofrece acceso rápido a los diagramas que componen el proyecto, independientemente de su posición en la estructura del proyecto. Los diagramas aparecen agrupados por tipo.
- La pestaña *Favoritos* es un repositorio de elementos de modelados que el usuario puede personalizar. En esta pestaña puede colocar todo tipo de elementos de modelado, haciendo clic en el comando **Agregar a favoritos** del menú contextual.

El panel central izquierdo ofrece varias vistas de determinadas propiedades del modelo:

- La pestaña *Propiedades* muestra las propiedades del elemento que está seleccionado en la *Estructura del modelo* o en el panel de diagramas. Las propiedades de los elementos se pueden definir y actualizar en esta pestaña.
- La pestaña *Estilos* muestra los atributos de los diagramas o de los elementos que están visibles en el panel de diagramas. Estos atributos de estilo se dividen en dos categorías: formato y presentación.
- La pestaña *Jerarquía* muestra todas las relaciones que tiene el elemento de modelado seleccionado. El elemento de modelado puede estar seleccionado en el diagrama, en la *Estructura del modelo* o en *Favoritos*.

El panel inferior izquierdo está compuesto por:

- La pestaña *Vista general*, que muestra la estructura general del diagrama activo.
- La pestaña *Documentación*, que sirve para documentar las clases una a una.

Iconos de los elementos de modelado en la *Estructura del modelo*

Tipos de paquetes:

Paquete UML
Paquete raíz del espacio de nombres Java
Paquete raíz del espacio de nombres C#
Paquete raíz del espacio de nombres Visual Basic
Paquete raíz de XML Schema
Paquete de código Java, C# y VB (las declaraciones de paquete se crean cuando se genera el código)

Tipos de diagramas:

Diagrama de actividades
Diagrama de clases
Diagrama de comunicación
Diagrama de componentes
Diagrama de estructura de un compuesto
Diagrama de implementación
Diagrama global de interacción
Diagrama de objetos
Diagrama de paquetes
Diagrama de perfil
Diagrama de secuencia
Diagrama de máquina de estados
Diagrama de ciclo de vida
Diagrama de caso de uso
Diagrama de esquema XML

Tipos de elementos:

Los elementos que están visibles en el diagrama activo aparecen con un punto azul en la parte inferior. En este caso se trata de un elemento de tipo clase.

Instancia de clase / objeto
Slot de instancia

Clase
Propiedad
Operación
Parámetro

Actor (visible en el diagrama de casos de uso activo)
Caso de uso
Componente
Nodo
Artefacto
Interfaz

Relaciones (o paquete)
Restricciones

3.2 Casos de uso

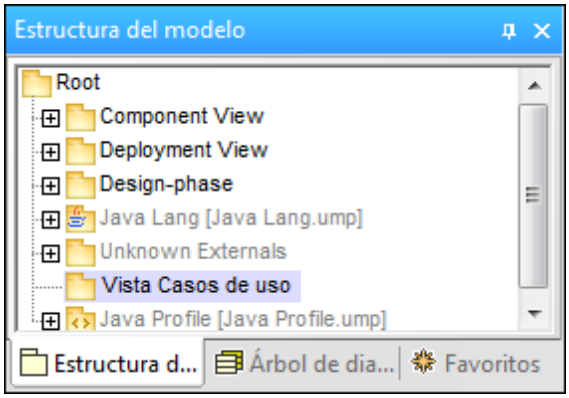
Sitio web de Altova: diagramas de casos de uso UML

En este apartado del tutorial aprenderá a:

- Agregar un **paquete** nuevo al proyecto.
- Agregar un **diagrama** de casos de uso nuevo al proyecto.
- Agregar **elementos** al diagrama y definir dependencias entre ellos.
- Alinear los elementos y ajustar su tamaño.

Para agregar un paquete nuevo al proyecto:

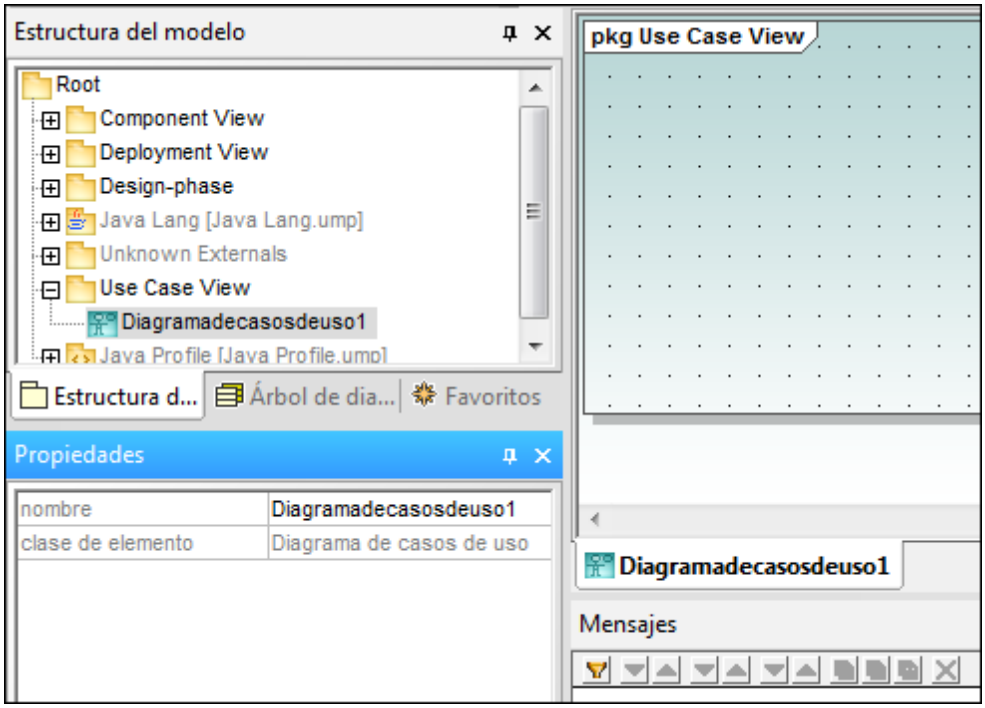
1. En la pestaña *Estructura del modelo* haga clic con el botón derecho en el paquete `Root` y seleccione **Elemento nuevo | Paquete**.
2. Escriba el nombre del nuevo paquete (p. ej. `Vista Casos de uso`) y pulse **Entrar**.



Para más información sobre los paquetes y sus propiedades consulte el apartado Paquetes de la documentación.

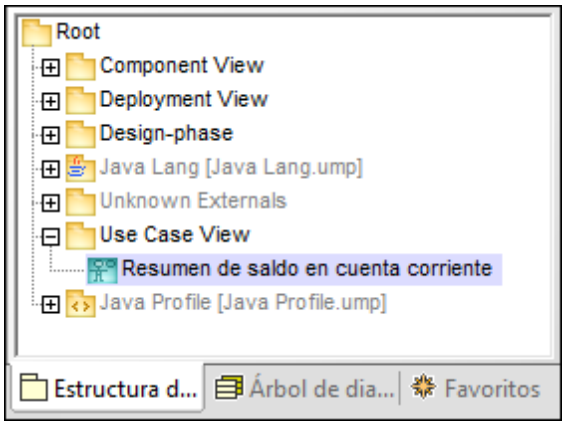
Agregar un diagrama a un paquete:

1. Haga clic con el botón derecho en el paquete `Vista Casos de uso` que acaba de crear.
2. Seleccione el comando **Diagrama nuevo | Diagrama de casos de uso**.



Al paquete se añade un diagrama de casos de uso (en la *Estructura del modelo*) y en el panel de diagramas aparece la pestaña *Diagramacasosdeuso1*, que lleva el nombre predeterminado creado automáticamente para el diagrama.

3. En la *Estructura del modelo* haga doble clic en el nombre del diagrama, escriba `Resumen de saldo en cuenta corriente` y pulse **Entrar** para confirmar.

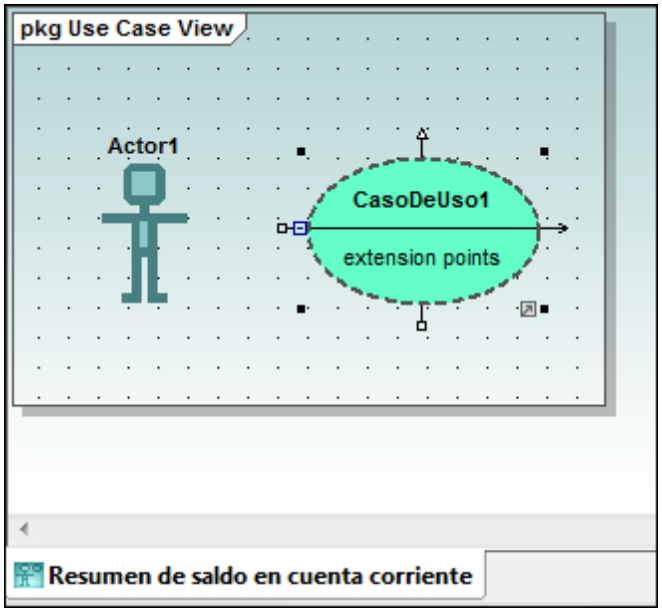


Para más información sobre los diagramas y sus propiedades consulte la sección Diagramas de la documentación.

Agregar elementos al diagrama de casos de uso:

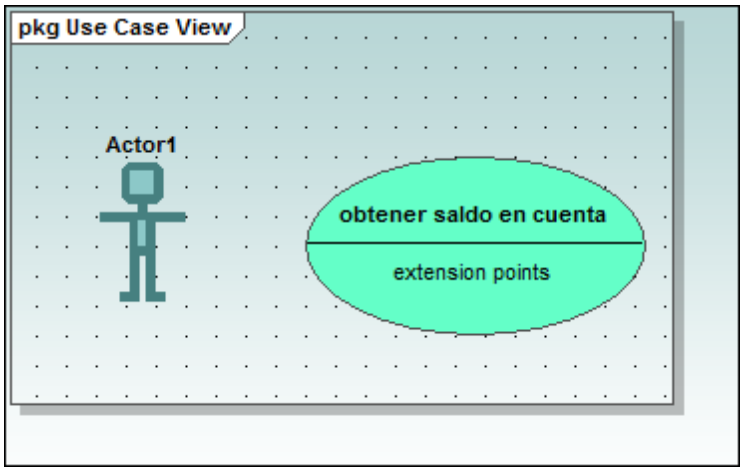
1. Haga clic con el botón derecho dentro del panel del diagrama recién creado y seleccione **Nuevo/a | Actor**.
   El elemento `actor` se inserta en la posición donde se hizo clic.
2. Haga clic en el icono **Caso de uso**  de la barra de herramientas y después en el panel del diagrama para insertar el elemento.
   El elemento `CasoDeUso1` se inserta en el diagrama. Observe que el elemento y su

nombre están seleccionados y, por tanto, sus propiedades se pueden ver en la pestaña *Propiedades*.



3. Cambie el nombre de este elemento a **`obtener saldo en cuenta`** y pulse **Entrar** para confirmar. Si el nombre del elemento no está seleccionado, haga doble clic para seleccionarlo.

   Observe que el tamaño del caso de uso se ajusta automáticamente a la longitud del texto.

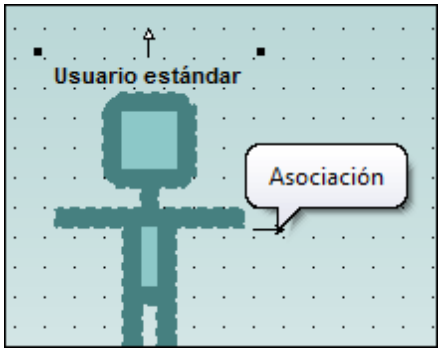


Los elementos del modelo tiene varios controladores de conexión y otros componentes que sirven para manipularlos.

Nota: use **Ctrl+Entrar** para añadir un salto de línea en el nombre del caso de uso.

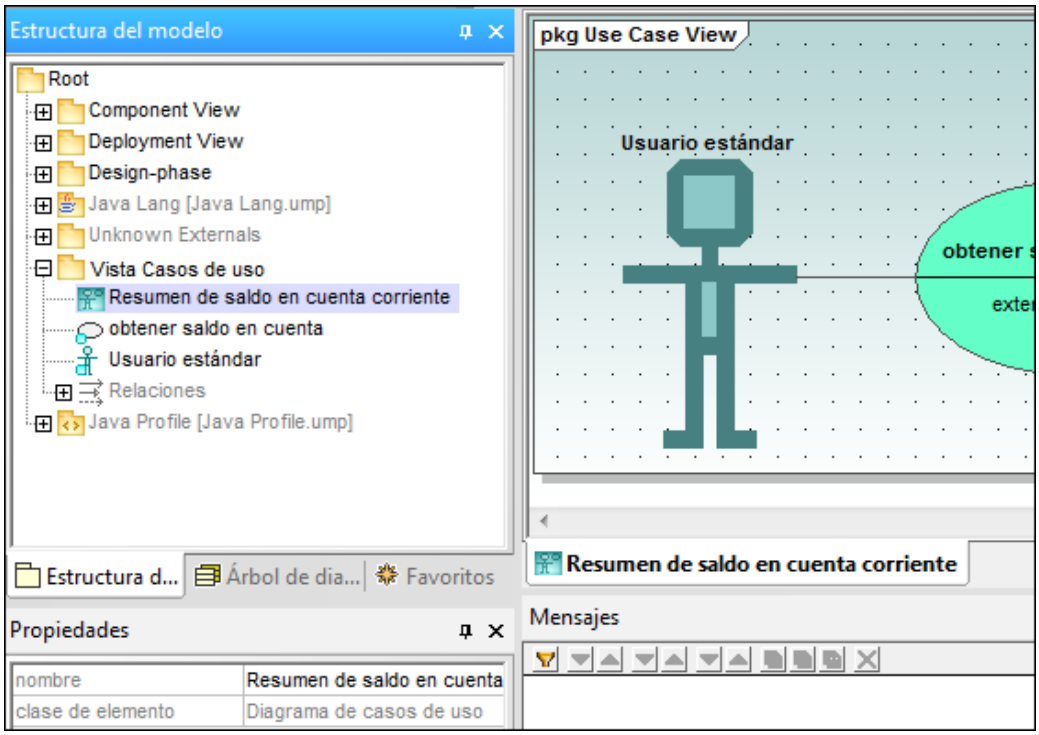
Manipular elementos de UModel: controladores y compartimentos

1. Haga doble clic en el texto **`Actor1`** del elemento `actor`, escriba el nuevo nombre **`Usuario estándar`** y pulse **Entrar** para confirmar.
2. Pase el cursor por encima del controlador derecho del actor.
   Aparece una nota de información rápida que dice `Asociación`.

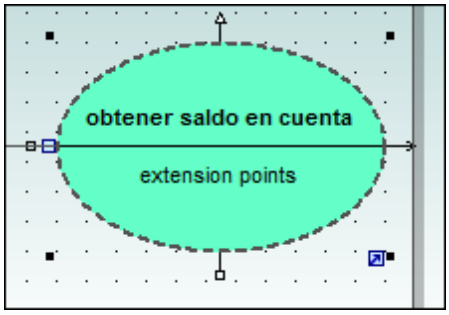


3. Haga clic en el controlador, arrastre la línea de asociación hacia la derecha y suéltela en el caso de uso **`obtener saldo en cuenta`**.

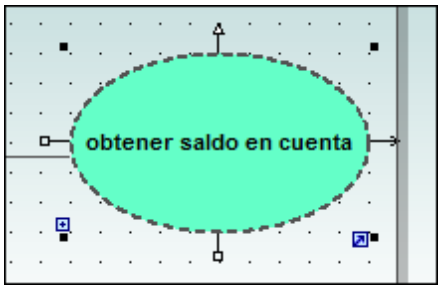
   Ahora se crea una asociación entre el actor y el caso de uso. Las propiedades de la asociación se pueden ver en la pestaña *Propiedades*. La nueva asociación también se añade a la *Estructura del modelo*, bajo el elemento `Relaciones` del paquete `Vista Casos de uso`.

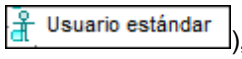


4. Haga clic en el caso de uso y arrástrelo hacia la derecha.
   Las propiedades de la asociación están visibles en el objeto asociación.
5. Haga clic en el caso de uso para seleccionarlo y después haga clic en el icono contraer situado en el borde izquierdo del caso de uso.



Esto oculta el compartimento `puntos de extensión`.



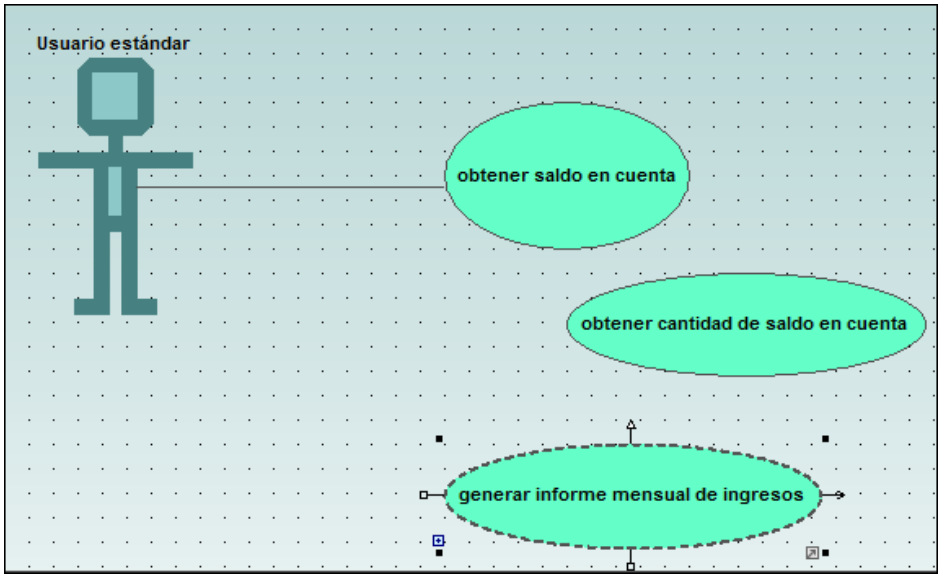
Nota: si en la *Estructura del modelo* el icono del elemento incluye un punto azul (p. ej. 

), esto significa que el elemento está visible en el panel de diagramas actual. Al ajustar el tamaño del actor también se ajusta el campo de texto, que puede ser multilínea. Puede insertar un salto de línea haciendo clic en **Ctrl+Entrar**.

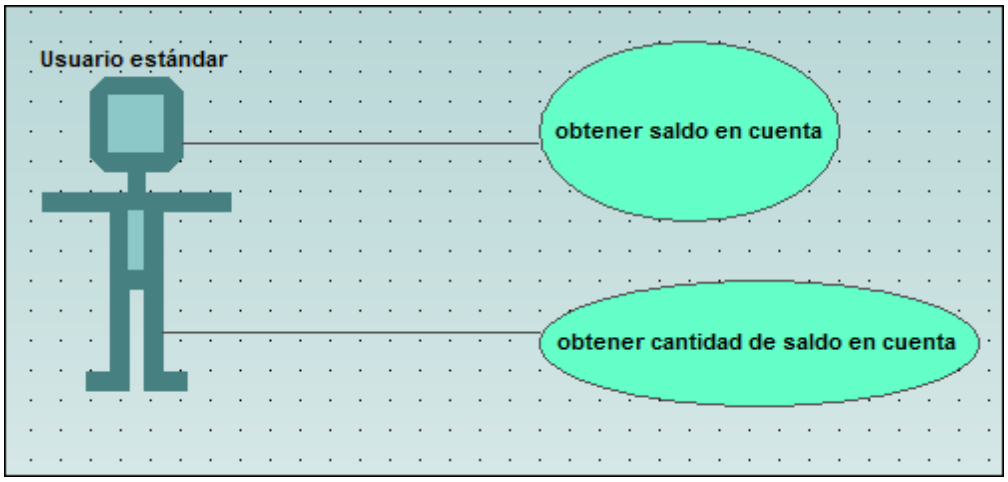
Terminar el diagrama de casos de uso:

Ahora, usando los métodos aprendidos hasta ahora:

1. Haga clic en el icono **Caso de uso** de la barra de herramientas mientras pulsa la tecla **Ctrl**.
2. Haga clic en dos posiciones verticales distintas del panel del diagrama para añadir dos casos de uso más. Cuando termine puede soltar la tecla **Ctrl**.
3. Al primer caso de uso lo llamamos **`obtener cantidad de saldo en cuenta`** y al segundo **`generar informe mensual de ingresos`**.
4. Haga clic en el icono contraer de ambos casos de uso para ocultar el compartimento de los puntos de extensión.

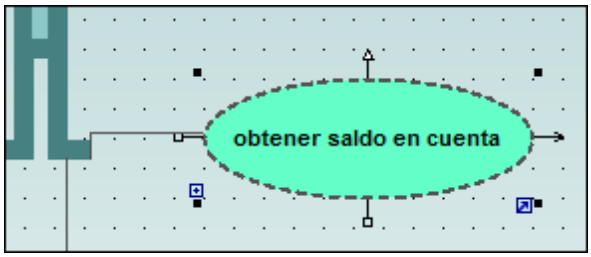


5. Haga clic en el actor y cree una asociación entre **`Usuario estándar`** y **`obtener saldo en cuenta`**.

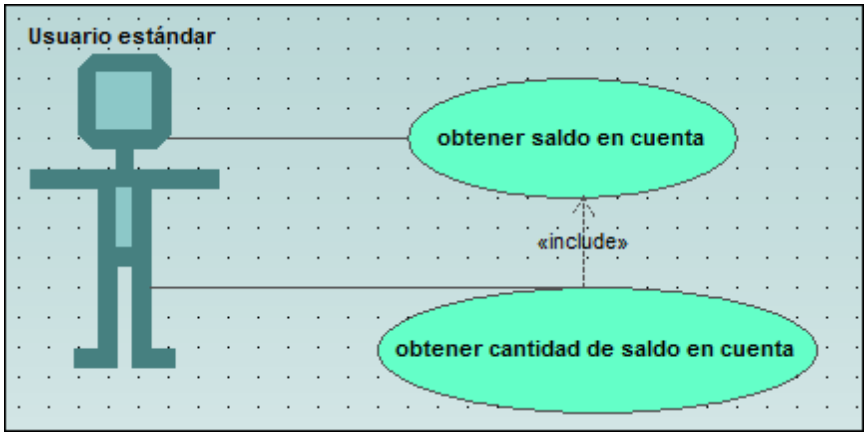


Para crear una dependencia de inclusión entre los casos de uso (creando un caso de uso subordinado):

1. Haga clic en el controlador *Inclusión* del caso de uso **`obtener cantidad saldo en cuenta`**, situado en el borde inferior, y arrástrelo hasta el caso de uso **`obtener saldo en cuenta`**.



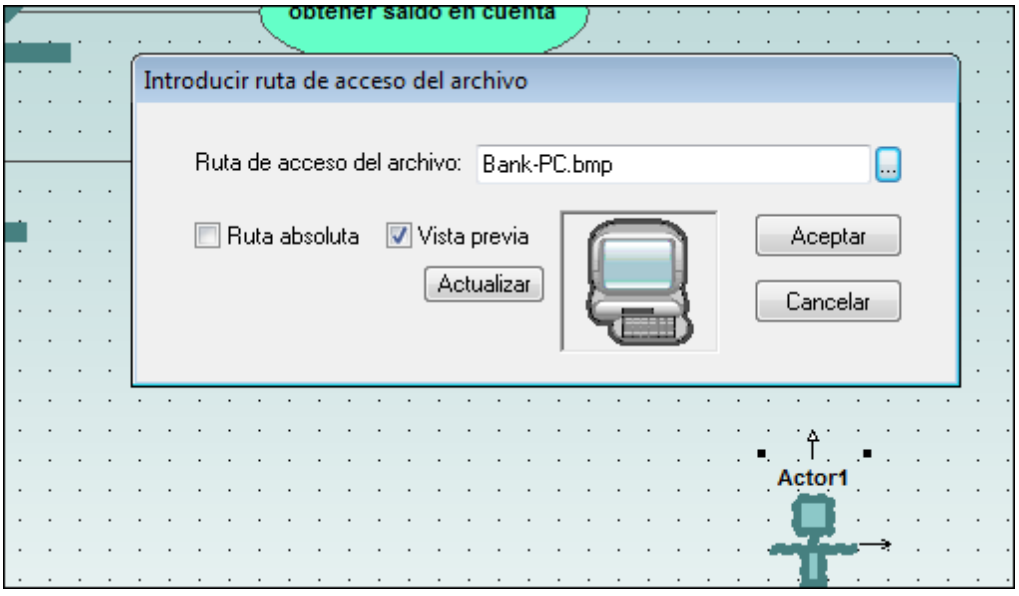
Se crea la dependencia de inclusión y el estereotipo `include` aparece en la flecha de línea discontinua.



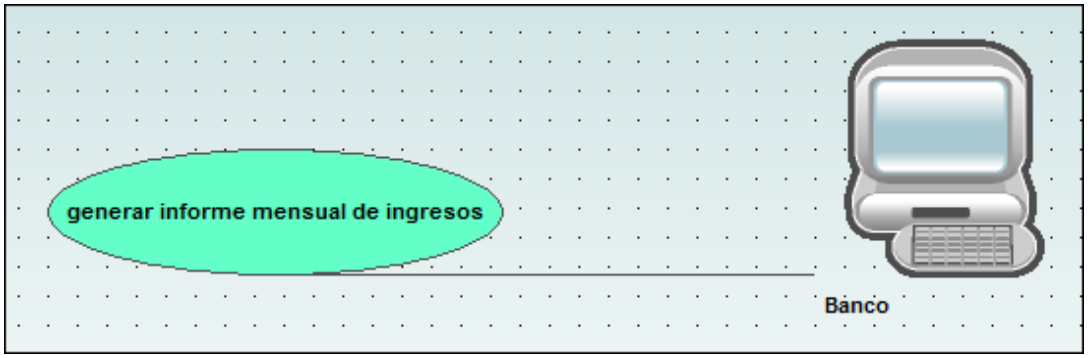
Insertar actores definidos por el usuario:

El actor del caso de uso **`generar informe mensual de ingresos`** no es una persona, sino un trabajo automatizado de procesamiento por lotes que ejecuta una computadora del banco.

1. Inserte un actor en el diagrama haciendo clic en el icono **Actor** de la barra de herramientas.
2. Cambie el nombre del actor por **`Banco`**.
3. En la pestaña *Propiedades*, junto a la entrada `nombre de archivo del icono`, haga clic en el icono  .
4. Busque el mapa de bits que desea usar (en este caso `Bank-PC.bmp`).
5. Desactive la casilla *Ruta absoluta* para que la ruta de acceso sea relativa. Aparece una vista previa del archivo seleccionado.



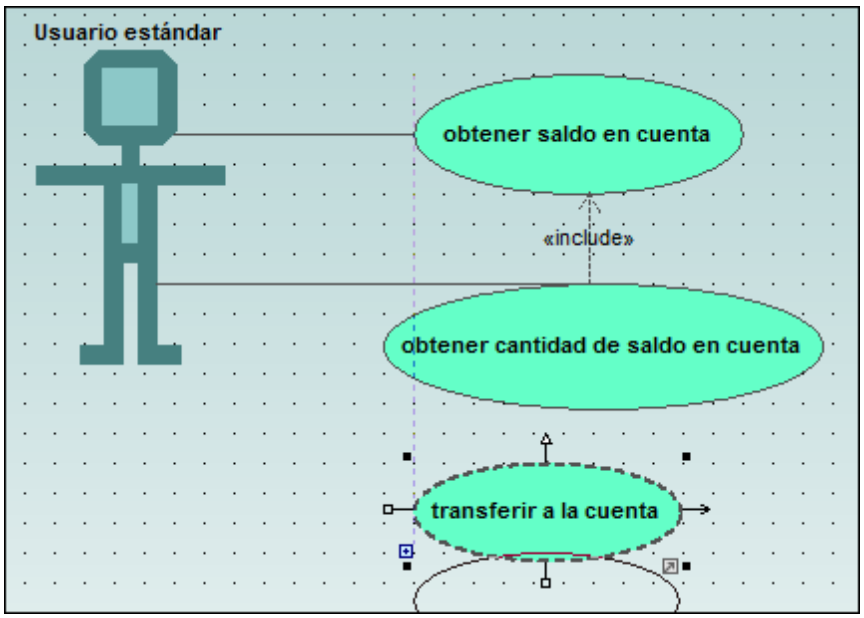
6. Haga clic en **Aceptar** para confirmar e insertar el nuevo actor.
7. Mueva el actor **`Banco`** a la derecha del caso de uso que está más abajo.
8. Haga clic en el icono **Asociación**  de la barra de herramientas. Seleccione el actor **`Banco`** y arrastre el puntero del ratón hasta el caso de uso **`generar informe mensual de ingresos`**. Esta es otra manera de crear asociaciones.



Nota: utilice los valores RGB 82.82.82 para el color de fondo si quiere que el mapa de bits sea transparente.

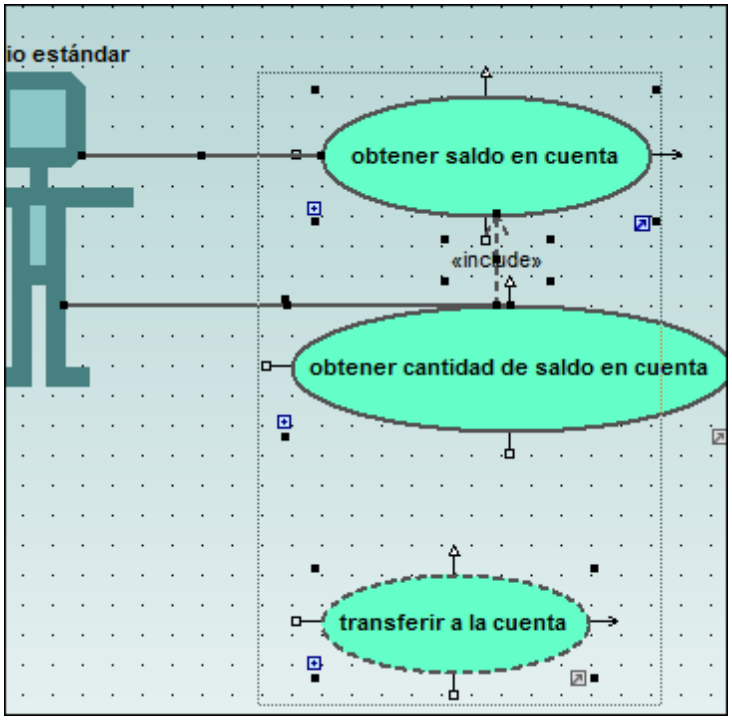
Líneas de ajuste mientras arrastramos elementos

Cuando se arrastran componentes del diagrama, aparecen unas líneas de cuadrícula que ayudan a alinear unos elementos con otros en el diagrama. Estas líneas de guía se pueden habilitar/deshabilitar con la opción de menú **Herramientas | Opciones | Vista** > *Alineación* > *Habilitar líneas de ajuste*).



Alinear los elementos y ajustar su tamaño:

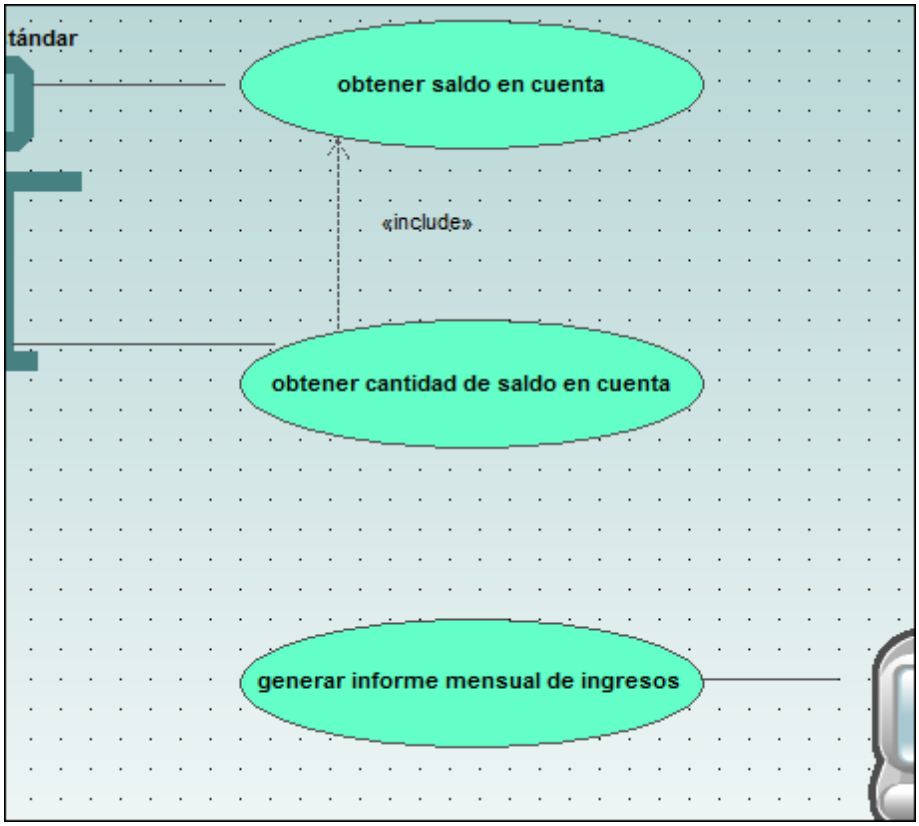
1. Haga clic en el fondo del diagrama y arrastre el puntero para seleccionar los tres casos de uso.
   Observe que el último de los casos de uso seleccionados aparece con un contorno discontinuo, no solo en el panel del diagrama sino también en la *Vista general*.



Ahora están seleccionados todos los casos de uso y el último se utiliza como referencia para los ajustes que vamos a hacer ahora.

2. Haga clic en el icono **Igualar tamaño** de la barra de herramientas.
3. Haga clic en el icono **Centrar horizontalmente** para alinear todos los casos de uso.
   Ahora los elementos caso de uso se centran en el panel y todos tienen el mismo tamaño.

Nota: también puede seleccionar varios elementos a la vez si pulsa la tecla **Ctrl** mientras hace clic en los elementos.


tándar
obtener saldo en cuenta
«include»
obtener cantidad de saldo en cuenta
generar informe mensual de ingresos

3.3 Diagramas de clases

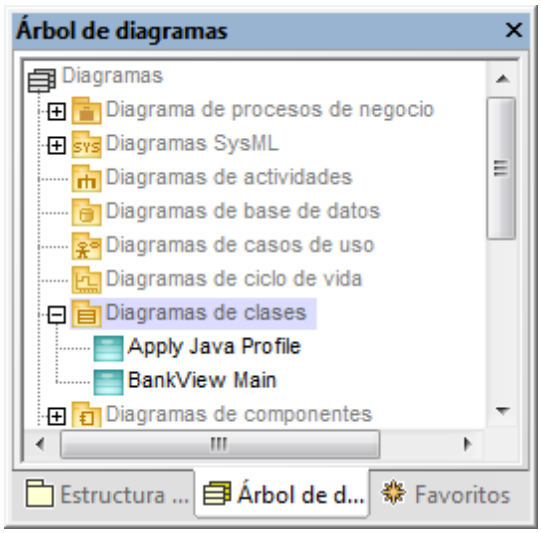
Sitio web de Altova: diagramas de clases UML

En este apartado del tutorial aprenderá a:

- Agregar una clase abstracta nueva (la clase **`Account`**), atributos y operaciones.
- Crear una asociación compuesta entre **`Bank`** y **`Account`**.

Para abrir un diagrama distinto en UModel:

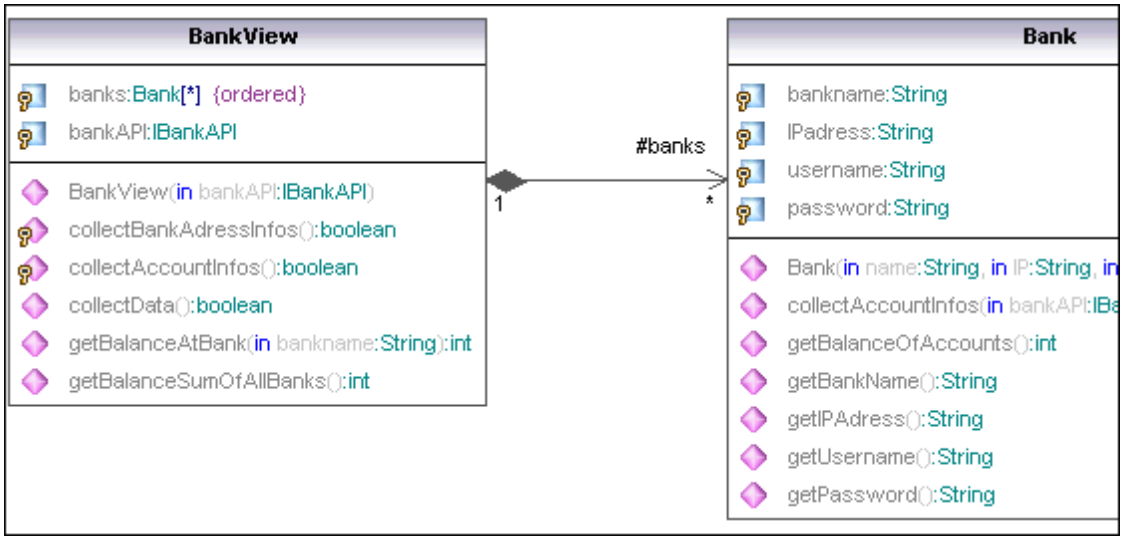
1. Haga clic en la pestaña *Árbol de diagramas*.
2. Expanda el paquete *Diagramas de clases* para ver su contenido.



   En la lista aparecen todos los diagramas de clases del proyecto.
3. Haga doble clic en el icono del diagrama **`BankView Main`** (en el icono, no en el nombre).
   El diagrama de clases se abre en el panel de diagramas.

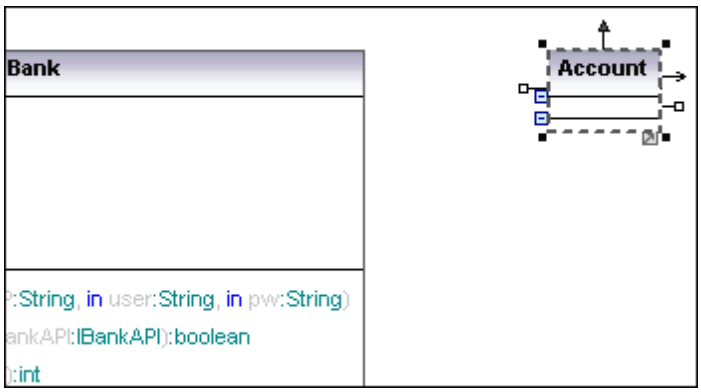
Nota: también puede hacer doble clic en el icono de diagrama de clases de la *Estructura de modelos*, debajo del paquete **`BankView`**.

En el diagrama de clases puede ver dos clases concretas que tienen una asociación compuesta.

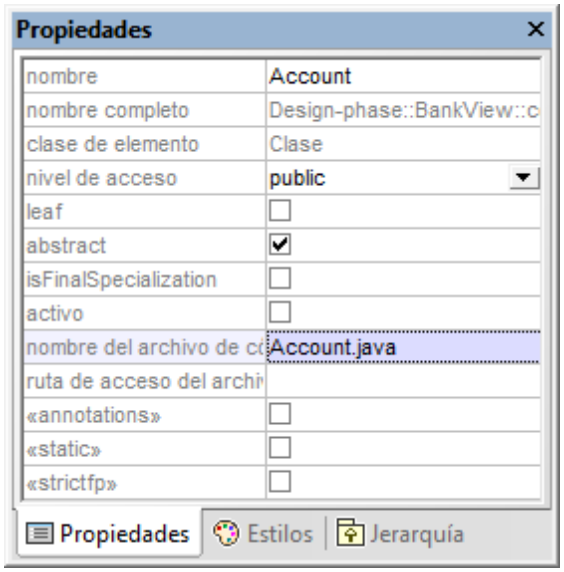


Para agregar una clase nueva y definirla como clase abstracta:

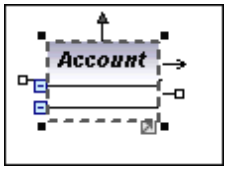
1. Haga clic en el icono **clase**  de la barra de herramientas y después haga clic a la derecha de la clase **`Bank`**. Esto inserta una clase nueva llamada **`Clase1`**.
2. Cambie el nombre de la clase por **`Account`** y pulse **Entrar** para confirmar.



   Observe que la pestaña *Propiedades* muestra las propiedades de la clase activa.
3. Marque la casilla `abstract` del panel *Propiedades* para que la clase sea abstracta.
4. Haga clic en el cuadro de texto `nombre del archivo de código` y escriba *Account.java* para definir la clase Java.

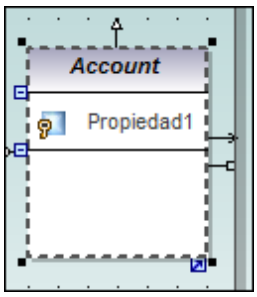


Ahora el nombre de la clase aparece en cursiva, lo cual nos sirve para identificarla como clase abstracta.

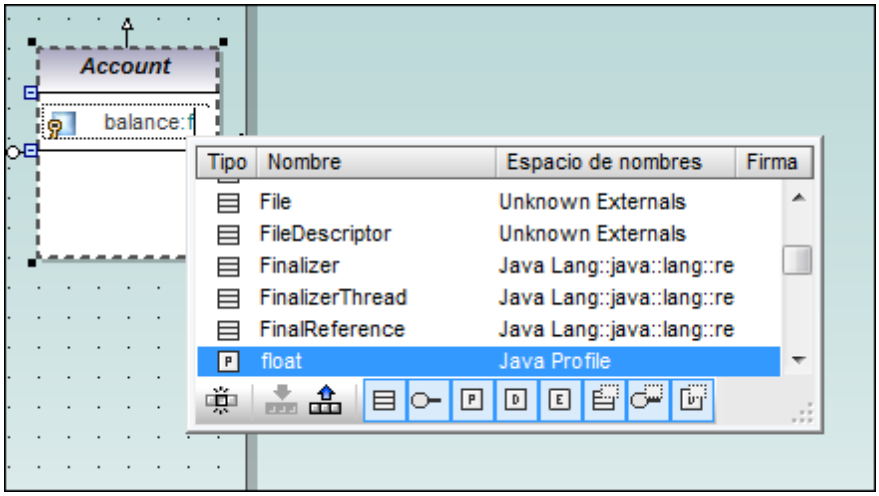


Para agregar propiedades a una clase:

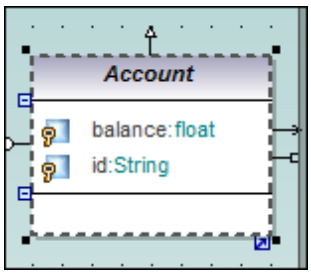
1. Haga clic con el botón derecho en la clase `Account` y seleccione **Nuevo/a | Propiedad** (o pulse la tecla **F7**).
   Se inserta una propiedad predeterminada llamada `Propiedad1`, con los identificadores de estereotipo `<< >>`.



2. Cambie el nombre de la propiedad por `balance` y añada dos puntos "**:**".
   Aparece una lista desplegable con todos los tipos válidos.
3. Escriba la letra **f** y pulse **Entrar** para insertar el tipo de datos de valor devuelto `float`.
   Tenga en cuenta que esta lista desplegable distingue entre mayúsculas y minúsculas.

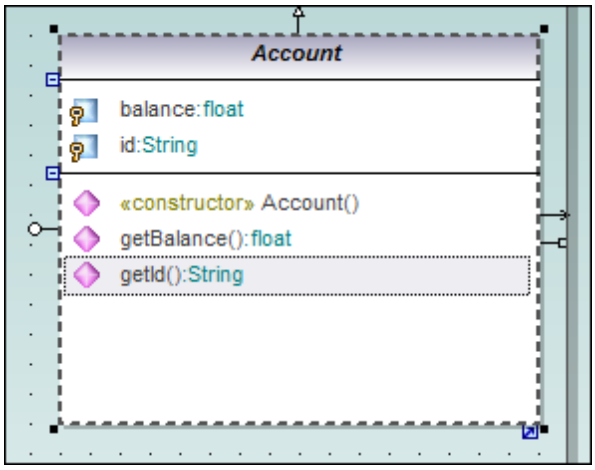


4. En la misma línea añada `=0` para definir el valor predeterminado.
5. Pulse la tecla **F7** para añadir otra propiedad más a la clase.
6. Escriba **Id:** y seleccione `String` en la lista desplegable.



Para agregar operaciones a una clase:

1. Haga clic con el botón derecho en la clase **`Account`** y seleccione **Nuevo/a | Operación** (o pulse la tecla **F8**).
2. Como constructor inserte `Account()`.
3. Ahora siga los mismos pasos que antes y cree dos operaciones más: `getBalance:float` y `getId:String`.

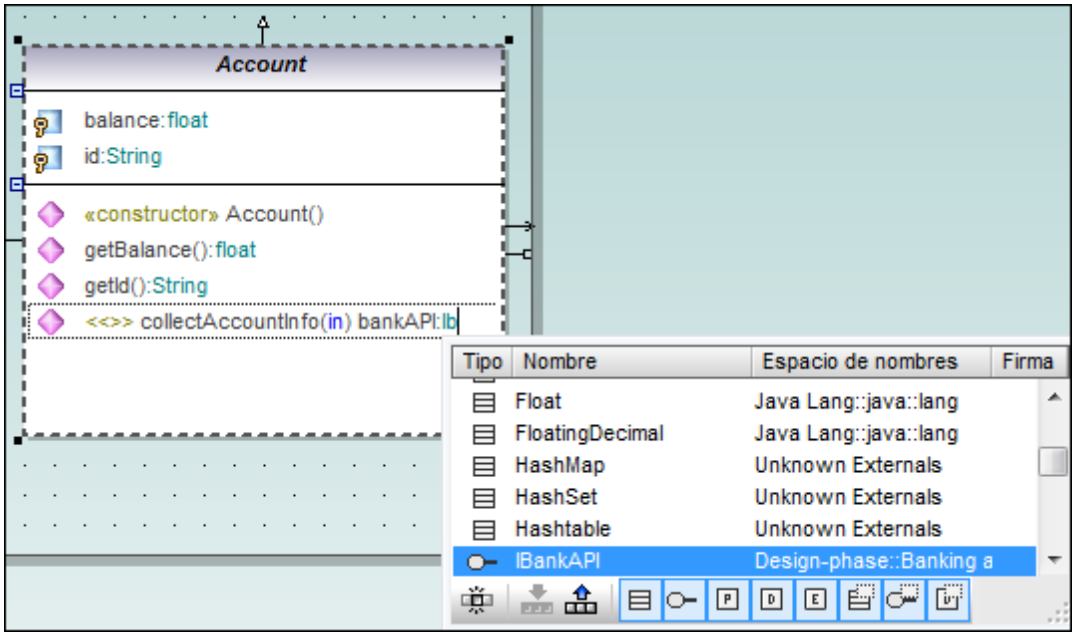


Ahora use la función de finalización automática para definir estas operaciones

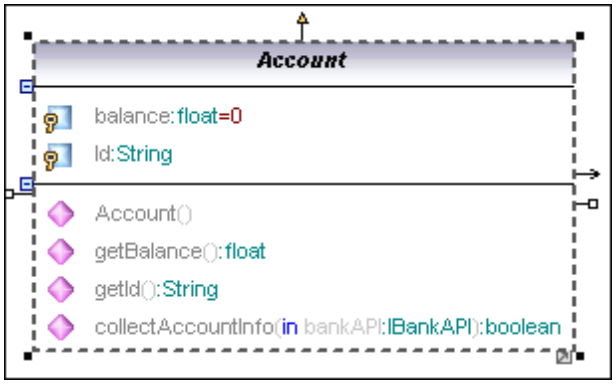
4. Pulse **F8** para crear la operación `collectAccountInfo` y escriba el carácter de

paréntesis "**(**".
Después escriba la letra **i**, lo cual abre la lista desplegable de finalización automática.
Seleccione uno de los parámetros de dirección: `in`, `inout`, `out`.

5. Seleccione `in` en la lista desplegable, inserte un carácter de espaciado y siga editando la misma línea.
6 Ahora escriba `bankAPI` y después dos puntos (:).
7. Seleccione `IBankAPI` de la lista desplegable, cierre el paréntesis con "**)**" y escriba dos puntos otra vez (:).



8. Pulse la tecla **b** para seleccionar el tipo de datos binario y después pulse **Entrar** para insertarlo.
9. Pulse **Entrar** para terminar la definición.



Nota: si hace clic en el icono de visibilidad situado a la izquierda de la operación ( ) o de la propiedad ( ), aparece una lista desplegable donde puede cambiar su nivel de acceso.

Para eliminar propiedades y operaciones de clase de un diagrama de clases:

1. Pulse **F8** y después **Entrar** para agregar la operación predeterminada `Operación1` a la

clase **Account**.

2. Haga clic en Operación1 y pulse **Supr** para eliminarla.

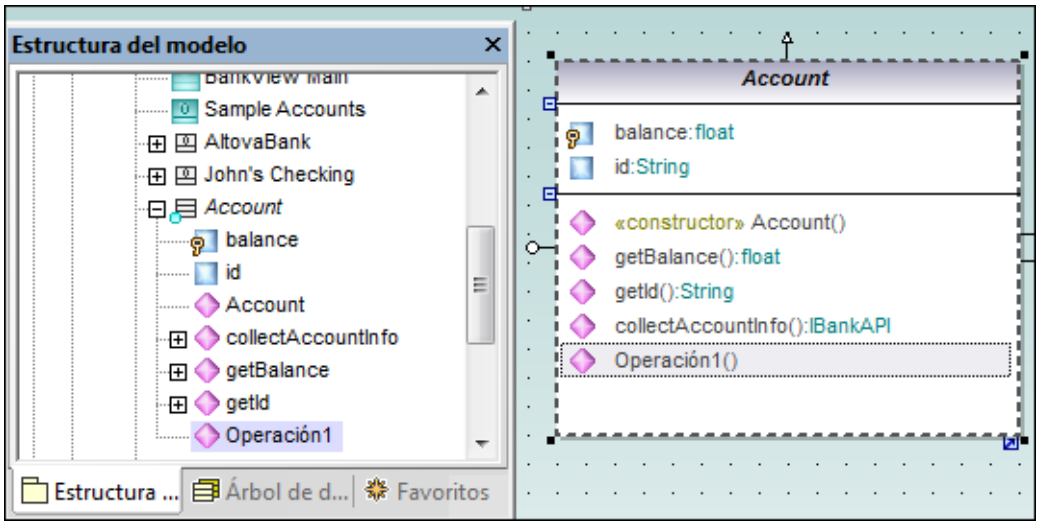
La aplicación pregunta si quiere eliminar el elemento del proyecto. Haga clic en **Sí** para eliminar Operación1 de la **clase** y también del **proyecto**.

Nota: si solo quiere eliminar la operación de la clase, pero no del proyecto, pulse la tecla **Ctrl +Supr**. También puede habilitar un aviso que aparecerá cuando elimine objetos (consulte **Herramientas | Opciones | Edición** para más información).

Para buscar (eliminar) propiedades de clase y opciones desde la *Estructura del modelo*:

Las propiedades y opciones también se pueden eliminar desde la *Estructura del modelo* directamente. Para hacerlo bien es importante buscar primero la propiedad que deseamos eliminar. Imaginemos que insertó la Operación1 en la clase **Account** (pulse **F8** y **Entrar** para insertarla):

1. Ahora haga clic con el botón derecho en Operación1 de la clase **Account**.
2. Seleccione la opción **Seleccionar en la estructura del modelo** (o pulse **F4**).
   El componente Operación1 aparece resaltado debajo de *Account* en la pestaña *Estructura del modelo*.

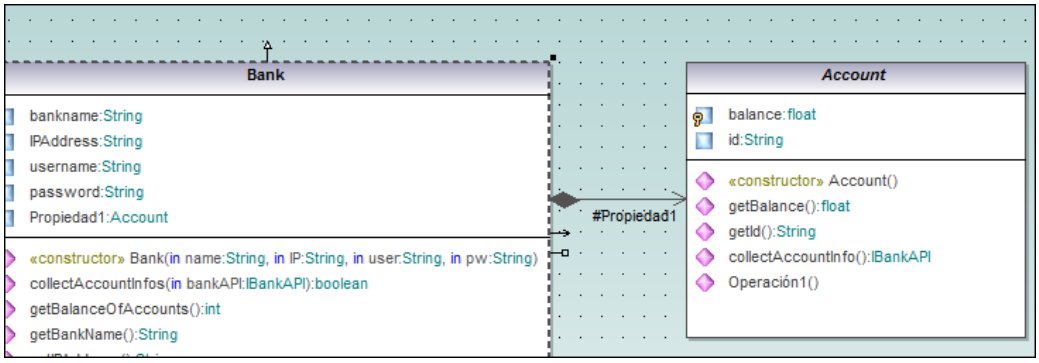


3. Pulse la tecla **Supr** para eliminar la operación de la clase y del proyecto.
   Recuerde que casi todos los elementos de modelado aparecen en la *Estructura del modelo* cuando se pulsa **F4**.

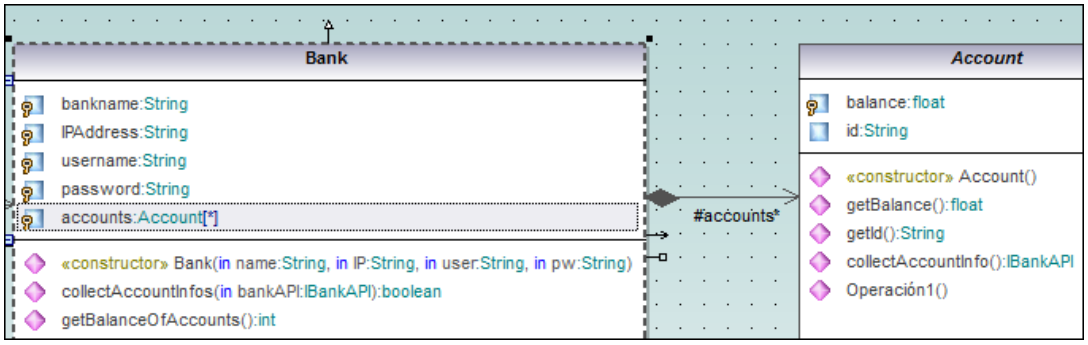
Nota: también puede navegar hasta la *Estructura del modelo* desde el panel *Propiedades*. Para más información consulte el apartado Propiedades de la sección *Interfaz del usuario*.

Crear una asociación de composición entre las clases Bank y Account:

1. Haga clic en el icono **Composición** de la barra de herramientas y cree una conexión entre la clase **Bank** y la clase **Account** arrastrando el puntero del ratón. La clase se resalta cuando la asociación se puede crear.
   En la clase **Bank** se crea una nueva propiedad (Propiedad1:Account) y una flecha de asociación compuesta une las dos clases.



2. Cambie el nombre de `Propiedad1` de la clase **`Bank`** por `accounts`, asegurándose de no eliminar la definición de tipo `Account` (que aparece en color verde).
3. Pulse la tecla **Fin** para poner el cursor al final de la línea.
4. Inserte el corchete "**[**" y seleccione ***** en la lista desplegable para definir la multiplicidad. Para confirmar pulse **Entrar**.



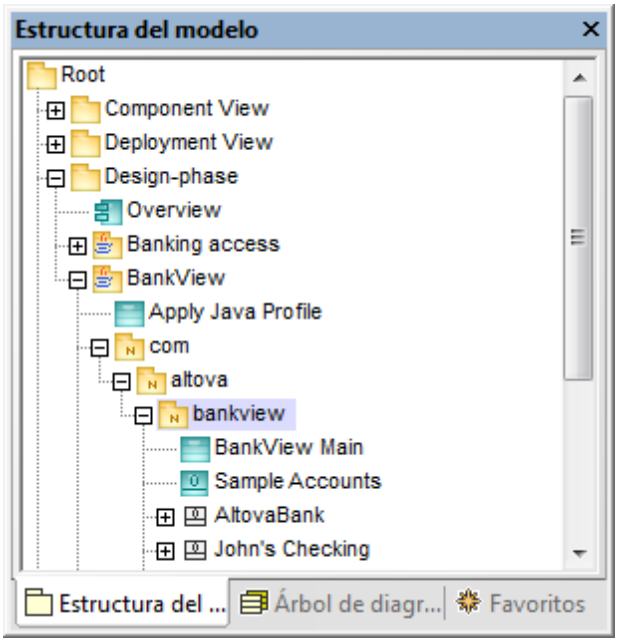
3.3.1 Crear clases derivadas

En este apartado del tutorial aprenderá a:

- Agregar al proyecto una diagrama de clases nuevo llamado **`Account Hierarchy`**
- Insertar clases ya existentes y crear una clase nueva llamada **`Savings account`**
- Crear tres clases **derivadas** de la clase base abstracta **`Account`** por medio de generalizaciones

Para crear un diagrama de clases nuevo:

1. En la *Estructura del modelo* haga clic con el botón derecho en el paquete **`bankview`** (bajo **Design-phase | BankView | com | altova**) y seleccione **Diagrama nuevo | Diagrama de clases**.

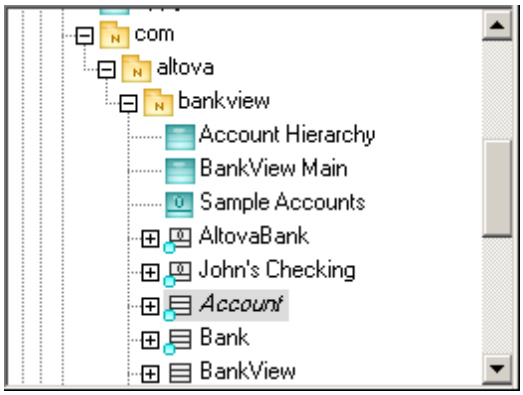


2. Haga doble clic en la nueva entrada **Diagramadeclases1**, llámelo **Account Hierarchy** y pulse **Entrar** para confirmar.

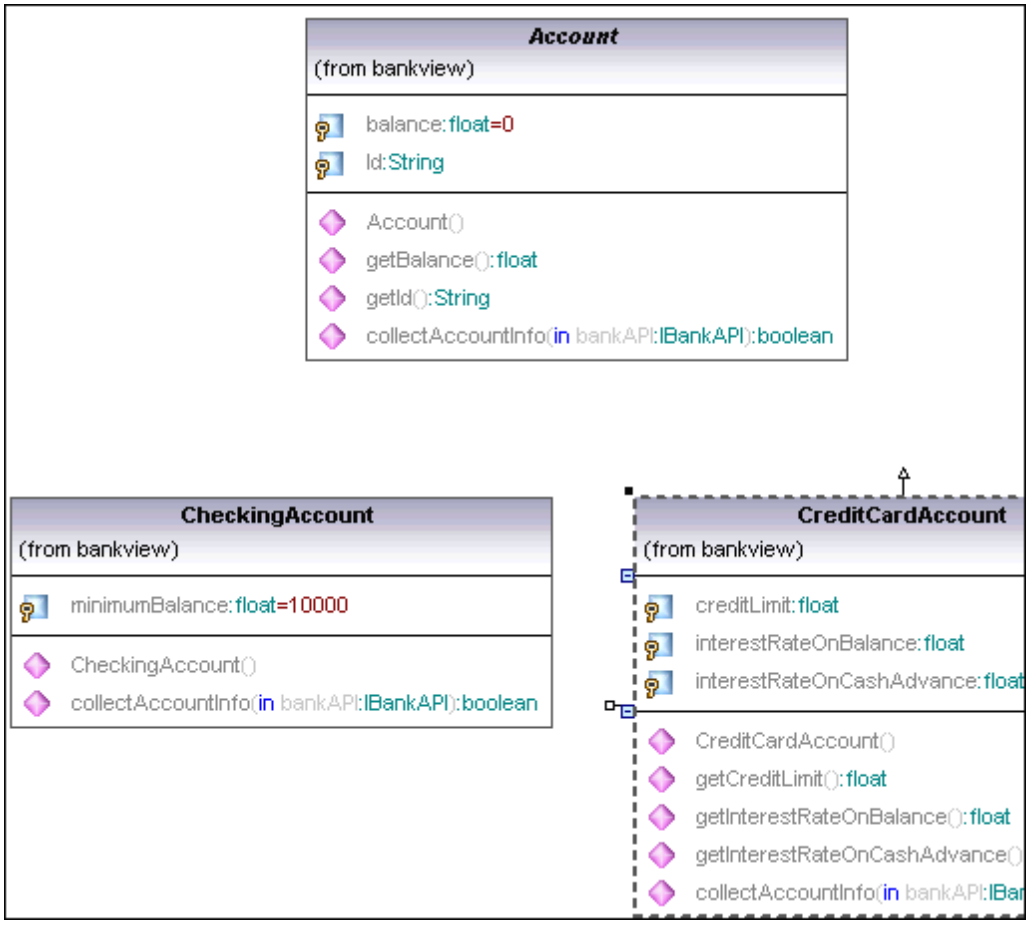
   En el área de trabajo se abre la pestaña del diagrama **Account Hierarchy**.

Insertar en el diagrama clases ya existentes:

1. Haga clic en la clase *Account* del paquete **bankview** (bajo **com | altova | bankview**).

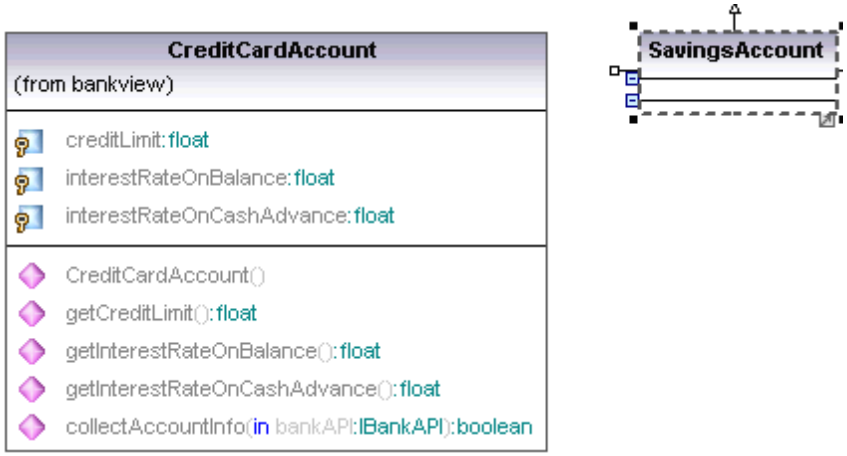


2. Arrástrela hasta el panel del diagrama **Account Hierarchy**.
3. Haga clic en la clase **CheckingAccount** (del mismo paquete) y arrástrela hasta el panel del diagrama **Account Hierarchy**.
4. Ponga la clase debajo de la clase **Account** y a la izquierda.
5. De la misma manera inserte la clase **CreditCardAccount** y póngala a la derecha de la clase **CheckingAccount**.



Agregar una clase nueva:

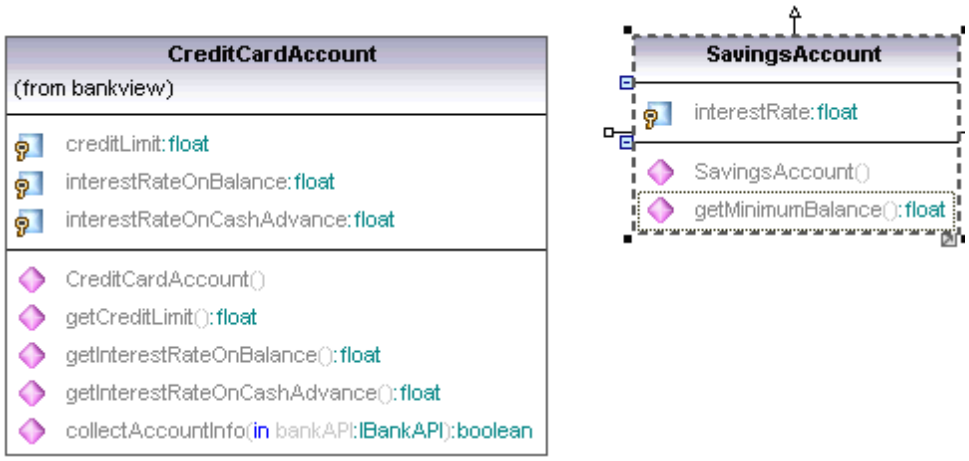
1. Haga clic con el botón derecho en el fondo del área de trabajo (p. ej. a la derecha de **`CreditAccountClass`**) y seleccione **Nuevo/a | Clase**.
   Una clase nueva se añade automáticamente al paquete correspondiente (es decir, **`bankview`**), que contiene el diagrama de clases actual **`Account Hierarchy`**.
2. Haga doble clic en el nombre de la clase y cámbielo por **`SavingsAccount`**.



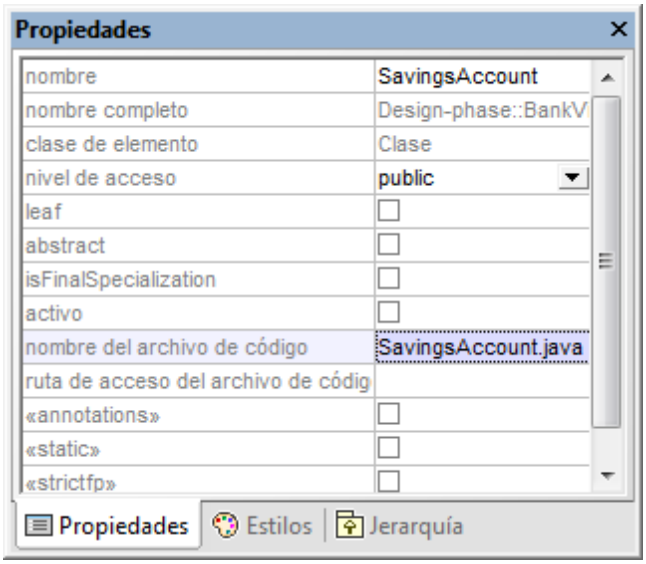
3. Pulse la tecla **F7** para añadir una propiedad nueva.
4. Escriba `interestRate` seguido de dos puntos y pulse la tecla "**f**" para seleccionar el tipo

de datos `float` de la lista desplegable. Pulse dos veces la tecla **Entrar** para seleccionar ese tipo de datos y confirmar su inserción.

5. Ahora pulse la tecla **F8** y añada la operación/el constructor **`SavingsAccount()`**.
6. Pulse **F8** y añada la operación **`getMinimumBalance:float`**.



7. En la pestaña *Propiedades* haga clic en la casilla `nombre del archivo de código` y escriba *SavingsAccount.java* para definir la clase de código Java.

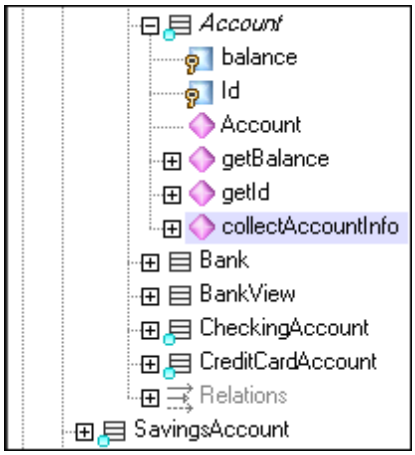


Reutilizar y copiar propiedades/operaciones ya existentes:
En UModel puede copiar (o mover) propiedades y operaciones de una clase a otra. Esto se hace mediante operaciones de arrastrar y colocar o con las teclas de acceso rápido estándar. El ámbito donde se pueden realizar estas operaciones es:

- dentro de una **misma clase** de la pestaña del diagrama
- entre **diferentes clases** de la pestaña del diagrama
- en la vista *Estructura del modelo*
- entre diferentes diagramas UML, soltando los datos copiados en una pestaña de diagrama diferente.

Para más información consulte el apartado Cortar, copiar y pegar en los diagramas de UModel.

1. En la vista *Estructura del modelo* expanda la clase ***Account*** para ver su contenido.
2. Haga clic con el botón derecho en la operación **collectAccountInfo** y seleccione el comando **Copiar**.



3. También en la *Estructura del modelo* haga clic con el botón derecho en la clase **SavingsAccount** y seleccione el comando **Pegar**.
   La operación se copia en la clase **SavingsAccount**, que se expande automáticamente para mostrar la nueva operación.

   La operación nueva se puede ver en la clase **SavingsAccount** del diagrama de clases.

Nota: puede usar las teclas de acceso rápido para copiar/pegar (**Ctrl+C** o **V**) o arrastrar y colocar elementos en la vista Estructura del modelo. En algunos casos quizás sea recomendable deshabilitar las opciones de ordenación para poder colocar las operaciones entre determinados componentes.

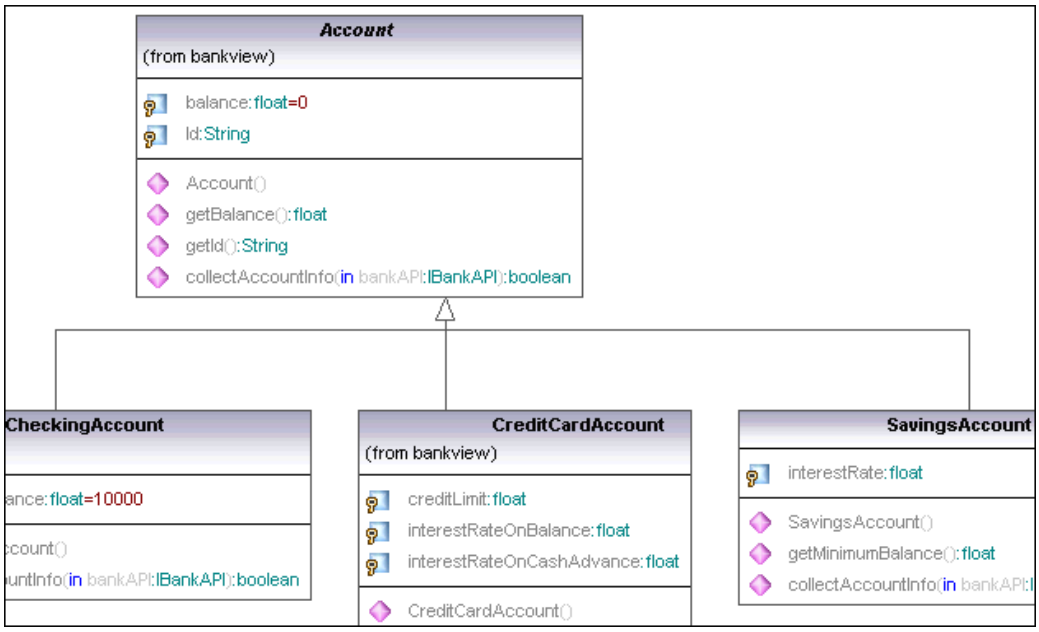
Para crear clases derivadas (generalización/especialización):

Llegados a este punto, el diagrama de clases contiene la clase abstracta ***Account*** y tres clases concretas. Ahora queremos definir, o crear, una relación de generalización/especialización entre ***Account*** y las otras clases. Es decir, queremos crear tres clases concretas derivadas.

1. Haga clic en el icono **Generalización**  de la barra de herramientas y pulse la tecla **Ctrl** al mismo tiempo.
2. Sin dejar de pulsar la tecla **Ctrl**, haga clic en **CreditCardAccount** (la clase que está en el medio del área de trabajo) y arrastre el puntero hasta la clase **Account**.
3. Sin dejar de pulsar la tecla **Ctrl**, haga clic en la clase **CheckingAccount** y arrastre el

puntero hasta la **punta de flecha** de la generalización creada en el paso anterior.

4. Sin dejar de pulsar la tecla **Ctrl**, haga clic en la clase **`SavingsAccount`** y arrastre el puntero hasta la **punta de flecha** de la generalización creada en el paso anterior. Ya puede dejar de pulsar la tecla **Ctrl**.
5. Entre las tres subclases y la superclase **`Account`** se crean flechas de generalización.



3.4 Diagramas de objetos

Sitio web de Altova: Diagramas de objetos UML

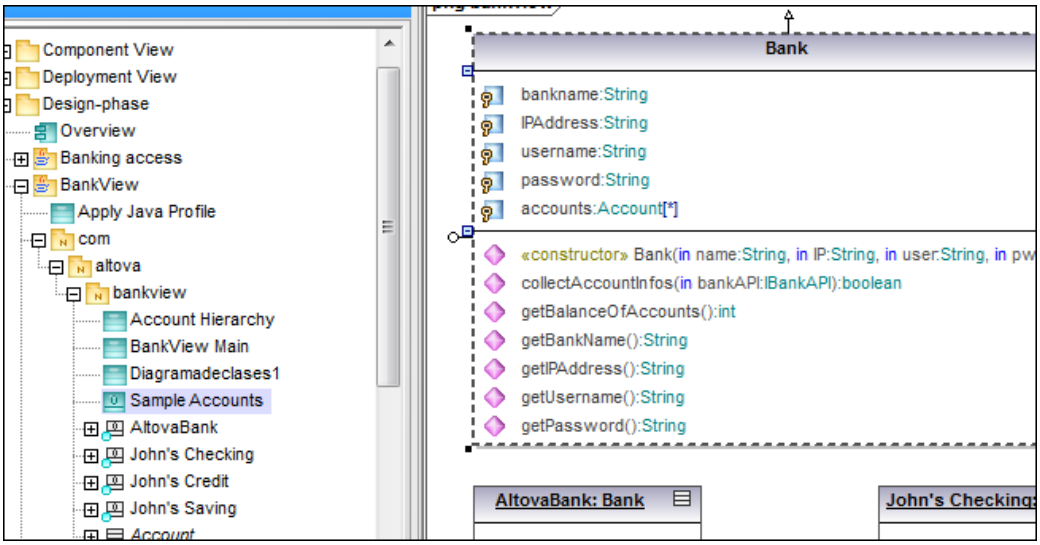
En este apartado del tutorial aprenderá a:

- Combinar **diagramas de clases** y **de objetos** en un solo diagrama para obtener una visión de conjunto de los objetos existentes en cada momento.
- Crear objetos/instancias y definir relaciones entre ellos.
- Dar formato a las asociaciones y vínculos.
- Introducir datos reales en objetos e instancias.

Para abrir el diagrama de objetos:

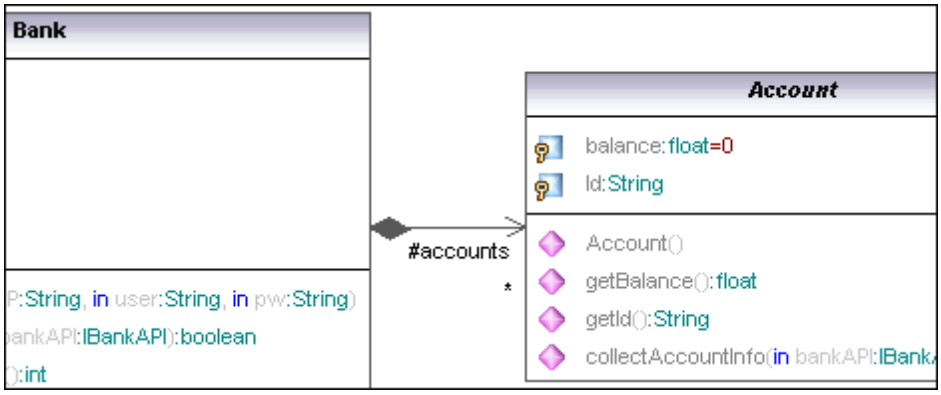
1. En la vista *Estructura del modelo* haga doble clic **en el icono** del diagrama **`Sample Accounts`** del paquete **`bankview`**.

   `AltovaBank:Bank` es el objeto/la instancia de la clase **`Bank`**, mientras que **`John's checking: CheckingAccount`** es una instancia de la clase **`CheckingAccount`**.



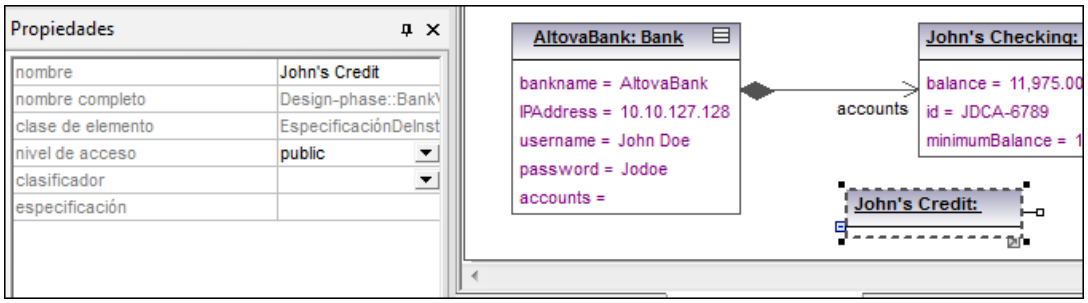
Para insertar una clase en un diagrama de objetos:

- En la *Estructura del modelo* haga clic en el icono de la clase *Account* y arrástrelo hasta la pestaña del diagrama **`Sample Accounts`**.
  La asociación compuesta definida previamente en el diagrama **`BankView Main`** se crea automáticamente.



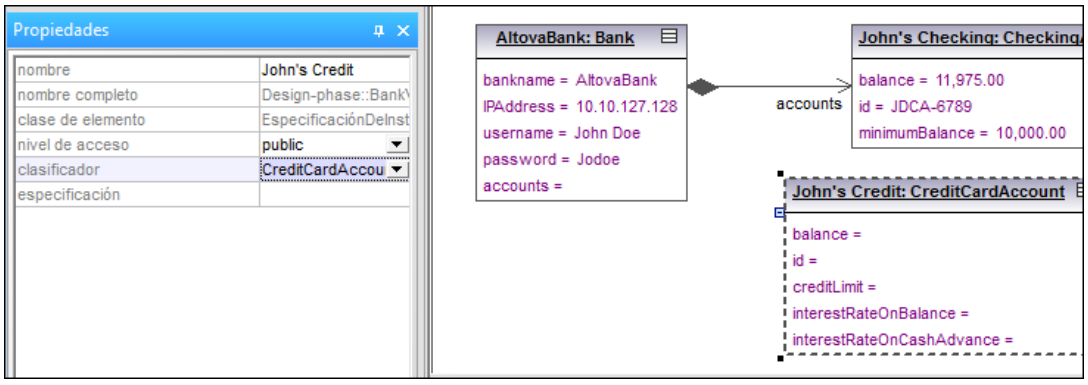
Para agregar un objeto nuevo / una instancia nueva seleccionando su tipo:

1. Haga clic en el icono **EspecificaciónDeInstancia**  de la barra de herramientas y después haga clic justo debajo del objeto `John's Checking` en el área de trabajo.
2. Cambie el nombre de la instancia por `John's Credit` y pulse la tecla **Entrar**.



Si la instancia está activa, todas sus propiedades aparecen en la pestaña *Propiedades*.

3. Haga clic en el cuadro combinado `clasificador` y seleccione la opción **CreditCardAccount** de la lista desplegable.

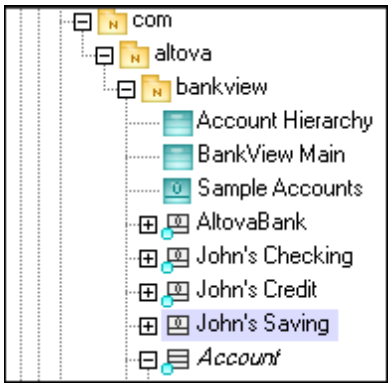


Puede ocultar/mostrar el contenido de un objeto haciendo clic con el botón derecho en la especificación de instancia y seleccionando el comando **Mostrar/ocultar el contenido del nodo**.

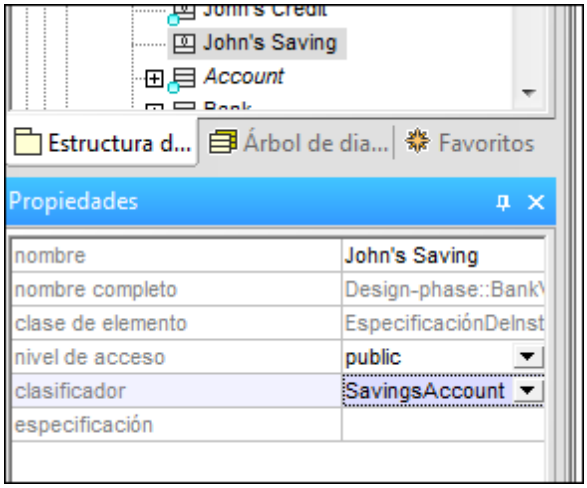
Para agregar un objeto nuevo en la vista *Estructura del modelo* (y después insertarlo en

el diagrama):

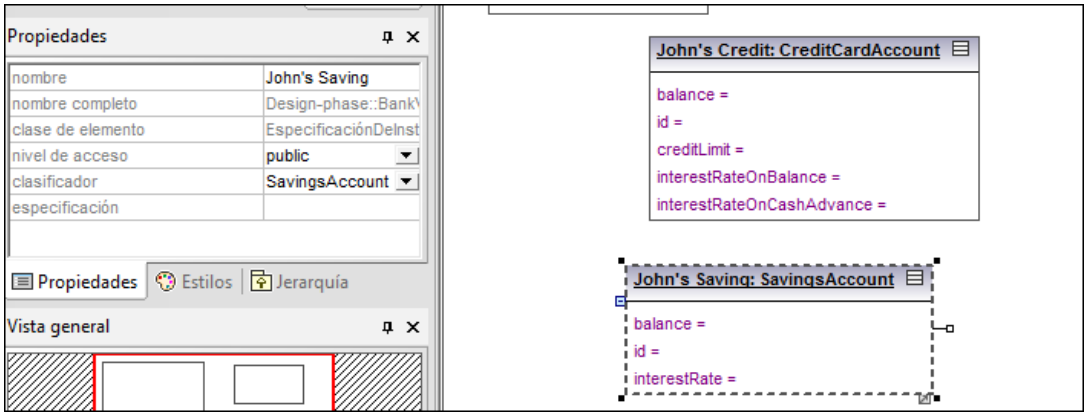
1. En la *Estructura del modelo* haga clic con el botón derecho en el paquete **`bankview`** y seleccione el comando **Elemento nuevo | EspecificaciónDeInstancia**.
2. Cambie el nombre predeterminado del objeto por **`John's Saving`** y pulse **Entrar**. El objeto nuevo se añade al paquete y se coloca en la posición correcta.



3. Compruebe que el objeto está seleccionado en la pestaña Estructura del modelo. Haga clic en el cuadro combinado `clasificador` de la pestaña *Propiedades* y seleccione la opción **SavingsAccount**.



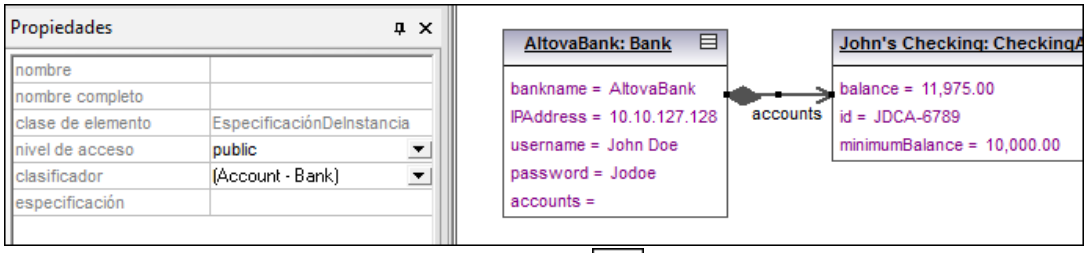
4. Ahora arrastre el objeto/la instancia **`John's Saving`** desde la *Estructura del modelo* hasta el área de trabajo (del diagrama la pestaña **`Sample Accounts`**) y colóquelo debajo de **`John's Credit`**.



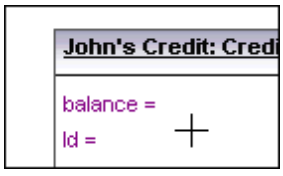
Para crear vínculos entre los objetos:

Los *vínculos* son las instancias de asociaciones de clases y describen las relaciones que existen entre los objetos/las instancias en un momento dado.

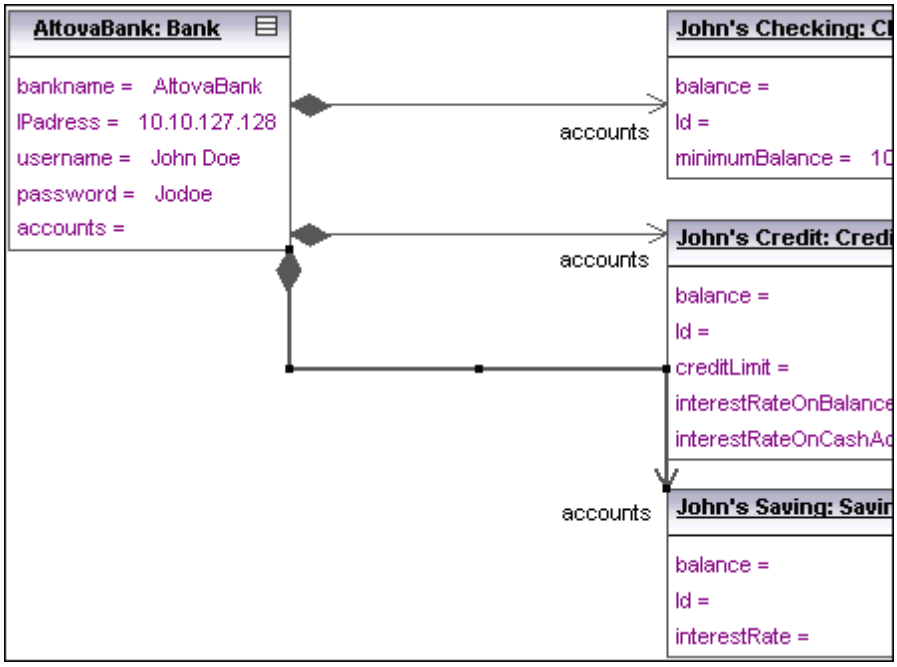
1. Haga clic en el vínculo (asociación) que existe entre **`AltovaBank`** y **`John's Checking`**.
2. En la pestaña *Propiedades* haga clic en el cuadro combinado `clasificador` y seleccione la entrada **Account - Bank.**
   El vínculo se convierte en una asociación compuesta, de acuerdo con las definiciones de clase.



3. Haga clic en el icono **EspecificaciónDeInstancia**  de la barra de herramientas y pase el cursor por encima de la clase **`John's Credit`**.
   El cursor ahora tiene forma de +.



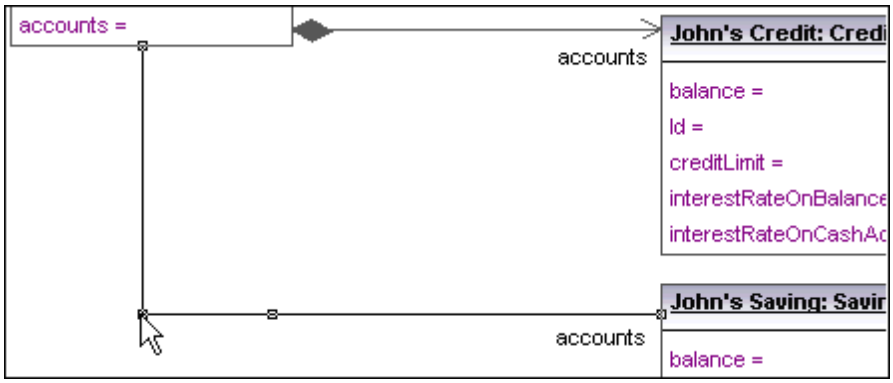
4. Haga clic en el objeto **`John's Credit`** y arrastre el puntero hasta **`AltovaBank`** para crear un vínculo entre ellos.
5. En el cuadro combinado `clasificador` de la pestaña *Propiedades* cambie el tipo de vínculo a **Account - Bank**.
6. De la misma manera cree un vínculo entre **`John's Saving`** y **`AltovaBank`**.



Nota: si cambia el tipo de asociación en un diagrama de clases, el tipo de asociación se actualiza automáticamente en el diagrama de objetos.

Para dar formato a las líneas de asociación/vínculo del diagrama:

1. Haga clic en el vínculo situado en la parte inferior del diagrama y arrastre el conector de la esquina a la izquierda.
   Esto sirve para ajustar la posición horizontal y vertical de la línea de asociación.

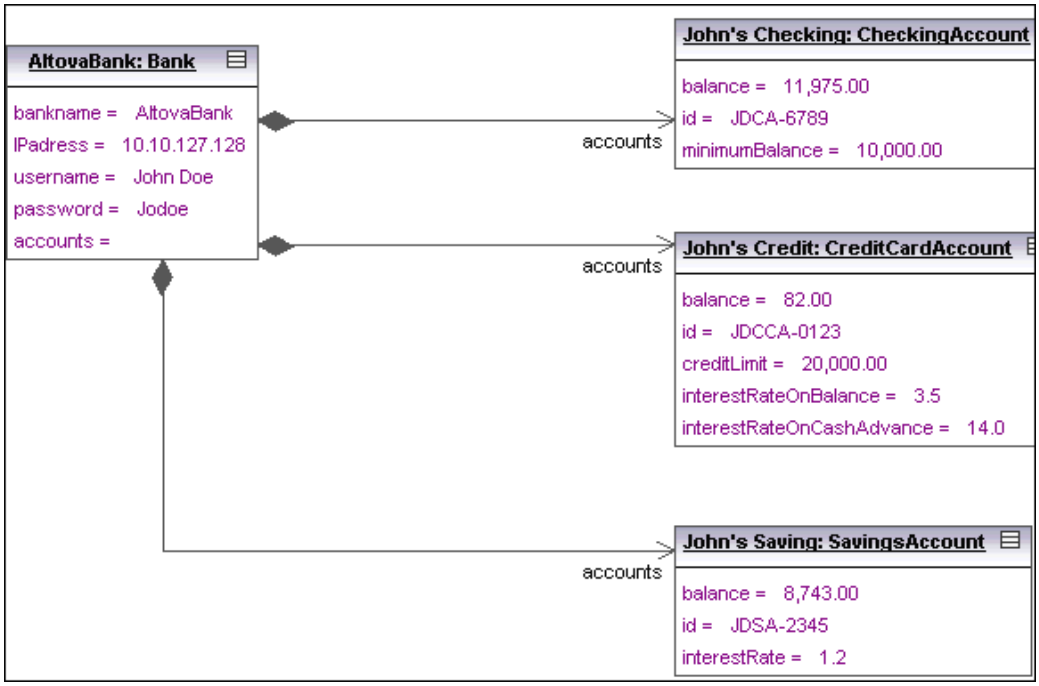


   Ajuste la posición de los vínculos del diagrama siguiendo este método.

Para introducir datos de muestra en los objetos:

El valor de instancia de un atributo o de una propiedad de un objeto recibe le nombre de ***slot***.

1. Haga clic en los slots correspondientes e introduzca datos de muestra.
2. Por ejemplo, en el objeto **`John's Checking`** haga doble clic en el slot `balance` y escriba la cifra `11.975,00`.
3. Rellene el resto de los datos para tener una idea del estado actual de la instancia.


AltovaBank: Bank
bankname = AltovaBank
IPadress = 10.10.127.128
username = John Doe
password = Jodoe
accounts =
John's Checking: CheckingAccount
balance = 11,975.00
id = JDCA-6789
minimumBalance = 10,000.00
John's Credit: CreditCardAccount
balance = 82.00
id = JDCCA-0123
creditLimit = 20,000.00
interestRateOnBalance = 3.5
interestRateOnCashAdvance = 14.0
John's Saving: SavingsAccount
balance = 8,743.00
id = JDSA-2345
interestRate = 1.2
accounts
accounts
accounts

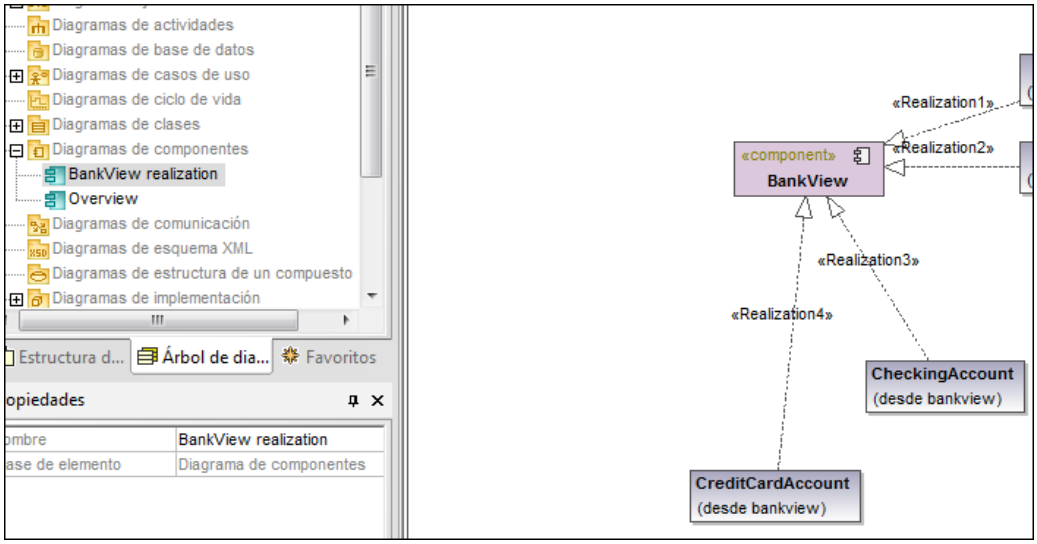
3.5 Diagramas de componentes

En este apartado del tutorial aprenderá a:

- Insertar clases en un diagrama de componentes
- Crear dependencias de realización entre las clases y el componente **`BankView`**
- Cambiar las propiedades de las líneas
- Insertar componentes en un diagrama de componentes y crear dependencias de utilización en una interfaz

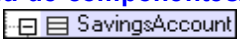
Para abrir el diagrama de componentes:

1. Haga clic en la pestaña *Árbol de diagramas*, expanda el nodo `Diagramas de componentes` y haga doble clic **en el icono** del diagrama **`BankView realization`**.
   En el área de trabajo se abre el diagrama de componentes **`BankView realization`**.

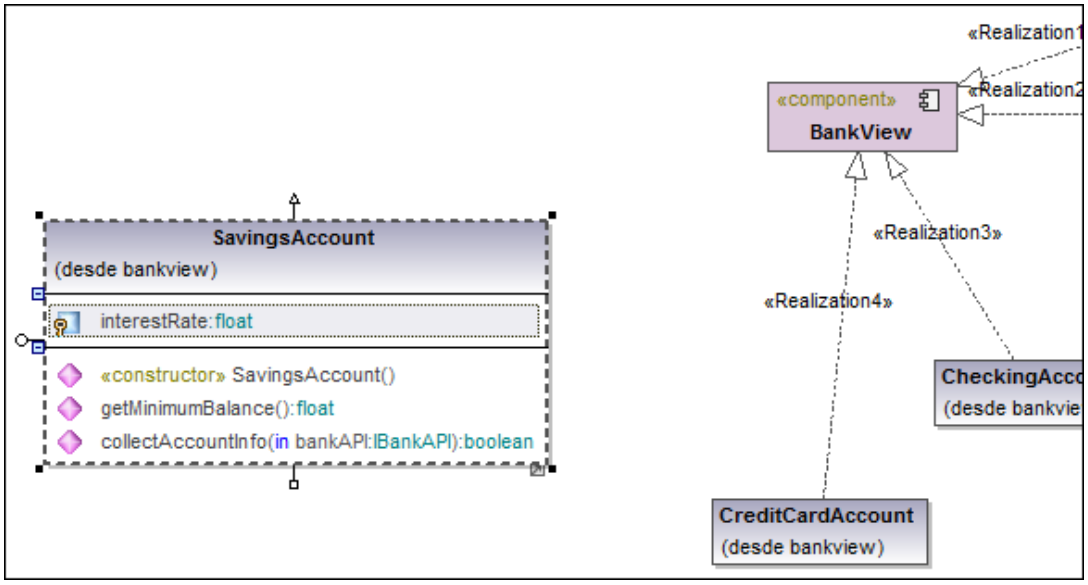


2. Vuelva a la pestaña *Estructura del modelo*.

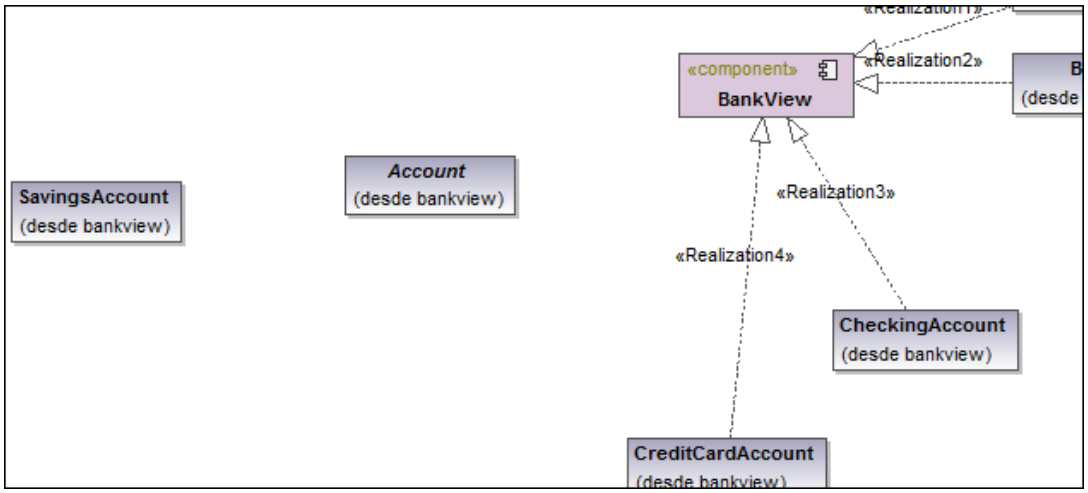
Para insertar clases ya existentes en un diagrama de componentes:

1. En la *Estructura del modelo* busque la clase 

(situada bajo el paquete **`bankview`**).
2. Arrastre la clase **`SavingsAccount`** hasta el diagrama de componentes.
   La clase aparece con todos sus compartimentos abiertos.



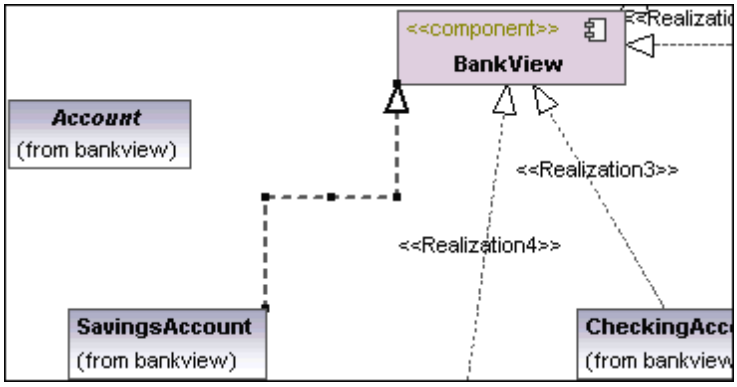
3. Haga clic en los iconos - para contraer los compartimentos de la clase (hasta que solamente se vea el nombre de la clase)
4. De la misma manera inserte la clase abstracta **`Account`**.



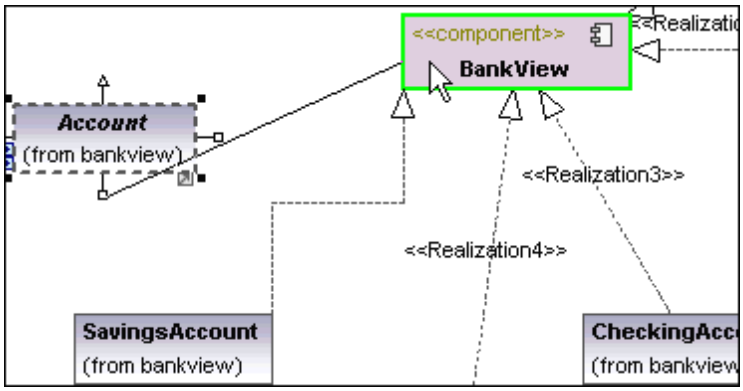
Nota: observe que el nombre del paquete que contiene las clases insertadas aparece en el compartimento que incluye el nombre de la clase (p. ej. *desde bankview*).

Para crear dependencias de realización entre una clase y un componente:

1. Haga clic en el icono **Realización** de la barra de herramientas.
2. Haga clic en **`SavingsAccount`** y arrastre el puntero hasta el componente **`BankView`**.



3. Haga clic en el controlador **RealizaciónDeComponente** de la clase ***Account*** (en la parte inferior) y arrastre el puntero del ratón hasta el componente **BankView**.

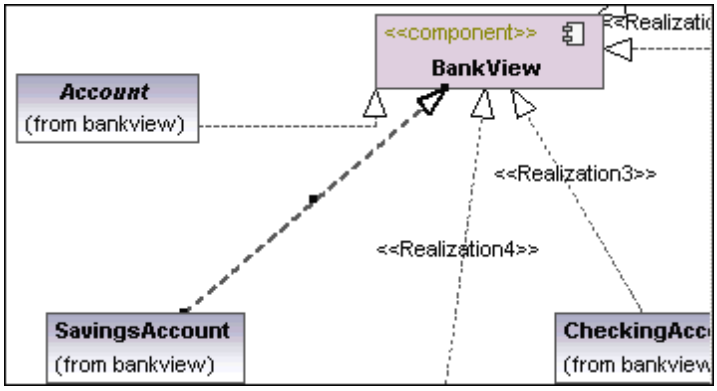


Estos dos métodos se pueden utilizar para crear dependencias de realización. Hay otro método más que permite crear dependencias de realización en la Estructura del modelo solamente. Para más información consulte el apartado Ingeniería de ida y vuelta (código - modelo - código).

Para cambiar las características de las líneas (de realización):

Al hacer clic en una dependencia o cualquier otro tipo de línea en un diagrama de UModel, se habilitan los comandos de la barra de herramientas Diseño que sirven para dibujar líneas.

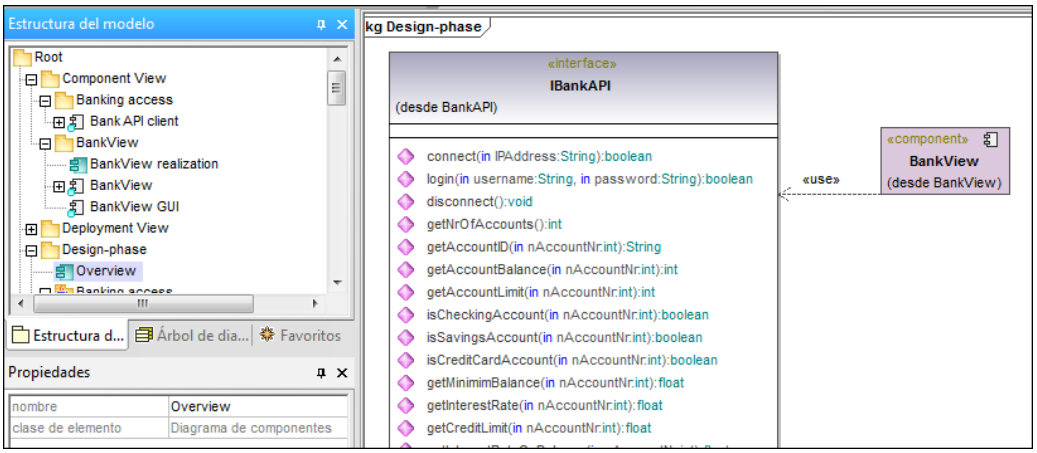
1. Haga clic en la línea de realización que une **SavingsAccount** con el componente **BankView**.
2. Haga clic en el icono **Línea directa**  de la barra de herramientas Diseño.



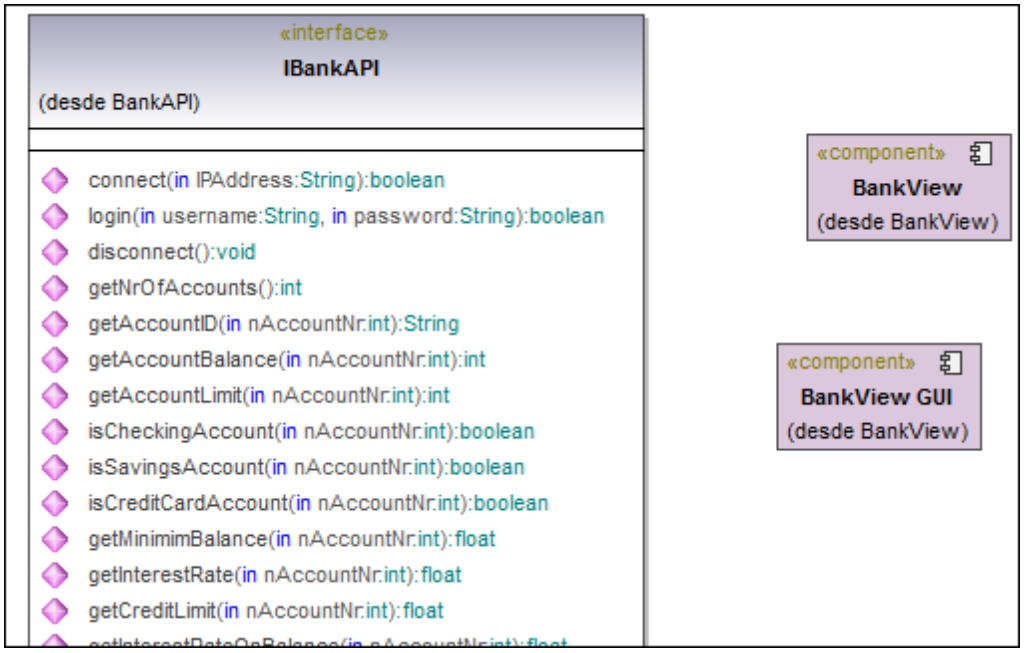
Las propiedades de la línea se modifican inmediatamente. Las líneas tienen pequeños iconos denominados **puntos de referencia**. Puede hacer clic en estos puntos de referencia y moverlos para modificar las características de la línea.

Para insertar componentes y crear dependencias de utilización:

1. En la *Estructura del modelo* haga doble clic **en el icono** del diagrama **`Overview`**, situado justo debajo del paquete **`Design-phase`**.
   El diagrama de componentes **`Overview`** se abre en el área de trabajo y muestra las dependencias de sistema definidas hasta ahora entre componentes e interfaces.



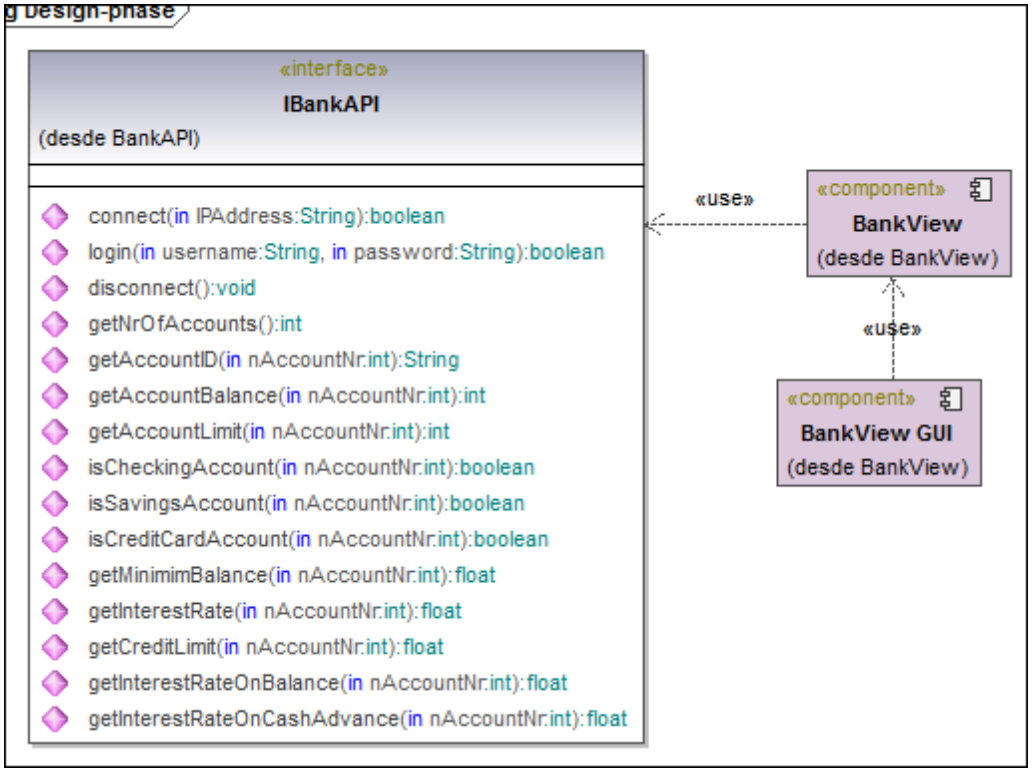
2. En la *Estructura del modelo* haga clic en el componente **`BankView GUI`** situado justo debajo del paquete **`Component View | BankView`** y arrástrelo hasta la pestaña del diagrama **`Overview`**.
   El paquete que contiene el componente insertado aparece en el compartimento del nombre (es decir, desde `BankView`).
3. De la misma manera inserte el componente **`BankView`**.



El componente **`BankView`** es el componente generado por el proceso de ingeniería directa descrito en este tutorial.

Para crear dependencias de utilización entre interfaces y componentes:

1. Haga clic en el icono **Utilización**  de la barra de herramientas.
2. Ahora haga clic en el componente **`BankView GUI`** y arrastre el puntero hasta el componente **`BankView`**.
3. Haga clic otra vez en el icono **Utilización**. Ahora haga clic otra vez en el componente **`BankView`** y arrastre el puntero hasta la interfaz **`IBankAPI`**.



La dependencia de utilización (<<use>>) conecta un elemento **cliente** con un elemento **proveedor**. En este caso la interfaz **`IBankInterfaceAPI`** utiliza los servicios de los componentes **`BankView`** y **`BankView GUI`**.

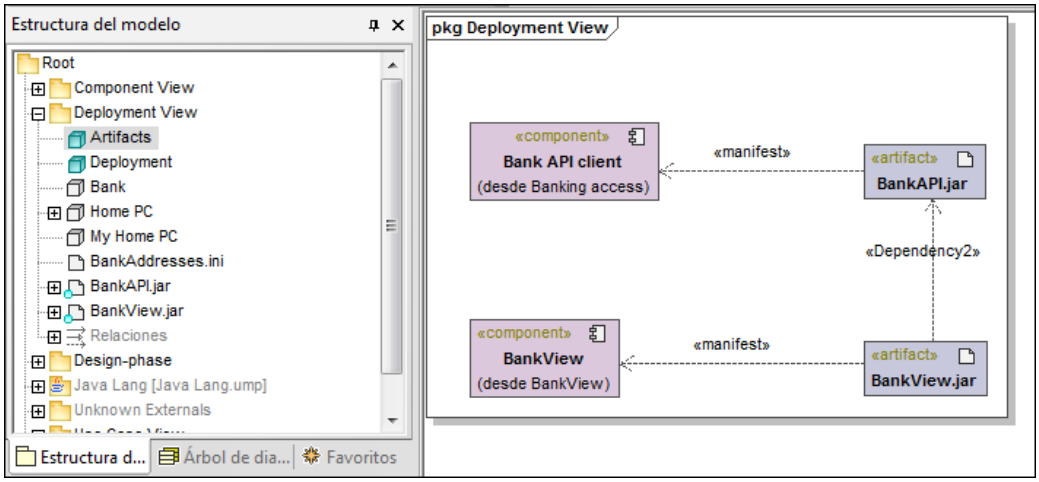
3.6 Diagramas de implementación

En este apartado del tutorial aprenderá a:

- Ver las manifestaciones artefacto de los componentes
- Agregar un nodo y una dependencia nuevos al diagrama de implementación
- Agregar artefactos a un nodo y crear relaciones entre ellos

Para abrir el diagrama de implementación Artifacts:

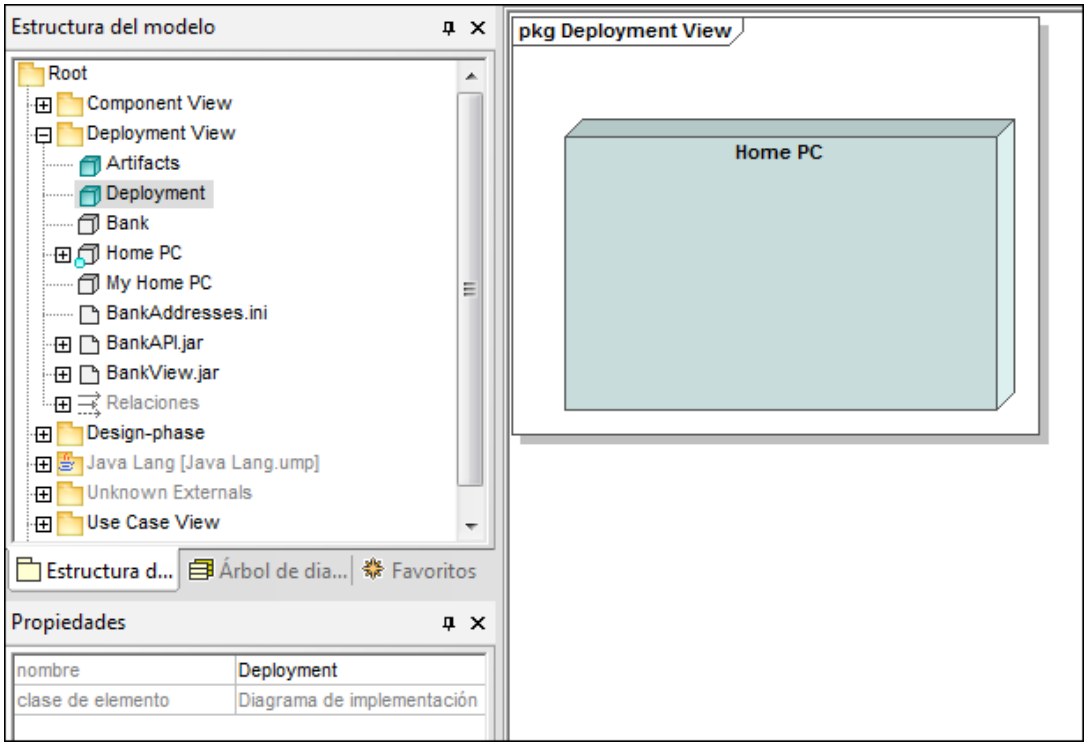
1. En la *Estructura del modelo* expanda el paquete de diagramas **`Deployment View`** y haga doble clic **en el icono** del diagrama **`Artifacts`**.



Este diagrama muestra la manifestación de los componentes **`Bank API client`** y **`BankView`** hacia sus correspondientes archivos Java .jar compilados.

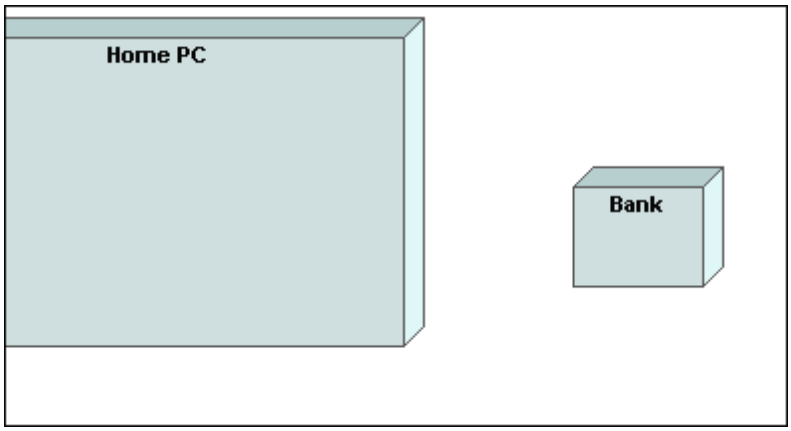
Para abrir el diagrama de implementación Deployment:

1. En la *Estructura del modelo* expanda el paquete de diagramas **`Deployment View`** y haga doble clic **en el icono** del diagrama **`Deployment`**.
   El diagrama se abre en el área de trabajo y muestra la arquitectura física del sistema, que actualmente solo está formado por el nodo **`Home PC`**.



Para agregar un nodo al diagrama de implementación:

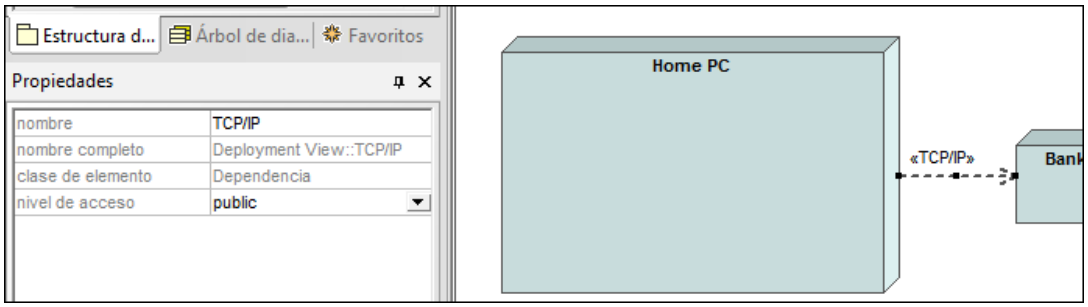
1. Haga clic en el icono **Nodo** de la barra de herramientas y después haga clic en el nodo **`Home PC`**.
2. Cambie el nombre predeterminado del nodo por **`Bank`** y arrastre sus bordes para aumentar un poco su tamaño.



Para crear una dependencia entre dos nodos:

1. Haga clic en el icono **Dependencia** . Haga clic en el nodo **`Home PC`** y arrastre el puntero hasta el nodo **`Bank`**.
   Esto crea una dependencia entre los dos nodos.
2. En la pestaña *Propiedades* haga clic en el campo `nombre` y escriba *TCP/IP*. Pulse

Entrar para confirmar.
El nombre de la dependencia aparece encima de la línea de dependencia.

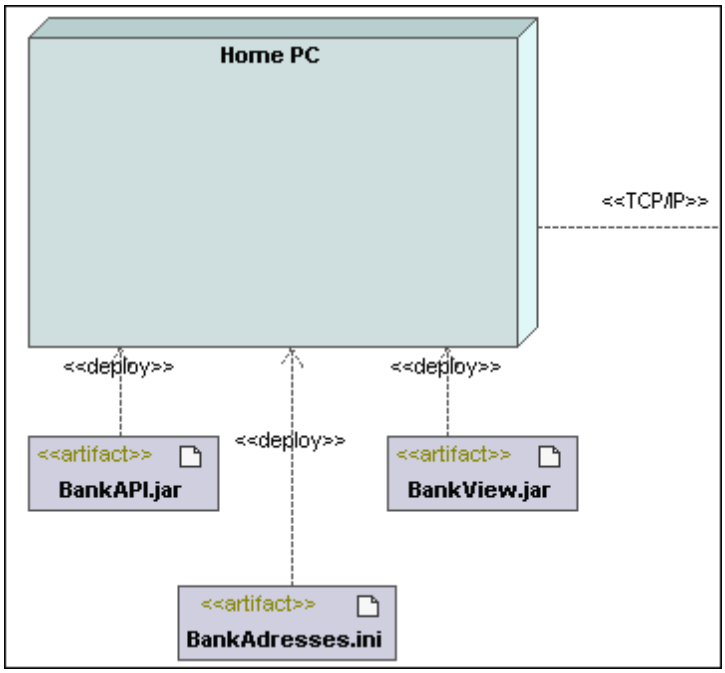


Nota: tras hacer clic en la flecha de la dependencia (o en cualquier elemento con nombre) puede teclear directamente texto nuevo, sin necesidad de hacer clic primero en el campo `nombre` de la pestaña *Propiedades*.

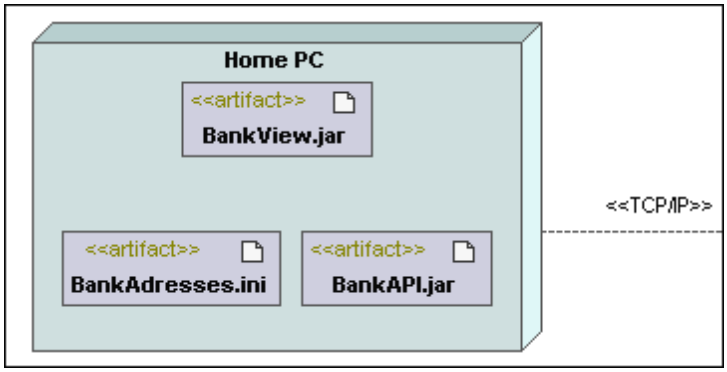
Para agregar artefactos a un nodo y crear dependencias entre ellos:

Expanda el paquete **`Deployment View`** de la *Estructura del modelo*:

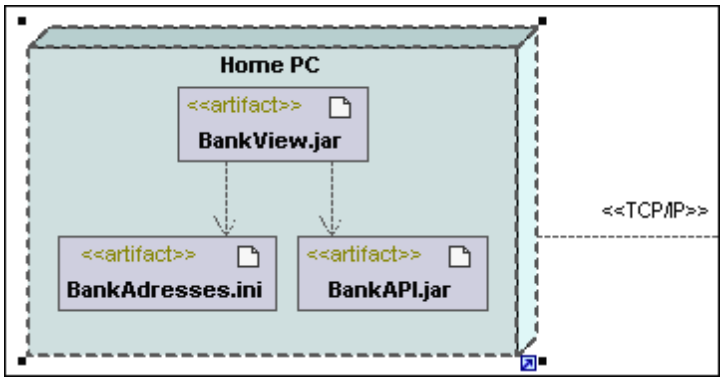
1. Ahora haga clic en los artefactos **`BankAddresses.ini`**, **`BankAPI.jar`** y **`BankView.jar`** y arrástrelos uno a uno hasta el área de trabajo del diagrama. Aparecen las dependencias de cada artefacto.



2. Haga clic en el artefacto **`BankView.jar`** y arrástrelo hasta el nodo **`Home PC`**.
   Cuando esté permitido soltar el artefacto, el nodo de destino aparece resaltado.
3. Haga lo mismo para arrastrar los demás artefactos hasta el nodo **`Home PC`**.
   Ahora los artefactos forman parte del nodo y se mueven con él si lo cambiamos de posición.



4. Haga clic en el icono **Dependencia** de la barra de herramientas y pulse la tecla **Ctrl**.
5. Sin dejar de pulsar la tecla **Ctrl**, haga clic en el artefacto `BankView.jar` y arrastre el puntero hasta el artefacto `BankAddresses.ini`.
6. Sin dejar de pulsar la tecla **Ctrl**, haga clic en el artefacto `BankView.jar` y arrastre el puntero hasta el artefacto `BankAPI.jar`.



Nota: si arrastra un artefacto fuera del nodo, se crea una dependencia de implementación automáticamente.

Para eliminar un artefacto de un nodo y del proyecto:

- Haga clic en el artefacto que desea eliminar y pulse la tecla **Supr**.

  El artefacto y las dependencias se eliminan del nodo y del **proyecto**.

Para eliminar un artefacto de un nodo y de su diagrama:

1. Arrastre el artefacto fuera del nodo.
2. Pulse **Ctrl+Supr**.

   El artefacto y sus dependencias se eliminan del **diagrama** actual, pero no del proyecto.

Para crear operaciones / propiedades o anidar artefactos:

1. Haga clic **con el botón derecho** en el artefacto en la *Estructura del modelo*.
2. Seleccione la acción correspondiente en el menú contextual (p. ej. **Elemento nuevo** | **Operación** / **Propiedad** / **Artefacto**)
   El elemento nuevo aparece bajo el artefacto seleccionado en la *Estructura del modelo*.

3.7 Ingeniería de ida y vuelta (modelo - código - modelo)

Sitio web de Altova: Generación de código UML e ingeniería de código UML de ida y vuelta

En este apartado del tutorial aprenderá a:

- Revisar la sintaxis del proyecto
- Generar código de proyecto
- Agregar código externo de método nuevo (a la clase **SavingsAccount**)
- Sincronizar el modelo de UModel con el código nuevo

Sincronizar paquetes con el código / modelo:
El código se puede combinar/sincronizar a varios niveles:

- A nivel de proyecto / paquete raíz (con ayuda de un comando de menú)
- A nivel de paquete (es posible seleccionar varios paquetes y generar código a partir de ellos)
- A nivel de clase (es posible seleccionar varias clases y generar código a partir de ellas)
- A nivel de componente

El diagrama **BankView** muestra cómo se realiza el componente **BankView** por medio de sus seis clases constituyentes. Este es el componente que se obtiene al seguir las instrucciones para ingeniería directa del tutorial.

Requisitos para poder generar código:

- El componente debe estar **realizado** por una clase como mínimo.
- El componente debe tener asignada una ubicación física (es decir, un directorio). El código generado se guarda en ese directorio.
- Para poder incluirlos en el proceso de generación de código, los componentes deben establecerse por separado.
- Debe definirse un paquete raíz de espacio de nombres Java, C# o VB.

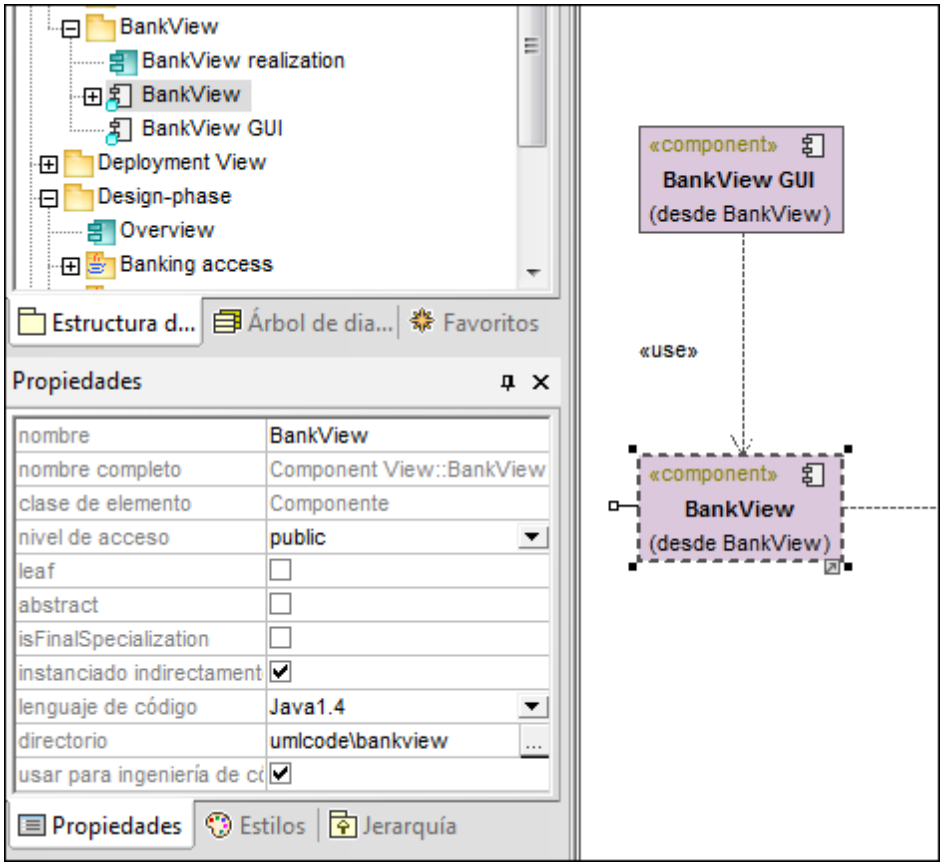
Nota: la raíz del espacio de nombres Java se definió en el paquete **Design-phase | BankView | com**, en la *Estructura del modelo*.

En un mismo proyecto puede combinar código Java, C# y VB y UModel se ocupa del código automáticamente durante el proceso de ingeniería de ida y vuelta. Por ejemplo, el archivo de proyecto `Bank_MultiLanguage.ump` de la carpeta de ejemplos `...\UModelExamples` incluye código Java y C#.

Para definir el directorio de destino para la generación de código:

1. En la *Estructura del modelo* haga doble clic en el icono de **Overview**, justo debajo del paquete **Design-phase**.
2. Ahora en el diagrama haga clic en el componente **BankView** y observe su configuración en la pestaña *Propiedades*.
3. Haga clic en el botón ... situado a la derecha del campo `directorio`.
4. Introduzca/seleccione el directorio de destino en el cuadro de diálogo (p. ej. `DirectorioInstalación\UModelExamples\Tutorial\umlcode\bankview`) o haga clic

en el botón **Carpeta nueva** para crear una carpeta nueva.
La ruta de acceso aparece ahora en el campo `directorio`.

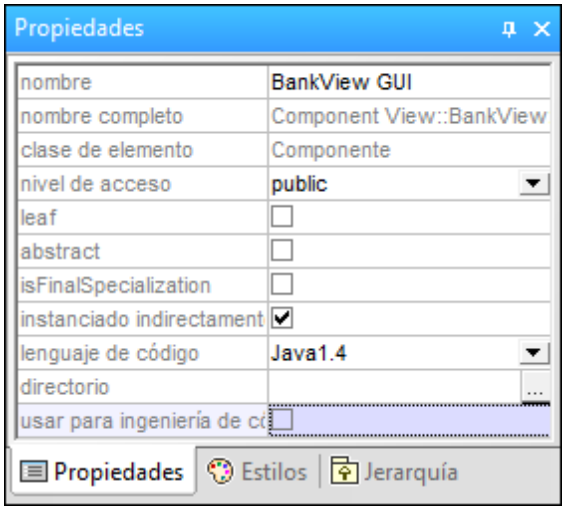


Nota: en UModel el código Java crea código en un directorio u otro dependiendo de su espacio de nombres (p. ej. `...\code\namespace1\C1.java`).

Si quiere usar la misma norma para nombrar directorios para C# y VB .NET, seleccione la opción de menú **Herramientas | Opciones | Ingeniería de código** y elija las opciones correspondientes en el grupo de opciones *Usar espacio de nombres para la ruta del archivo de código*.

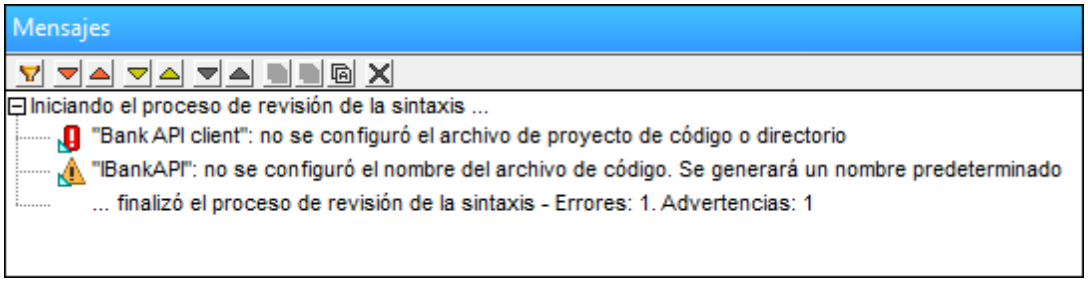
Para incluir/excluir componentes en la generación de código:

1. Haga clic en el componente **`BankView GUI`**.
2. Si está marcada, desactive la casilla `usar para ingeniería de código` de la pestaña *Propiedades*.

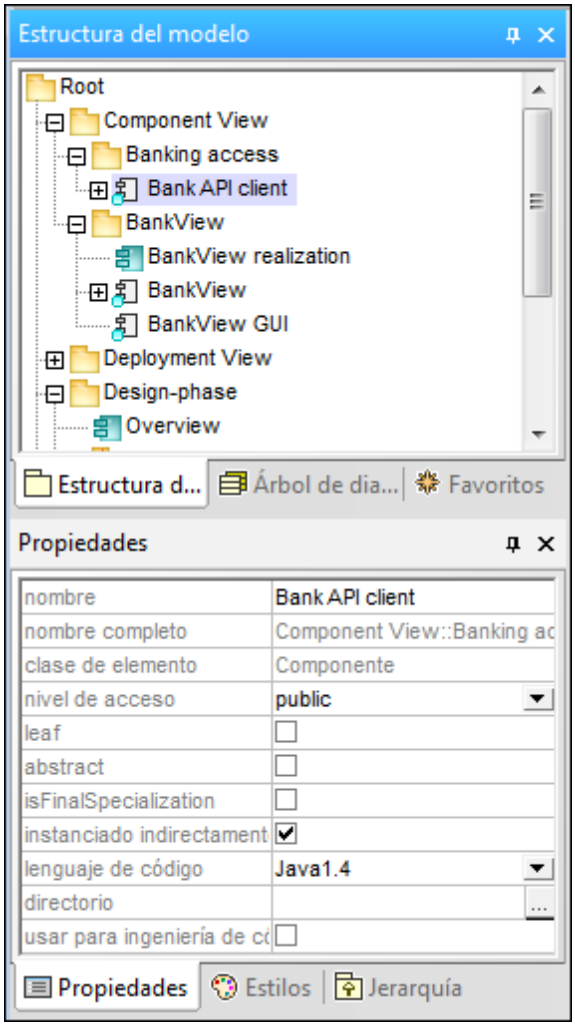


Revisar la sintaxis del proyecto antes de generar el código:

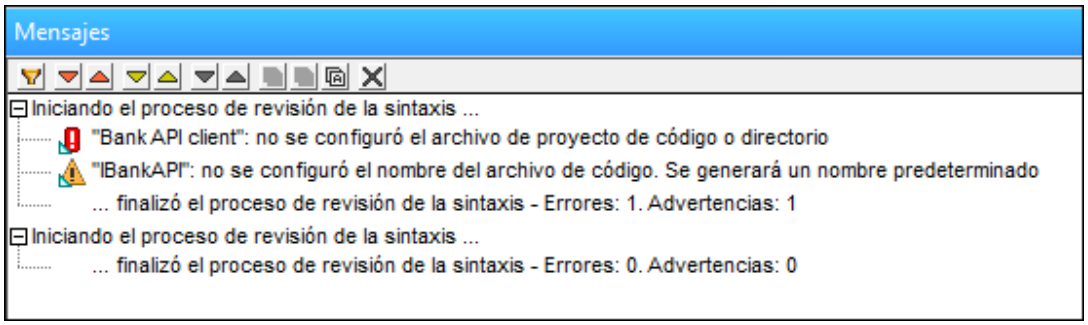
1. Seleccione la opción de menú **Proyecto | Revisar la sintaxis del proyecto**.
2. Cuando termina la revisión de la sintaxis aparece un mensaje en la ventana Mensajes: `"Bank API-client: no se configuró el archivo de proyecto de código o directorio" - "IBankAPI: no se configuró el nombre del archivo de código".`



3. Haga clic en el primer mensaje de la ventana.
4. El paquete **`Bank API client`** aparece resaltado en la vista *Estructura del modelo* y sus propiedades aparecen en la pestaña *Propiedades*.
5. Desactive la casilla `usar para ingeniería de código` de la pestaña *Propiedades*.



6. Revise otra vez la sintaxis del proyecto con el comando **Proyecto | Revisar la sintaxis del proyecto**.

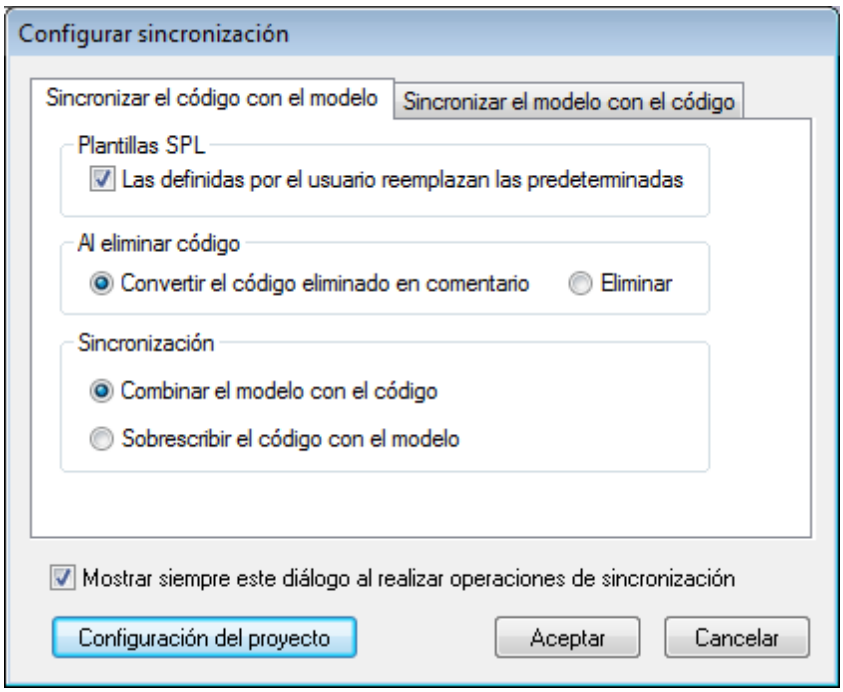


La revisión ya no encuentra más errores. Ahora podemos generar código de programa para el proyecto. Para más información consulte el apartado Revisar la sintaxis del proyecto.

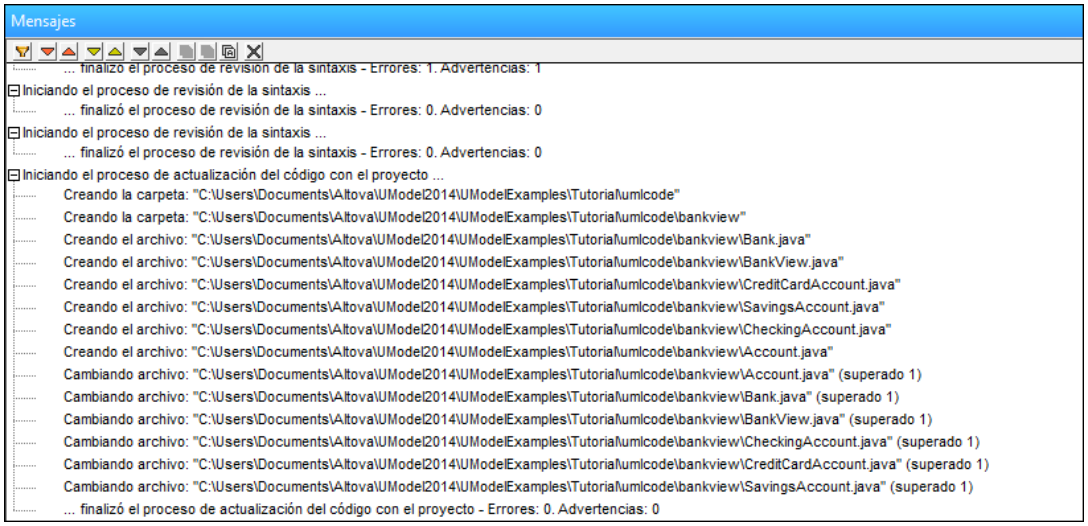
Para generar código de proyecto:

1. Haga clic en el paquete **BankView** para seleccionarlo.

2. Seleccione la opción de menú **Proyecto | Combinar el código de programa con el proyecto de UModel**.
3. Seleccione las opciones de sincronización en el cuadro de diálogo y pulse **Aceptar** para continuar (para este ejemplo del tutorial no hace falta cambiar ninguna opción. Consulte el apartado Combinar el código de programa con el proyecto de UModel para más información).



El panel de mensajes muestra el resultado del proceso de generación de código.

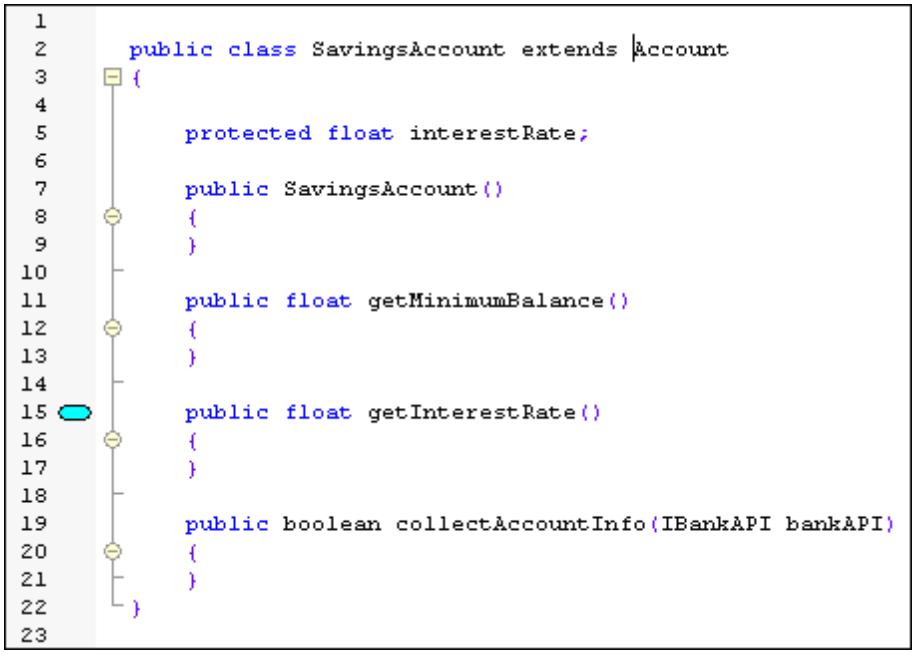


4. Navegue hasta el directorio de destino.
   Se crearon seis archivos **.Java** para el proyecto.

Sincronizar el modelo de UModel después de actualizar el código Java en una

aplicación externa:

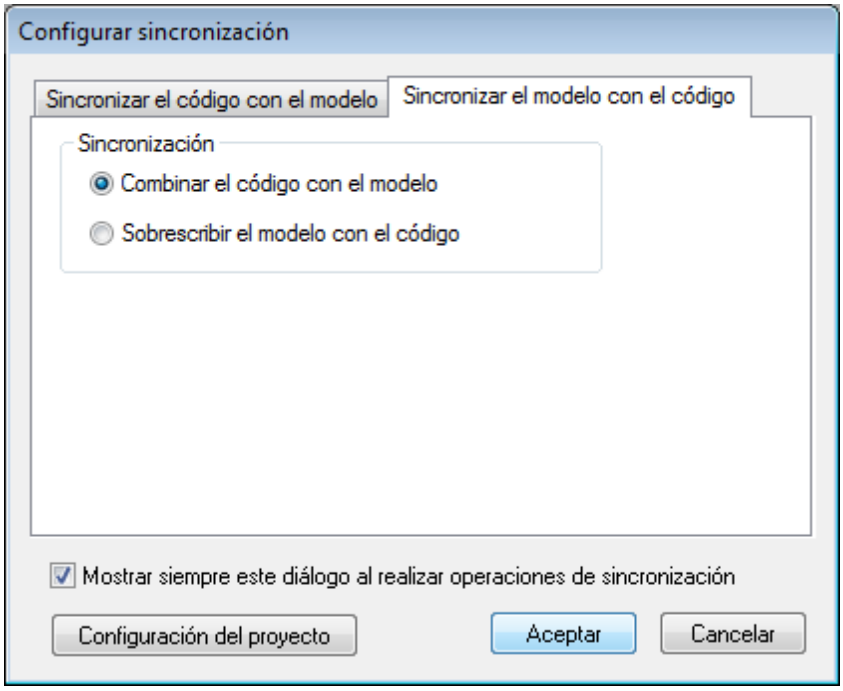
1. Abra el archivo `SavingsAccount.java` en un editor de texto (en XMLSpy por ejemplo).
2. Añada el método **`public float getInterestRate() {}`** al código y guarde el archivo.



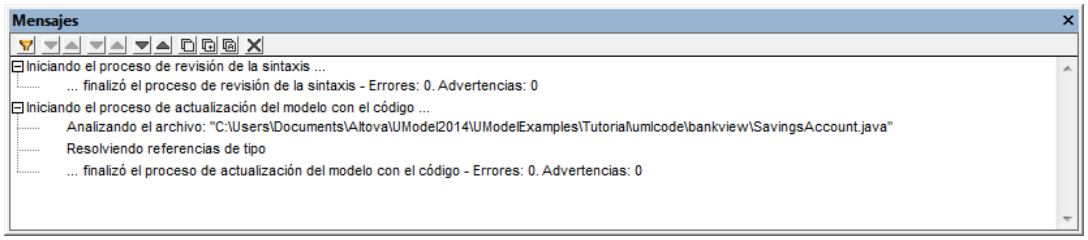
```
1
2     public class SavingsAccount extends Account
3     {
4
5         protected float interestRate;
6
7         public SavingsAccount()
8         {
9         }
10
11        public float getMinimumBalance()
12        {
13        }
14
15        public float getInterestRate()
16        {
17        }
18
19        public boolean collectAccountInfo(IBankAPI bankAPI)
20        {
21        }
22    }
23
```

3. Vuelva a UModel y, en la *Estructura del modelo*, haga clic con el botón derecho en la clase **`SavingsAccount`** situada debajo del paquete **`BankView`**.
4. Seleccione el comando **Ingeniería de código | Combinar la clase de UModel con el código de programa**.

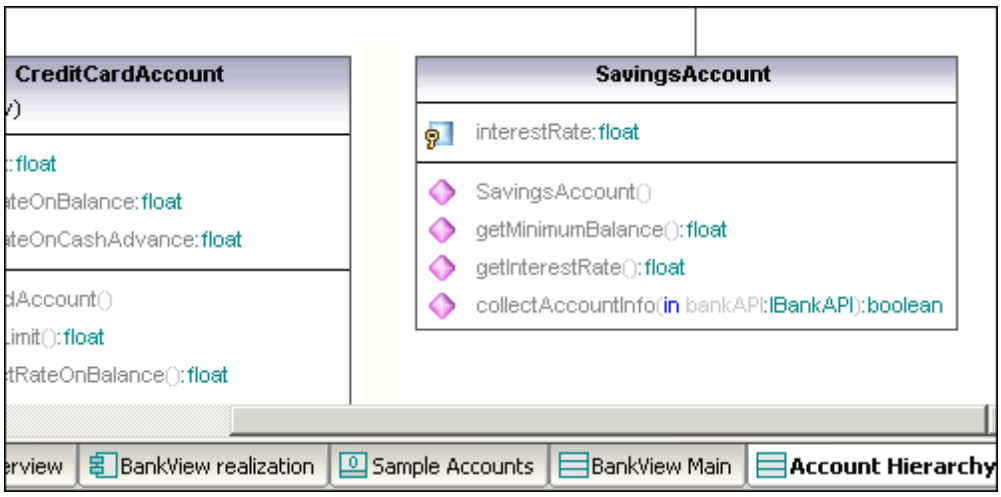
   Esto abre el cuadro de diálogo "Configurar sincronización" por la pestaña *Sincronizar el modelo con el código* (para este ejemplo del tutorial no hace falta cambiar ninguna opción. Consulte el apartado Combinar el proyecto de UModel con el código para más información).



5. Haga clic en **Aceptar** para combinar el modelo con el código.



6. Ahora haga clic en la pestaña del diagrama **Account Hierarchy** para ver el resultado del proceso de combinación.



El método nuevo se añadió al código (`getInterestRate`) y generó una **operación** nueva en la **clase** `SavingsAccount`.

3.8 Ingeniería de ida y vuelta (código - modelo - código)

Sitio web de Altova: Ingeniería inversa de modelos UML a partir de código Java, C# y Visual Basic

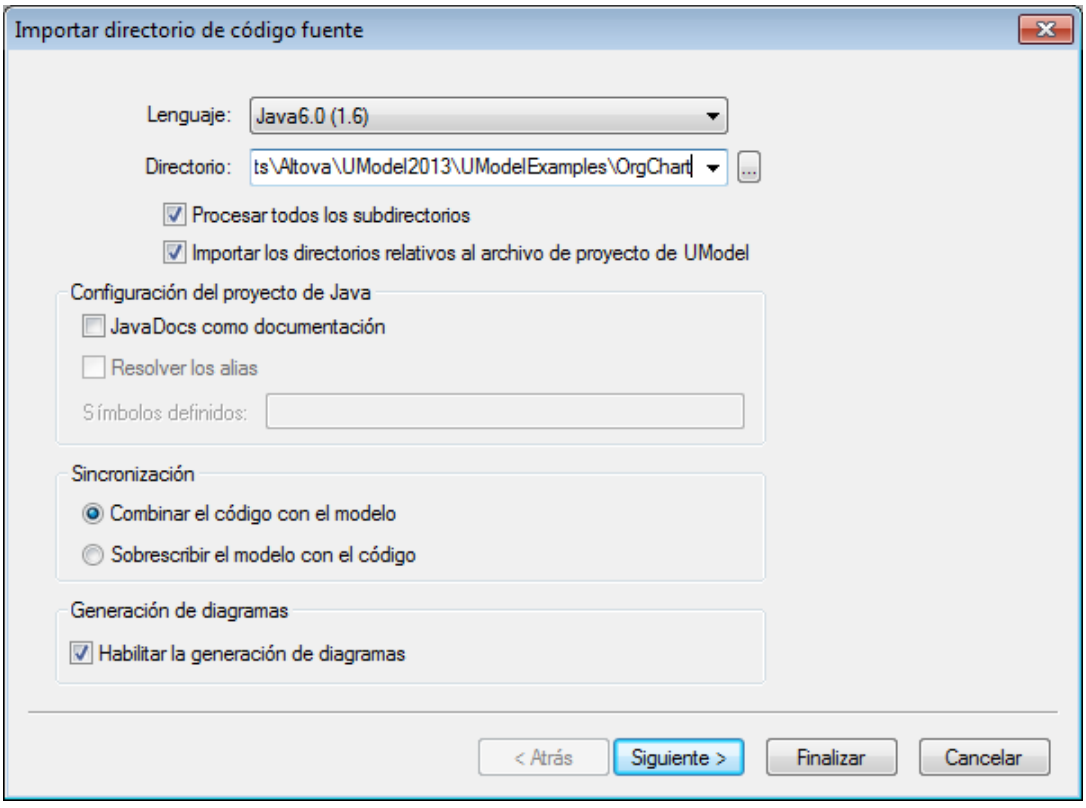
En este apartado del tutorial aprenderá a:

- Importar un directorio que contiene código Java generado con XMLSpy
- Agregar una clase nueva al proyecto en UModel
- Combinar un paquete de UModel con el código de programa

Los archivos utilizados para este ejemplo están disponibles en el archivo comprimido `OrgChart.zip` de la carpeta `...\UModelExamples`. Descomprima el archivo antes de empezar con este apartado del tutorial. Esto crea el directorio `OrgChart`, que después se usará para importar el código ya existente.

Importar código ya existente desde un directorio

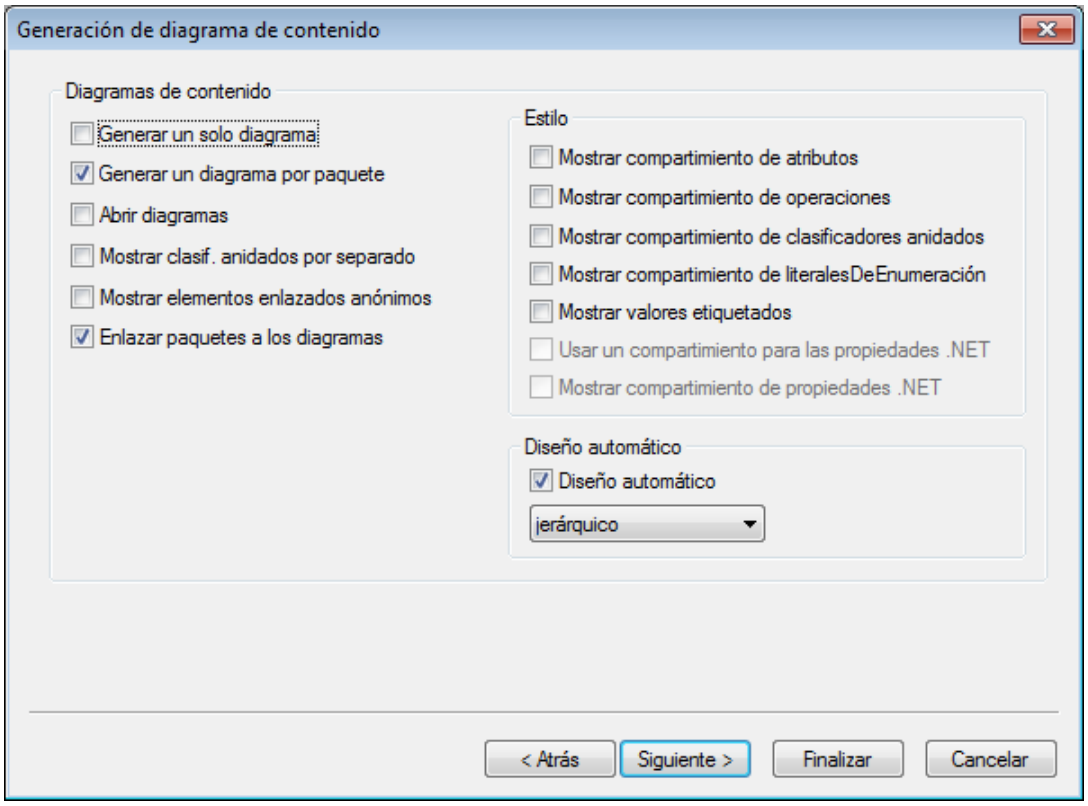
1. Haga clic en **Archivo | Nuevo** para crear un proyecto nuevo.
2. En el menú **Proyecto** seleccione el comando **Importar directorio de código fuente**.
3. Seleccione la versión de C#, Java o VB del código fuente.
4. Ahora haga clic en el botón **Examinar** ... y seleccione el directorio `OrgChart` de la carpeta `...\UModelExamples`.



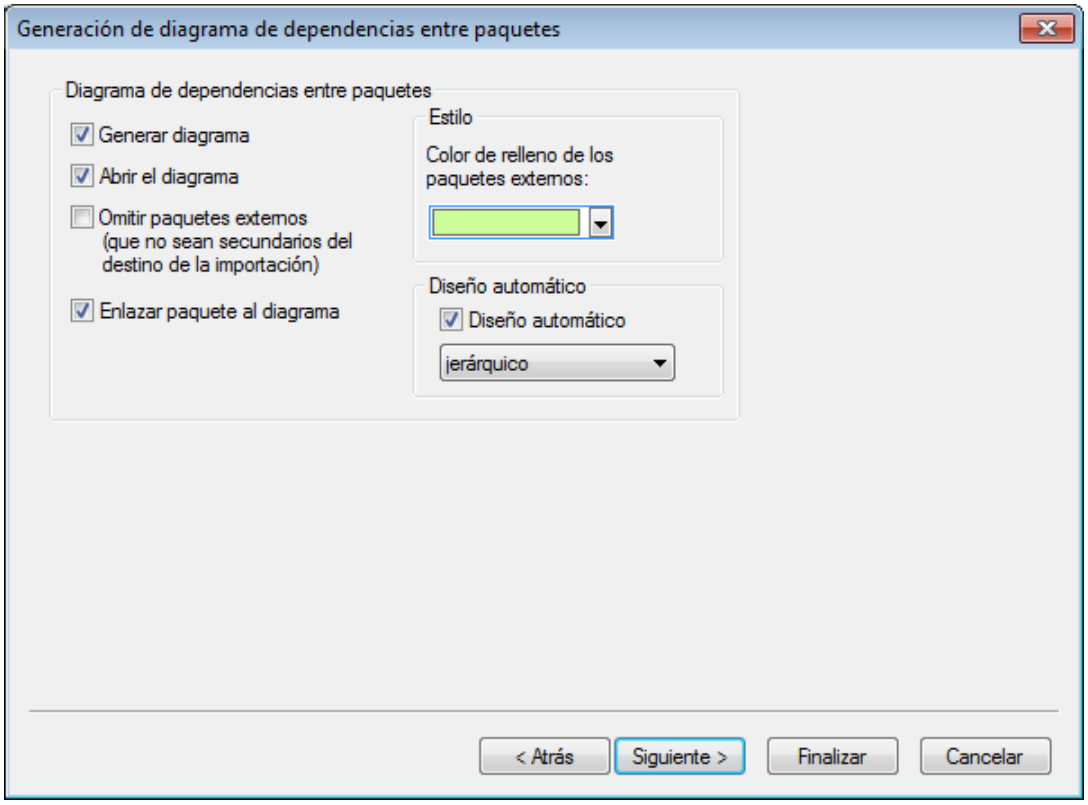
5. Para que UModel pueda generar diagramas de clases y diagramas de paquetes a partir

del código fuente, marque la casilla *Habilitar la generación de diagramas* y haga clic en **Siguiente**.

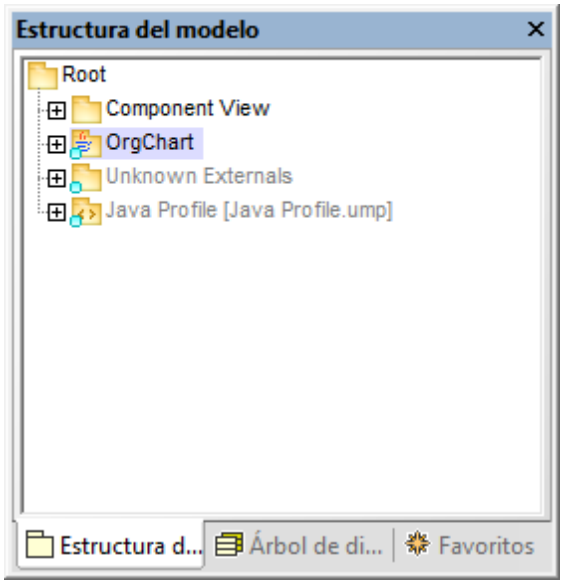
6. Seleccione la opción *Generar un solo diagrama* o *Generar un diagrama por paquete*, dependiendo de lo que necesite. Las opciones de estilos de los diagramas se pueden cambiar más adelante si es necesario.



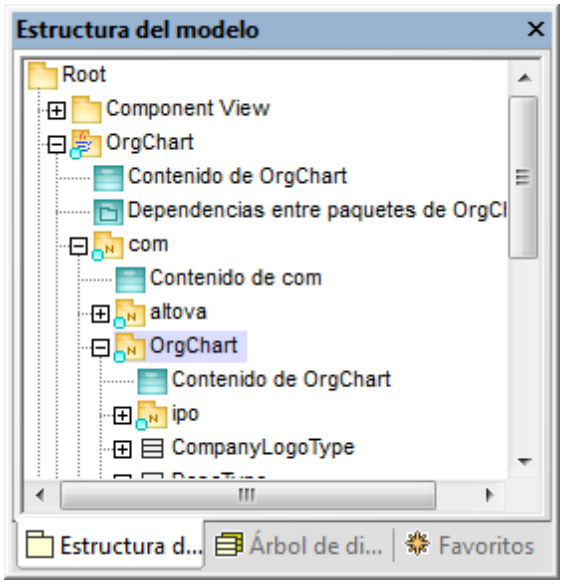
7. Haga clic en **Siguiente** para continuar. Ahora aparece un cuadro de diálogo donde puede configurar la generación de diagramas de dependencias entre paquetes.



8. Para terminar haga clic en **Finalizar**. Después deberá guardar el nuevo modelo en un directorio del sistema. Los datos se analizan y se crea un paquete nuevo llamado **`OrgChart`**.



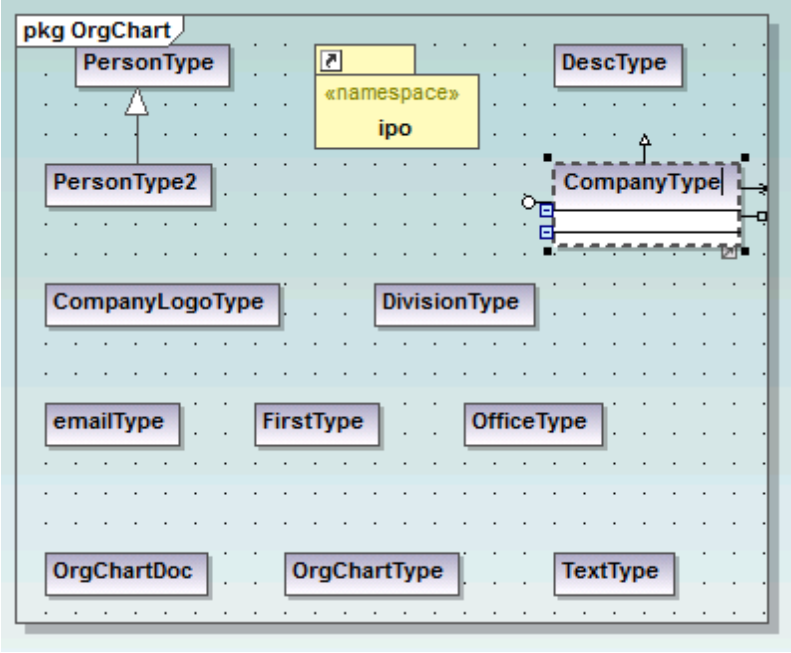
9. Expanda el paquete nuevo y siga expandiendo los subpaquete hasta llegar al paquete **`OrgChart`** (**`com | OrgChart`**). Haga doble clic en **`Contenido de OrgChart`**.



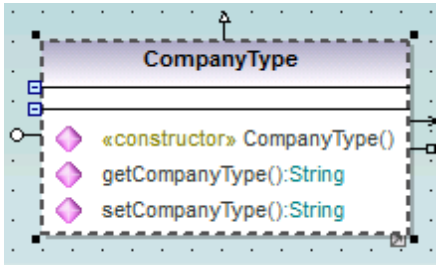
En la pestaña principal aparecen las clases contraídas que componen `OrgChart`.

Agregar una clase nueva al diagrama OrgChart

1. Haga clic con el botón derecho dentro del panel principal y seleccione **Nuevo/a | Clase** en el menú contextual.
2. Haga clic en el título de la clase nueva y escriba el nombre `CompanyType`.

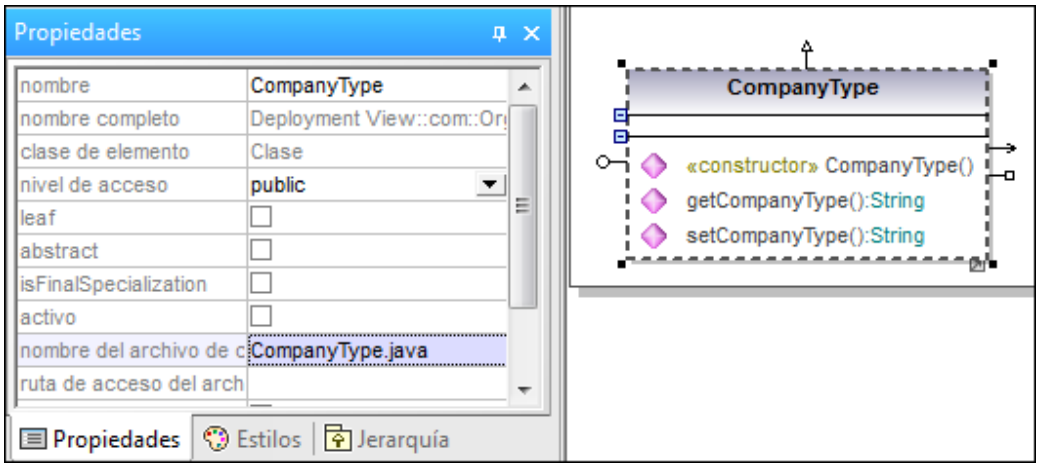


3. Añada operaciones nuevas a la clase con la tecla **F8**. En el caso que nos ocupa, por ejemplo, puede añadir estas operaciones: `CompanyType()`, `getCompanyType():String`, `setCompanyType():String`.

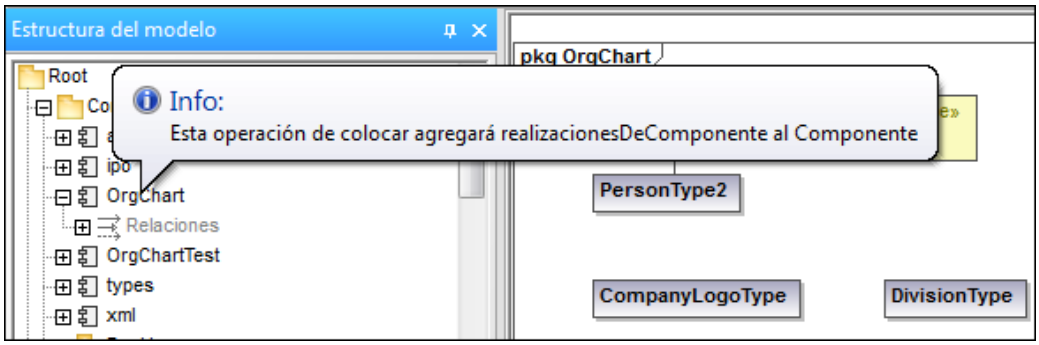


Habilitar la clase nueva para la generación de código

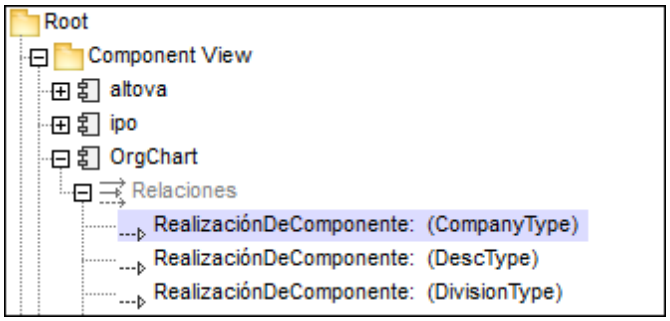
1. Compruebe que la clase activa es **`CompanyType`**, haga clic en el campo `nombre del archivo de código` y escriba el nombre del archivo Java de la nueva clase `CompanyType.java`.



2. En la *Estructura del modelo* haga clic en la nueva clase **`CompanyType`**, arrástrela hacia arriba y suéltela encima del componente **`OrgChart`** (situado justo debajo del paquete **`Component View`**). Al pasar el puntero del ratón por el componente aparece información rápida.

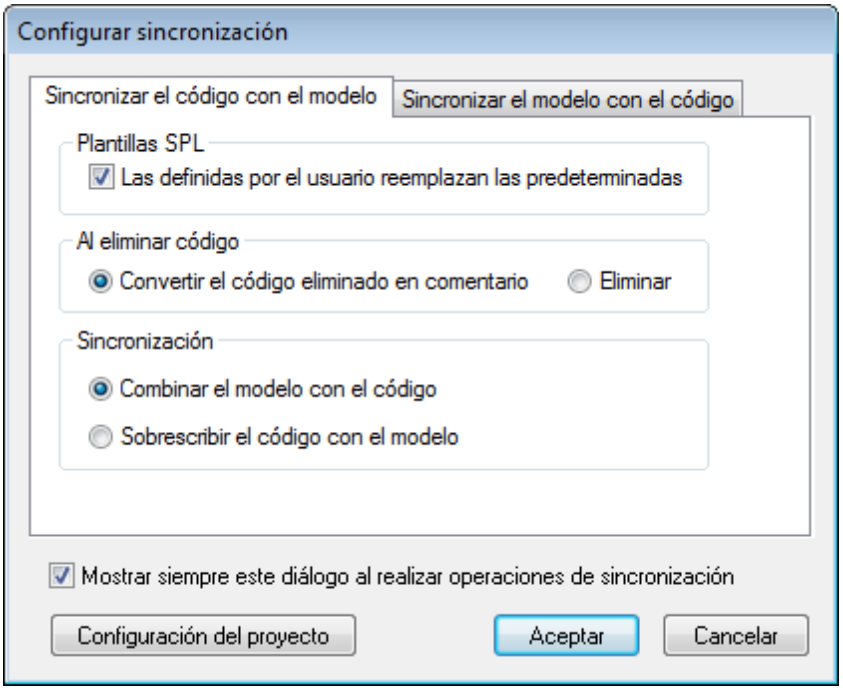


Este método crea una Realización entre una clase y un componente, sin necesidad de usar diagramas de componentes ni de implementación. Expanda el nodo `Relaciones` situado debajo del componente **`OrgChart`** para ver la nueva Realización que acabamos de crear.

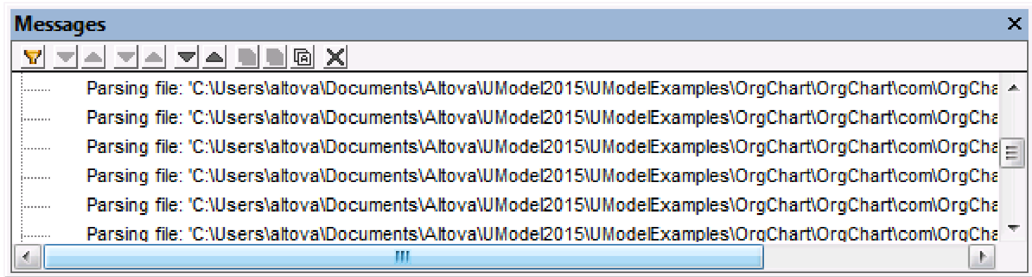


Combinar código de programa desde un paquete

1. Haga clic con el botón derecho en el paquete `OrgChart`, seleccione el comando **Ingeniería de código | Combinar el código de programa con el paquete de UModel** y pulse **Entrar** para confirmar.



La ventaja de mensajes muestra el resultado de la revisión sintáctica y el estado del proceso de sincronización.



Una vez finalizado el proceso, la nueva clase **`CompanyType.java`** se añade a la carpeta `...\OrgChart\com\OrgChart\`.

> El cuerpo de los métodos y los cambios realizados en el código se convertirán en comentarios o se eliminarán, dependiendo de la opción elegida en el grupo *Al eliminar código* del cuadro de diálogo "Configurar sincronización".

Ahora ya sabe crear un proyecto de modelado usando el proceso de ingeniería directa y completar un ciclo de ingeniería de ida y vuelta con UModel. El resto de la documentación describe cómo obtener los mejores resultados para su proyecto de modelado con UModel.

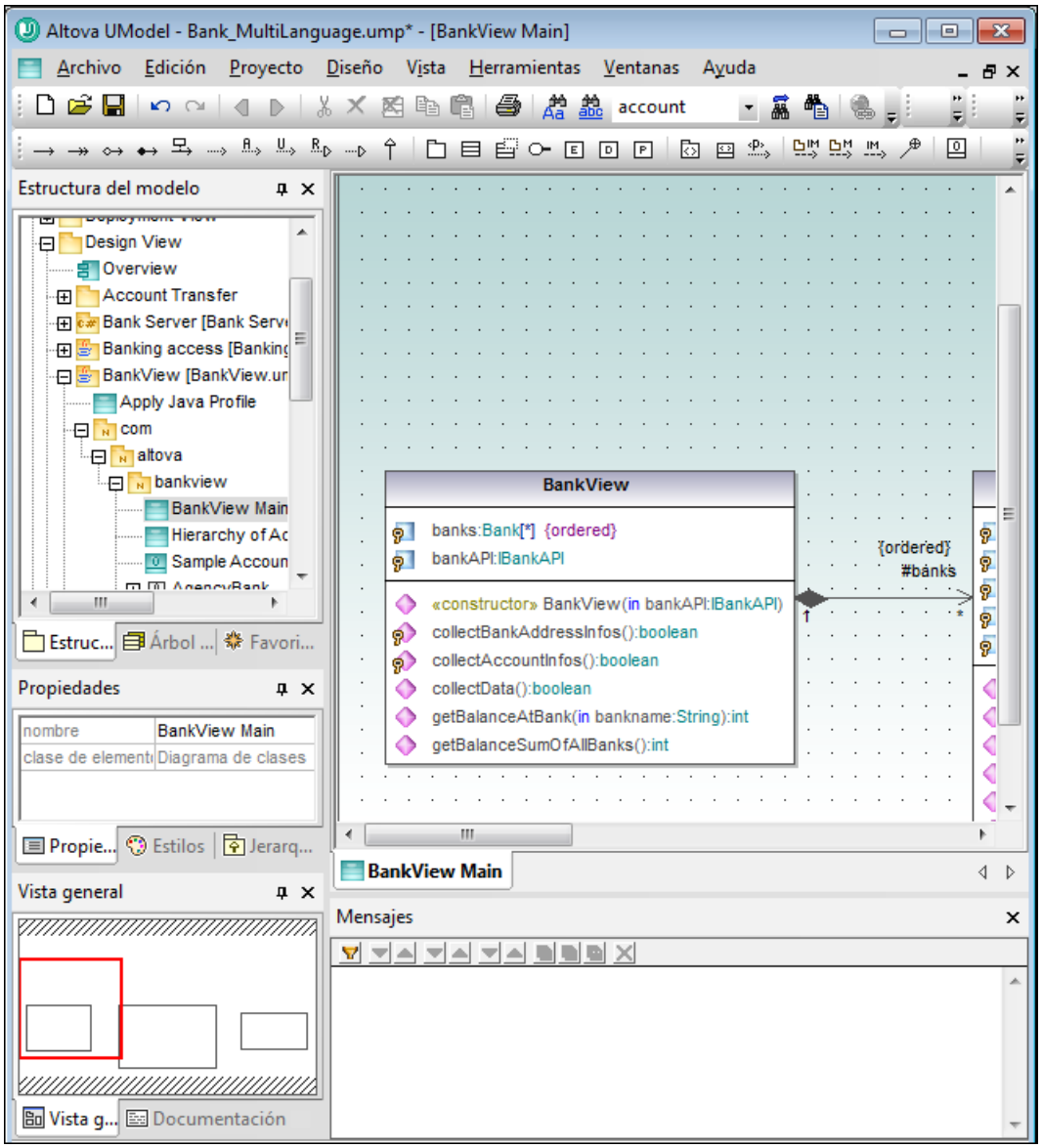
Chapter 4

Interfaz del usuario

4 Interfaz del usuario

La interfaz del usuario de UModel se divide en dos secciones: a la izquierda de la pantalla se encuentra una serie de paneles y a la derecha está la zona de trabajo donde se abren las pestañas de los diagramas. Los paneles de la izquierda permiten visualizar el proyecto de UModel desde diferentes puntos de vista y editar datos. Se trata de los paneles *Estructura del modelo*, *Propiedades* y *Vista general*. El área de trabajo situada a la derecha muestra la pestaña del diagrama activo (p. ej. en la imagen siguiente aparece el diagrama de clases del paquete **`BankView Main`**).

Nota: en todos los paneles y en todas las pestañas de diagramas puede buscar elementos y componentes desde el cuadro combinado *Buscar* de la barra de herramientas principal (p. ej. en la imagen siguiente se buscó el texto `account`) o pulsando la tecla **Ctrl+F**.

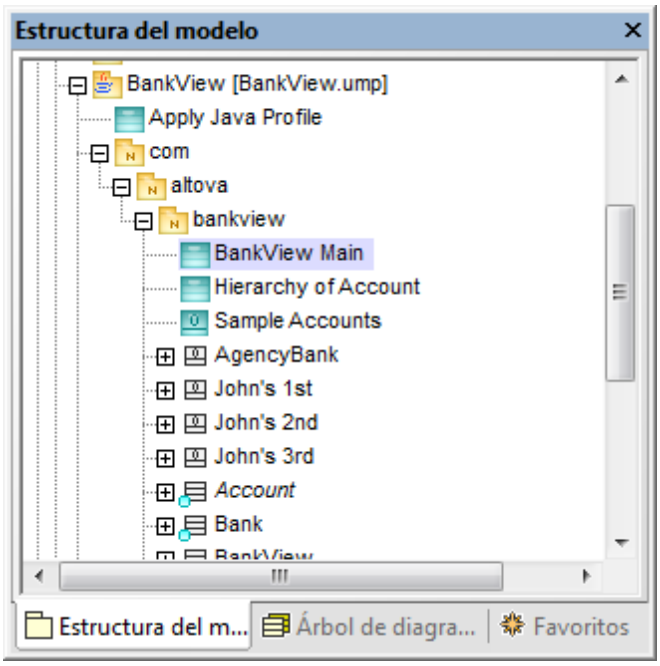


4.1 Estructura del modelo

La pestaña *Estructura del modelo* sirve para manipular componentes de los modelos y para navegar por determinados componentes del área de trabajo. Al hacer clic con el botón derecho en un elemento aparece un menú contextual con varios comandos. Dependiendo del tipo de elemento seleccionado, el menú contextual incluye unos comandos u otros.

Estas son las operaciones que se pueden realizar en los elementos de modelado desde la pestaña *Estructura del modelo* directamente:

- Agregar / insertar
- Copiar / mover
- Eliminar
- Cambiar de nombre
- Ordenar según diferentes criterios
- Restringir



En la pestaña *Estructura del modelo* cada icono en forma de carpeta equivale a un paquete UML.

Para agregar un paquete nuevo (o cualquier otro elemento de modelado):

1. Haga clic con el botón derecho en la **carpeta** bajo la que debe aparecer el nuevo paquete o elemento.
2. Seleccione **Nuevo/a | Paquete** (o el elemento de modelado que corresponda).

Para copiar o mover elementos de modelado:

Tiene dos opciones

1. Usar los comandos estándar de Windows para copiar, cortar o pegar o
2. arrastrar elementos de modelado a otro paquete. Si pulsa la tecla **Ctrl** mientras arrastra un elemento, se crea una copia del elemento.

En algunas ocasiones, al arrastrar el elemento aparece un mensaje diciendo que se debe activar la opción **No ordenar** para poder completar la acción. Consulte el apartado Cortar, copiar y pegar en diagramas de UModel para más información.

Para ordenar elementos en la *Estructura del modelo* (activando la opción No ordenar):

1. Haga clic con el botón derecho en el fondo de la pestaña *Estructura del modelo*.
2. Seleccione **Ordenar | No ordenar**.
   Ahora los elementos se pueden colocar en cualquier posición de la *Estructura del modelo*.

Nota: en el submenú **Ordenar** también hay opciones para ordenar las propiedades y las operaciones de los diagramas.

Para cambiar el nombre de un elemento:

1. Haga doble clic en el **nombre** del elemento y edítelo.
   Los paquetes `Root` y `Component View` son los únicos dos elementos cuyo nombre no se puede editar.

Para eliminar un elemento:

1. Haga **clic** en el elemento que desea eliminar (o **Ctrl+clic** para seleccionar varios elementos a la vez).
2. Pulse la tecla **Supr**.

   El elemento de modelado se elimina de la Estructura del modelo. Esto significa que también se elimina de la pestaña del diagrama y, si está en el proyecto, también del proyecto. Si quiere puede eliminar los elementos del diagrama pero no del proyecto. Para ello use **Ctrl+Supr**. Para más información consulte el apartado Eliminar elementos.

Para abrir un diagrama en el área de trabajo principal:

1. Haga doble clic en el icono del diagrama que desea abrir en el área de trabajo.

Iconos de los elementos de modelado en la *Estructura del modelo*

Tipos de paquetes:

Paquete UML
Paquete raíz del espacio de nombres Java
Paquete raíz del espacio de nombres C#
Paquete raíz del espacio de nombres Visual Basic
Paquete raíz de XML Schema
Paquete de código Java, C# y VB (las declaraciones de paquete se crean cuando se genera el código)

Tipos de diagramas:

Diagrama de actividades
Diagrama de clases
Diagrama de comunicación
Diagrama de paquetes
Diagrama de perfil
Diagrama de secuencia

Diagrama de componentes
Diagrama de estructura de un compuesto
Diagrama de implementación
Diagrama global de interacción
Diagrama de objetos
Diagrama de máquina de estados
Diagrama de ciclo de vida
Diagrama de caso de uso
Diagrama de esquema XML

Tipos de elementos:

Los elementos que están visibles en el diagrama activo aparecen con un punto azul en la parte inferior. En este caso se trata de un elemento de tipo clase.

Instancia de clase / objeto
Slot de instancia

Clase
Propiedad
Operación
Parámetro

Actor (visible en el diagrama de casos de uso activo)
Caso de uso
Componente
Nodo
Artefacto
Interfaz

Relaciones (o paquete)
Restricciones

Para abrir y expandir paquetes en la *Estructura del modelo*:
Hay dos maneras de abrir paquetes en la *Estructura del modelo*. Una consiste en abrir todos los paquetes y paquetes subordinados y la otra consiste en abrir el paquete seleccionado solamente.

Haga clic en el paquete que desea abrir y elija una de estas dos opciones:

- Pulse la tecla ***** para abrir el paquete seleccionado y todos sus paquetes subordinados.
- Pulse la tecla **+** para abrir el paquete seleccionado solamente.

  Para contraer los paquetes pulse la tecla **-** del teclado. Para contraerlos haga clic en el paquete `Root` y pulse **-**.
  Recuerde que puede usar las teclas estándar o las del teclado numérico.

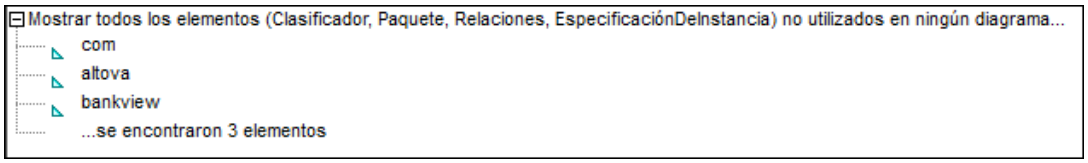
Para buscar elementos de modelado en las pestañas de diagramas:
Mientras navega por los elementos de la *Estructura de modelo* puede ver si el elemento está en el diagrama y, si lo está, en qué posición. Hay dos maneras de buscar elementos:

1. En la *Estructura del modelo* haga clic con el botón derecho en el elemento que desea

buscar y después seleccione:
- **Mostrar el elemento en el diagrama activo**
- **Mostrar el elemento en todos los diagramas**

Para generar una lista de elementos no utilizados en ningún diagrama:

1. Haga clic con el botón derecho en el paquete que desea inspeccionar.
2. Seleccione el comando **Mostrar elementos no utilizados en ningún diagrama**.
   En el panel Mensajes aparece una lista de elementos no utilizados. Los elementos que aparecen entre paréntesis son los tipos de elemento no utilizados. Consulte el apartado sobre la pestaña *Vista* del cuadro de diálogo "Opciones" de la *Referencia del usuario*.



   Para buscar los elementos que faltan en la Estructura del modelo, haga clic en el nombre del elemento en el resultado del panel Mensajes.

Nota: los elementos no utilizados son los del paquete actual y de sus paquetes subordinados.

Paquetes en la *Estructura del modelo*:

Al iniciar UModel por primera vez solamente se pueden ver los paquetes `Root` y `Component View` (porque no hay cargado ningún proyecto).

En la *Estructura del modelo* puede:

- **Crear** paquetes o eliminarlos (los paquetes son contenedores que almacenan los demás elementos de modelado UML, los diagramas de casos de uso, etc.).
- **Mover**/copiar paquetes y su contenido a otros paquetes (y a diagramas válidos del área de trabajo).
- **Ordenar** paquetes y su contenido según determinados criterios.
- **Colocar** los paquetes dentro de otros paquetes.
- **Utilizar** los paquetes como elementos de origen y destino cuando genere o sincronice código.

Para generar y combinar código:

En UModel puede generar o combinar código de programa desde la *Estructura del modelo* directamente. Consulte el apartado Sincronizar el modelo y el código fuente para obtener más información.

Para restringir elementos UML desde la *Estructura del modelo*:

Puede definir restricciones para la mayoría de los elementos de modelado. Tenga en cuenta que la revisión de sintaxis pasa por alto las restricciones porque no participan en el proceso de generación de código Java.

1. En la *Estructura del modelo* haga clic con el botón derecho en el elemento que desea

restringir y seleccione **Nuevo/a | Restricción**.
2. Escriba el nombre de la restricción y pulse **Entrar**.
3. Haga clic en el campo `especificación` de la pestaña *Propiedades* y escriba la restricción (p. ej. longitud del nombre > 10).

Para restringir elementos UML desde el área de trabajo del diagrama:

1. Haga doble clic en el elemento para poder editarlo.
2. Añada la restricción entre llaves (p. ej. `interestRate:float #{interestRate >=0}`).

Para asignar restricciones a varios elementos de modelado:

1. Haga clic con el botón derecho en el campo `elementos restringidos` de la pestaña *Propiedades*.
2. Seleccione el comando **Agregar elemento a elementos restringidos** en el menú contextual.
   Esto abre el cuadro de diálogo "Seleccione el elemento que se debe restringir".
3. Seleccione el elemento al que desea asignar la restricción actual.

   El campo elementos restringidos ahora contiene los nombres de los elementos de modelado a los que se asignó la restricción.

4.2 Árbol de diagramas

La pestaña *Árbol de diagramas* muestra todos los diagramas disponibles actualmente en UModel de dos formas diferentes:

- Agrupados por tipo de diagrama (y en orden alfabético)
- En una lista alfabética de todos los diagramas del proyecto

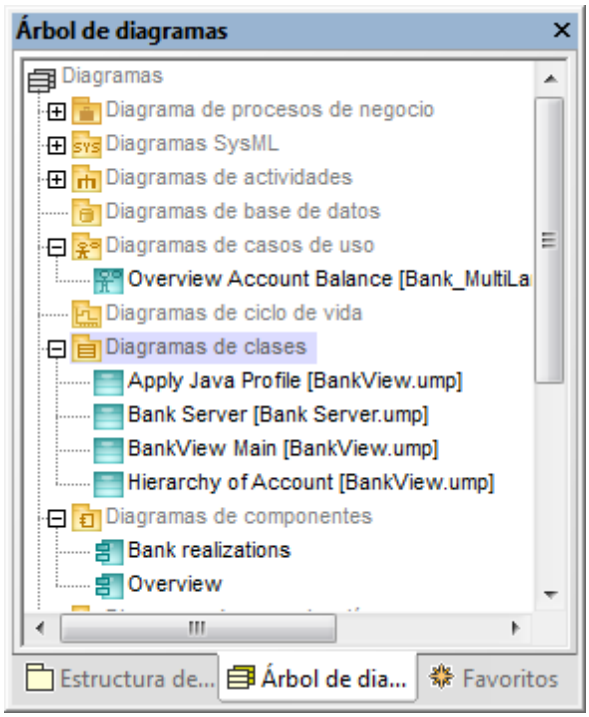
Nota: en la pestaña *Árbol de diagramas* puede añadir o eliminar diagramas haciendo clic con el botón derecho y seleccionando el comando correspondiente.

Para abrir un diagrama en la pestaña Diagramas:

- Haga doble clic en el diagrama que desea ver en el área de trabajo.

Para ver los diagramas agrupados según el tipo de diagrama:

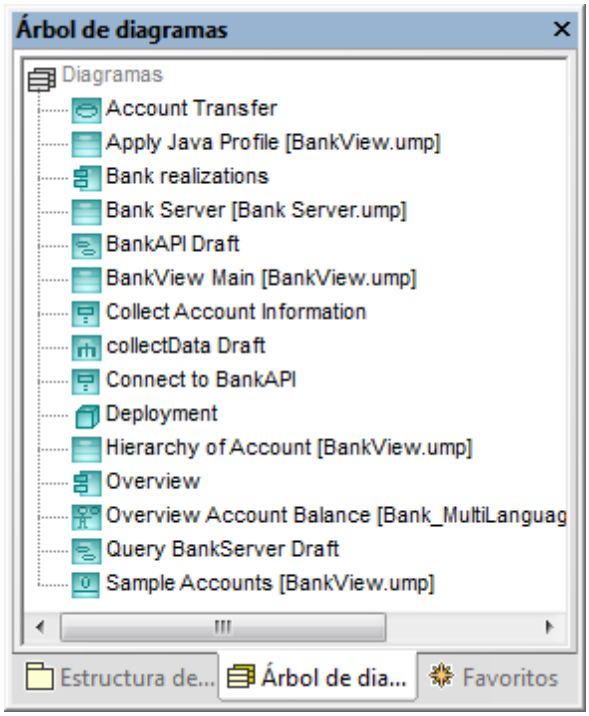
- Haga clic con el botón derecho en el fondo de la pestaña y active la opción **Agrupar según el tipo de diagrama**.



Los diagramas aparecen organizados por grupos.

Para ver todos los diagramas en una lista alfabética:

- Haga clic con el botón derecho y desactive la opción **Agrupar según el tipo de diagrama**.



Todos los diagramas del proyecto aparecen ordenados alfabéticamente.

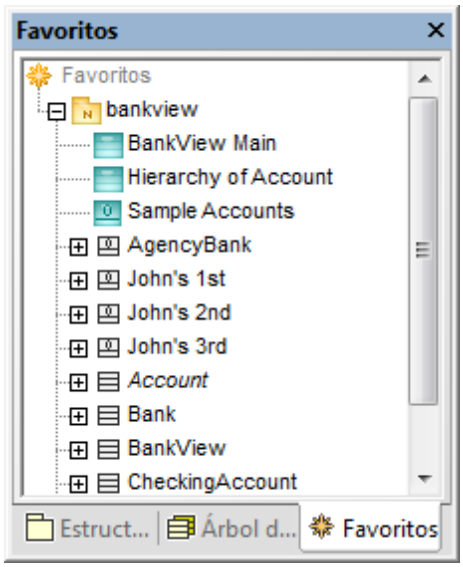
4.3 Favoritos

La pestaña *Favoritos* sirve para crear un repositorio (o biblioteca) personal donde puede almacenar elementos UML **con nombre** (es decir, clases, objetos, asociaciones, etc.) pero no dependencias de tipo `AplicaciónDePerfil` ni `Generalización`. Por tanto, en esta pestaña puede crear una lista de elementos de modelado favoritos y así poder acceder a ellos rápidamente.

El contenido de la pestaña *Favoritos* se guarda automáticamente en el archivo de proyecto. Esto se puede cambiar en la pestaña *Archivo* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**), activando/desactivando la casilla *Cargar y guardar con el archivo de proyecto*.

Para agregar un elemento de modelado a la pestaña *Favoritos*:

1. En la *Estructura del modelo* o en el área de trabajo haga clic con el botón derecho en el elemento.
2. Seleccione el comando **Agregar a favoritos** del menú contextual.
3. Haga clic en la pestaña *Favoritos* y compruebe que el elemento se añadió efectivamente al repositorio.



El elemento que aparece en la pestaña *Favoritos* es una vista del elemento (es decir, no es ni una copia ni un clon del elemento).

Para agregar un elemento nuevo a la pestaña *Favoritos*:

1. Haga clic con el botón derecho en el paquete al que desee añadir un elemento.
2. Seleccione la opción **Elemento nuevo | <elemento de modelado>** (una clase, un componente, etc.).
   Al mismo paquete/elemento del proyecto se añade el elemento nuevo, que por tanto se podrá ver también en la *Estructura del modelo*.

Para quitar un elemento de la pestaña *Favoritos*:

1. Haga clic con el botón derecho en el elemento/paquete en la pestaña *Favoritos*.

2. Seleccione **Quitar de favoritos**.

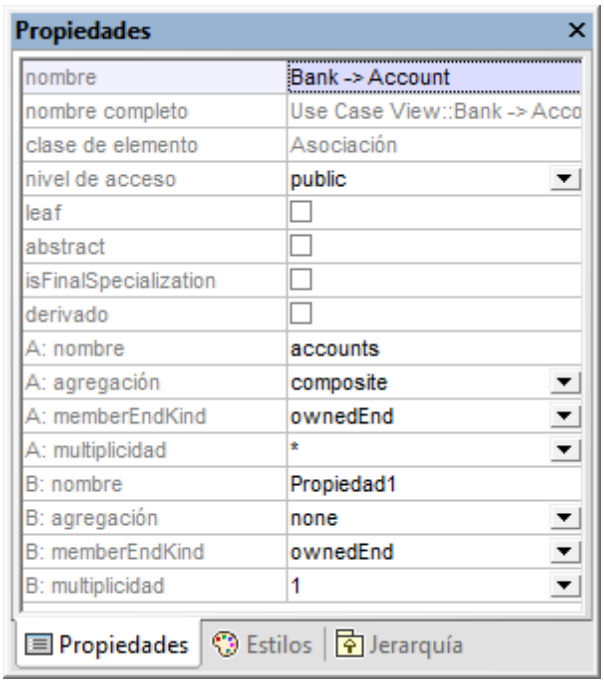
Nota: los elementos se pueden quitar o agregar a favoritos desde la pestaña *Favoritos* y desde la pestaña *Estructura del modelo*.

Para eliminar elementos de la pestaña *Favoritos*:

1. Haga clic con el botón derecho en el elemento que desea eliminar y pulse la tecla **Supr**. Aparece un mensaje informando de que el elemento se eliminará del proyecto.
2. Haga clic en **Aceptar** si de verdad quiere eliminarlo del proyecto.
3. Haga clic en **Cancelar** para conservar el elemento. Puede usar el comando **Quitar de favoritos** para eliminarlo de la pestaña *Favoritos* solamente.

4.4 Propiedades

La pestaña *Propiedades* muestra las propiedades UML del elemento de modelado activo.

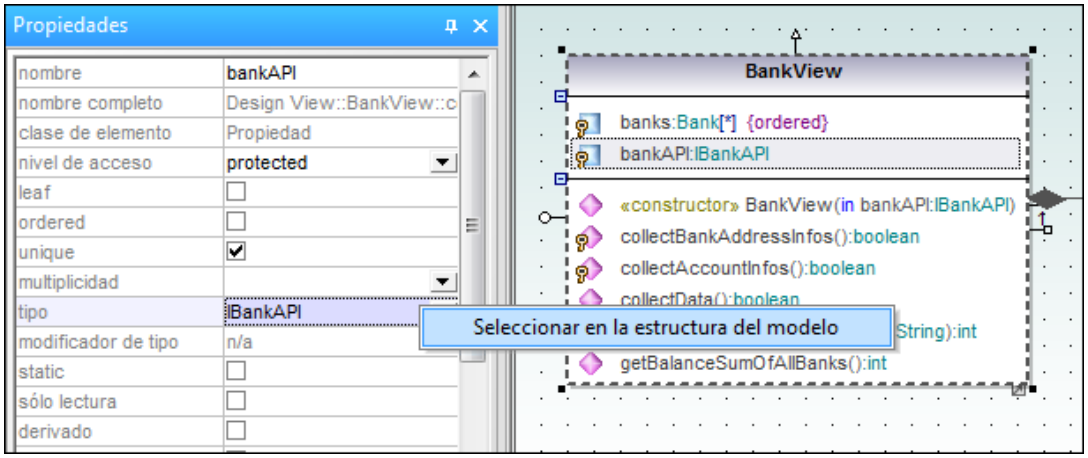


- Al hacer clic en un elemento de modelado en cualquiera de las vistas y pestañas, la pestaña *Propiedades* muestra las propiedades del elemento.
- Las propiedades que aparecen en la pestaña *Propiedades* se pueden editar, rellenar y modificar seleccionando diferentes opciones disponibles en menús contextuales y desplegables.
- Las propiedades seleccionadas en la pestaña *Propiedades* se pueden ver en la pestaña del diagrama haciendo clic en la opción **Mostrar el elemento en el diagrama activo** del menú contextual.

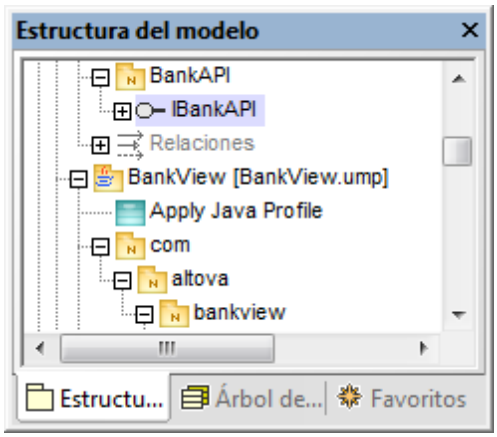
Selección en la *Estructura del modelo*

Si hace clic en un atributo de un diagrama de clases, sus propiedades aparecen en la pestaña *Propiedades*. Para buscar el atributo en la *Estructura del modelo*:

1. Haga clic con el botón derecho en la entrada `tipo` en la pestaña *Propiedades*.
2. En el menú contextual elija **Seleccionar en la estructura del modelo**.

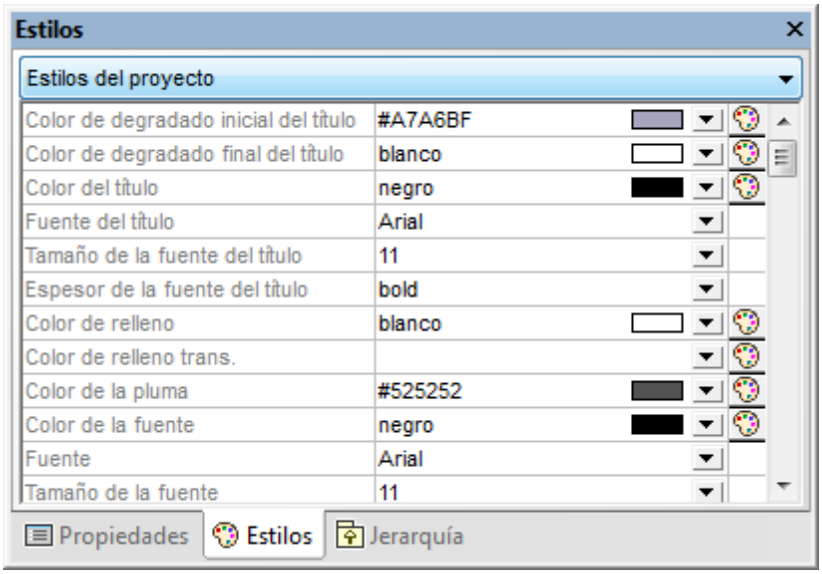


Ahora puede ver la interfaz **`IBankAPI`** en la *Estructura del modelo*.



4.5 Estilos

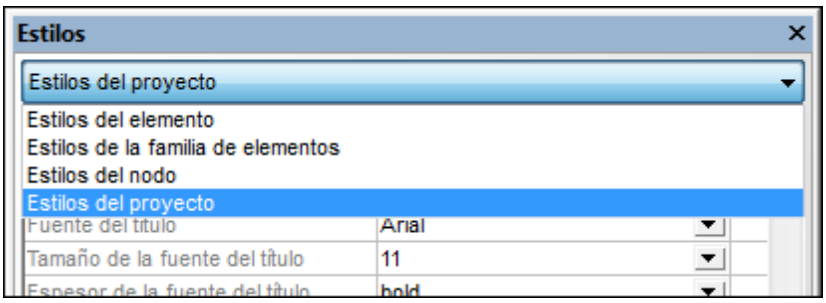
La pestaña *Estilos* sirve para ver y modificar los atributos de los diagramas o elementos que aparecen en el área de trabajo.



Estos atributos de estilo se pueden dividir en dos grupos:

- Atributos de formato (p. ej. tamaño, espesor, color de la fuente, etc.)
- Atributos de presentación (p. ej. ver color de fondo, ver cuadrícula, opciones de visibilidad, etc.)

La pestaña *Estilos* está dividida a su vez en varias secciones. Para cambiar de sección haga clic en el cuadro combinado situado en la parte superior y elija una opción del menú desplegable. Dependiendo del elemento seleccionado, el menú desplegable ofrece unas opciones u otras.



Si selecciona un elemento en la pestaña del diagrama, se selecciona automáticamente el contexto **`Estilos del elemento`**. Si hace clic en un elemento en la pestaña *Estructura del modelo*, se selecciona el contexto **`Estilos del proyecto`**.

El orden de prioridad de los estilos es ascendente, es decir, los cambios realizados en un nivel más concreto, reemplazan los estilos más generales. Por ejemplo, los cambios realizados en un objeto en el contexto **`Estilos del elemento`** reemplazan la configuración actual en el contexto **`Estilos de la familia de elementos`** y **`Estilos del proyecto`**. Sin embargo, si selecciona

otro objeto y cambia alguna opción en **Estilos de la familia de elementos**, se actualizan los demás objetos excepto el que acabamos de cambiar en el contexto **Estilos del elemento**.

Nota: los cambios realizados en los estilos de los elementos de modelado no se pueden deshacer.

Estilos del elemento:
Este contexto afecta el elemento seleccionado en el diagrama activo. Puede seleccionar varios elementos a la vez.

Estilos de los elementos con este estereotipo:
Se aplican a la clase de estereotipo seleccionada en el diagrama (consulte Estilos de estereotipo definidos por el usuario para más información).

Estilos de la familia de elementos:
Abarca todos los elementos que sean del mismo tipo que el elemento seleccionado. P. ej. si quiere que todos los elementos de tipo `Componente` estén en color azul.

Estilos del nodo / de la línea:
"Nodo" abarca todos los objetos rectangulares. "Líneas" se refiere a todos los conectores: asociaciones, dependencias, líneas de realización, etc. de todo el proyecto.

Estilos del proyecto:
Se aplican a todo el proyecto de UModel activo (p. ej. si quiere cambiar la fuente Arial predeterminada por Times New Roman en todo el texto de todos los diagramas del proyecto).

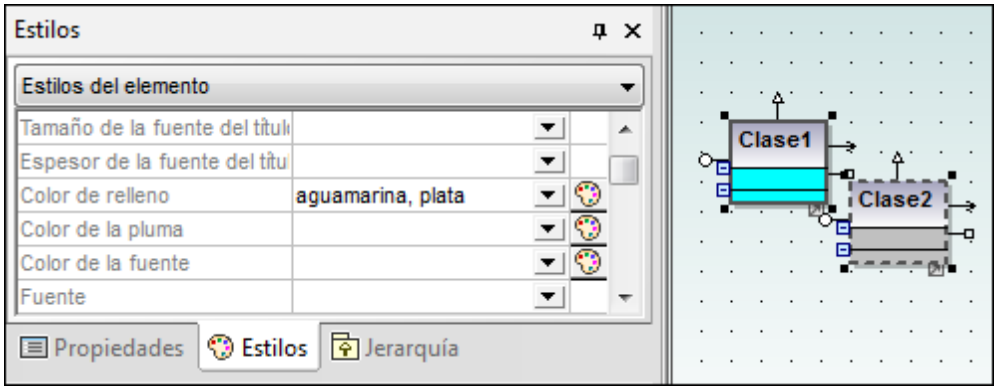
Estilos del diagrama:
Estos estilos están disponibles si hace clic/selecciona el fondo de un diagrama. Estos estilos solo se aplican al diagrama UML para el que se definieron opciones en el proyecto.

Para cambiar la configuración de todos los diagramas del proyecto:

1. Haga clic en el diagrama correspondiente.
2. En el cuadro combinado de la pestaña *Estilos* seleccione la entrada **Estilos del proyecto** y desplácese hasta el final de la lista de la pestaña.
3. Seleccione uno de los estilos `Diagrama - xxxx` (p. ej. `Diagrama - Color de fondo`). A continuación el color de fondo de todos los diagramas del proyecto actual cambiar al color elegido.

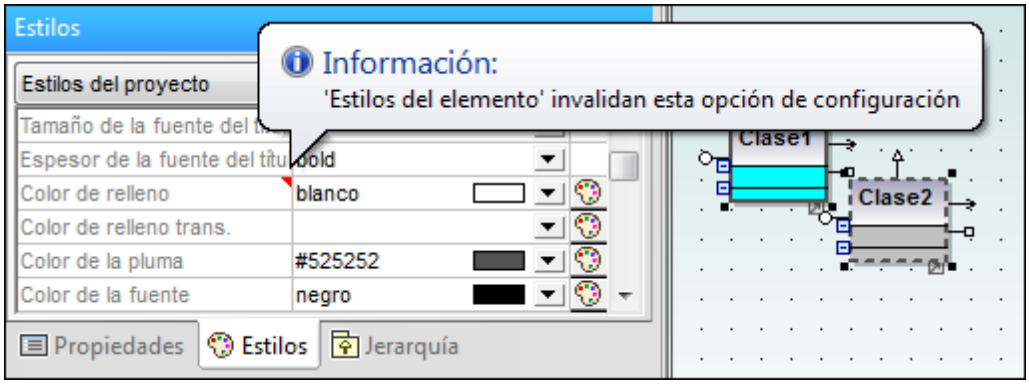
¿Cómo se ven los estilos cuando hay varios elementos seleccionados?

Si selecciona varios elementos en el área de trabajo, todos los valores de estilo aparecen en el campo correspondiente de la pestaña *Estilos*. En la imagen siguiente, p. ej., se seleccionaron **Clase1** y **Clase2**. El campo `Color de relleno` muestra los valores de ambos elementos (es decir, `aguamarina` y `plata`).



Para ver estilos en cascada:

Si un estilo es reemplazado en un nivel más concreto, entonces aparece un pequeño triángulo rojo en el campo del estilo en la pestaña *Estilos*.
Al pasar el puntero del ratón por encima del campo aparece información sobre el orden de prioridad de los estilos.



Por ejemplo, los elementos **`Enumeración`**, **`Paquete`** y **`Perfil`** tienen definido su color de relleno predeterminado en el contexto **`Estilos de la familia de elementos`**. Para cambiar el color de fondo a nivel de proyecto, borre el valor en el contexto **`Estilos de la familia de elementos`** (seleccione la entrada vacía de la lista), seleccione **`Estilos del proyecto`** y cambie aquí el color de relleno.

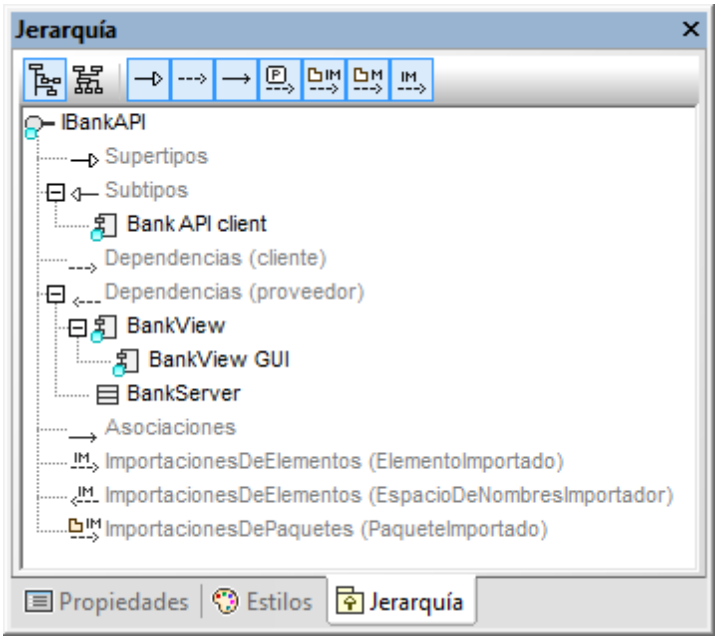
4.6 Jerarquía

La pestaña *Jerarquía* muestra de dos formas distintas todas las relaciones que tiene el elemento seleccionado. El elemento de modelado se puede seleccionar en el área de trabajo del diagrama, en la *Estructura del modelo* o en la pestaña *Favoritos*.

Nota: la configuración general de la pestaña *Jerarquía* se define en la pestaña *Vista* del cuadro de diálogo "Opciones locales" (**Herramientas | Opciones**), en el grupo de opciones *Jerarquía*.

Para ver las relaciones del elemento en forma jerárquica (de árbol):

Seleccione el elemento y después pulse el icono  de la barra de herramientas de la pestaña *Jerarquía*.

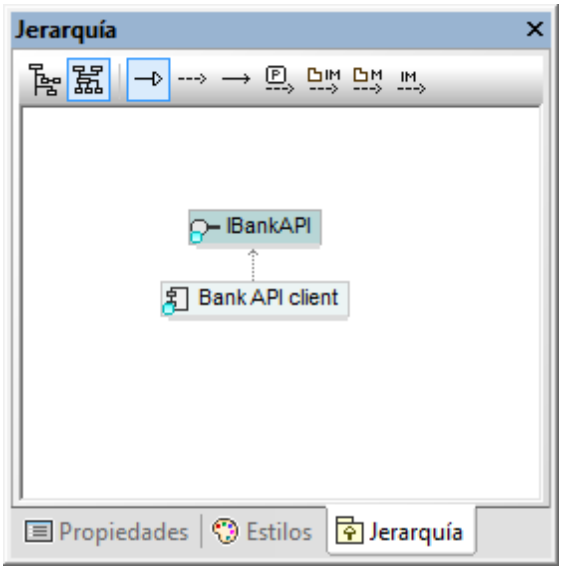


Esta vista de la pestaña *Jerarquía* muestra varias relaciones del elemento seleccionado p. ej. **`IBankAPI`**. Haga clic en los iconos de la barra de herramientas para seleccionar qué tipo de relaciones aparecen en la vista. En la imagen anterior, por ejemplo, están activos todos los iconos. Es decir, la vista muestra todas las relaciones del elemento.

Haga doble clic **en el icono** de un elemento de la lista para ver las relaciones de dicho elemento.

Para ver las relaciones del elemento en forma de diagrama:

Seleccione el elemento y después pulse el icono  de la barra de herramientas de la pestaña *Jerarquía*.



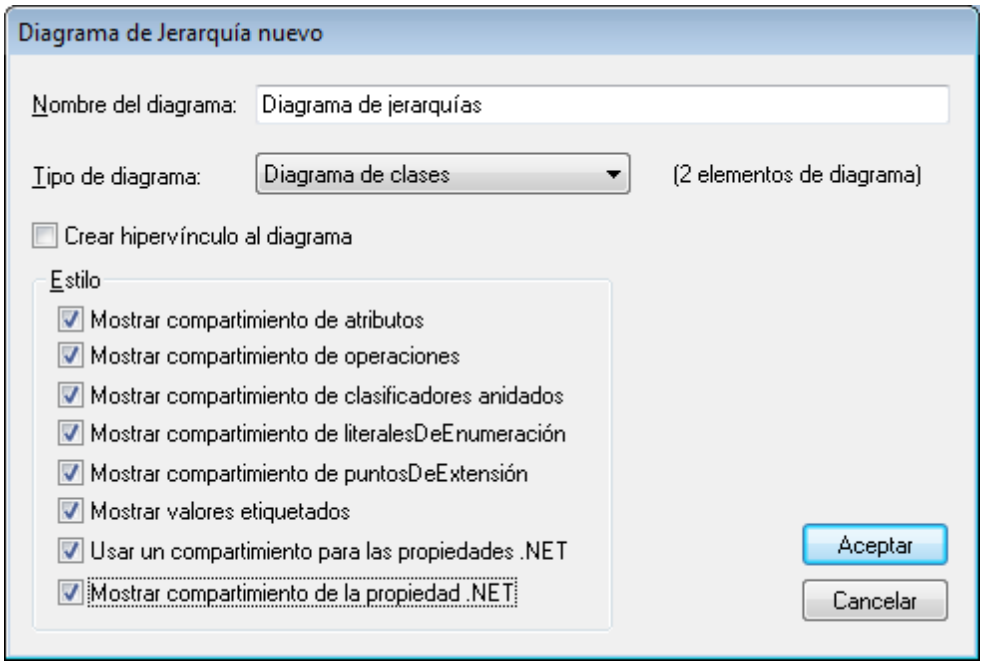
Esta vista de la pestaña *Jerarquía* resume de forma jerárquica un solo conjunto de relaciones. En esta vista solo puede haber un icono activo en la barra de herramientas. En la imagen anterior, por ejemplo, está activo el icono **Mostrar generalizaciones**.

Haga doble clic **en el icono** de un elemento de la vista (p. ej. `Bank API client`) para ver las relaciones de dicho elemento.

Para crear un diagrama nuevo a partir del contenido de la pestaña *Jerarquía*:

El contenido actual de la vista en forma de diagrama de la pestaña *Jerarquía* puede verse en un diagrama nuevo.

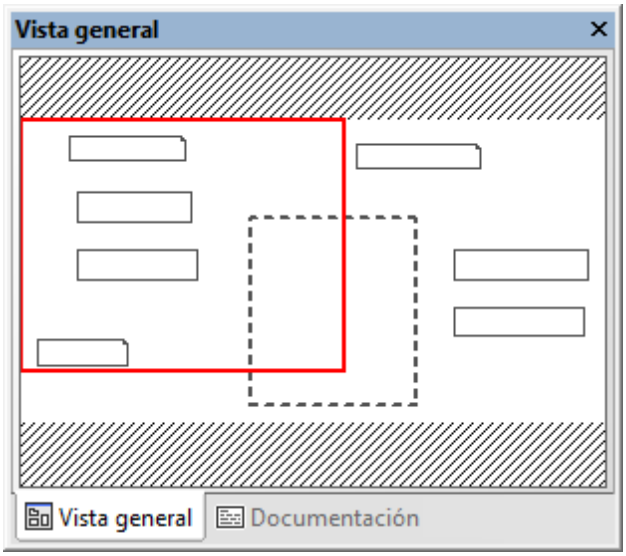
1. Haga clic con el botón derecho en el fondo de la pestaña y seleccione **Crear diagrama basado en este gráfico**.



2. Edite el nombre del diagrama, seleccione las opciones de estilo y haga clic en **Aceptar**.

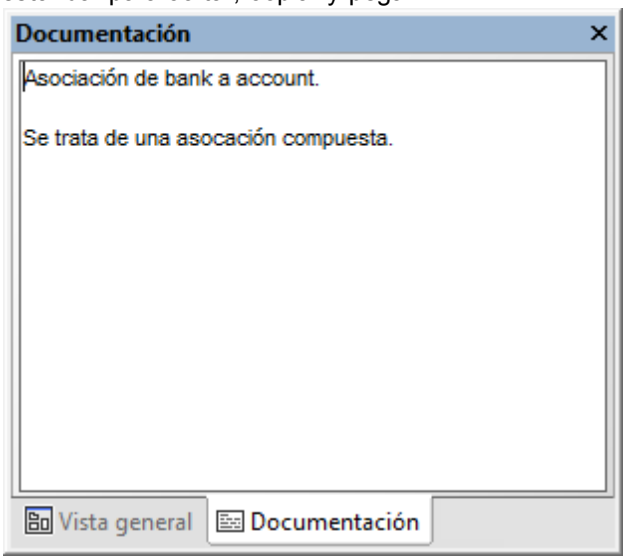
4.7 Vista general

La pestaña *Vista general* ofrece una vista resumen del diagrama activo. Haga clic en el rectángulo rojo y arrástrelo para desplazarse por la vista del diagrama.



4.8 Documentación

La pestaña *Documentación* sirve para documentar los elementos UML que están disponibles en la pestaña *Estructura del modelo*. Haga clic en el elemento que desea documentar y escriba sus comentarios en la pestaña *Documentación*. Aquí puede usar las teclas de acceso rápido estándar para cortar, copiar y pegar.



Documentación y generación de código:
Durante la generación de código la única entrada/salida es la documentación de clases e interfaces. Esto incluye la documentación definida para propiedades y operaciones de clase/interfaz.

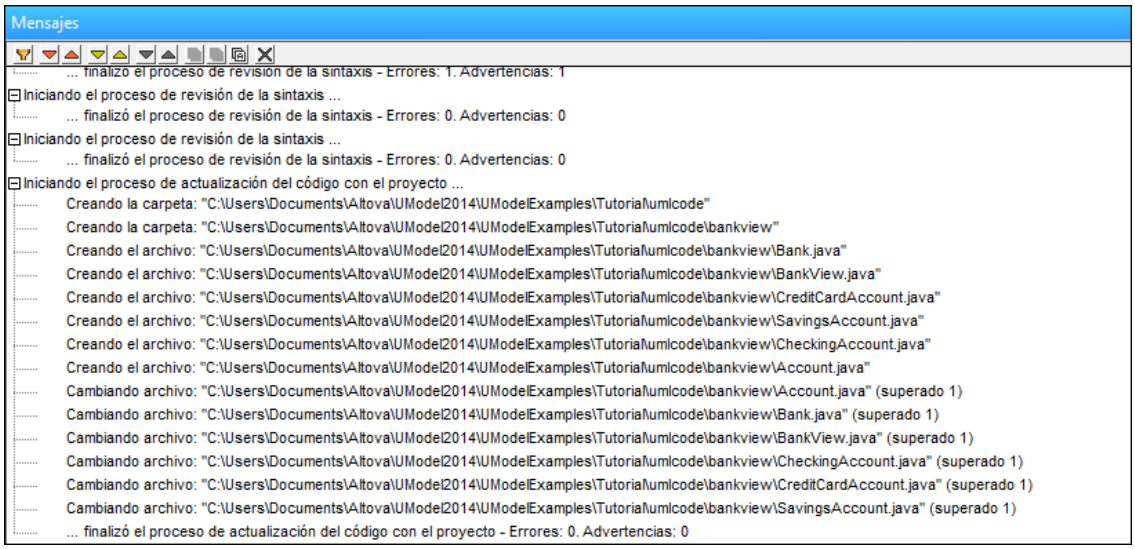
1. Seleccione la opción de menú **Proyecto | Configuración del proyecto**.
2. Marque la casilla *JavaDocs como documentación*.

Nota: cuando importe esquemas XML, en la pestaña *Documentación* solo aparece la primera anotación de un tipo complejo o simple.

4.9 Mensajes

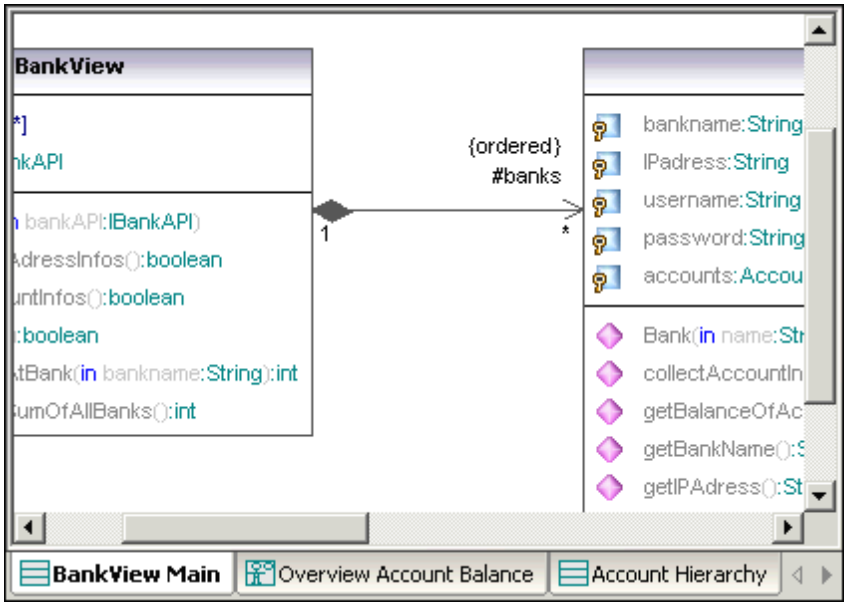
La ventana Mensajes muestra mensajes de advertencia, sugerencias y errores durante las operaciones de combinación de código y revisión de la sintaxis del proyecto.

Al hacer clic en un mensaje de error el elemento correspondiente se resalta en la *Estructura del modelo* y en el diagrama pertinente, si está activo/abierto.



4.10 Panel de diagramas

El panel de diagramas es el área de trabajo principal. Aquí se abren los diagramas de UModel en forma de pestañas.



Para crear un diagrama nuevo:

1. En la *Estructura del modelo* haga clic con el botón derecho en un paquete.
2. Seleccione el comando **Diagrama nuevo | <tipo de diagrama>**.

Para crear un diagrama nuevo con el contenido de un paquete ya existente:

1. En la *Estructura del modelo* haga clic con el botón derecho en un paquete.
2. Seleccione el comando **Mostrar en un diagrama nuevo | Contenido**.

Para abrir un diagrama tiene dos opciones:

- En la *Estructura del modelo*, en *Favoritos* o en el *Árbol de diagramas* haga doble clic **en el icono** del diagrama que quiere abrir.
- Haga clic en la pestaña del diagrama al que desea acceder en el área de trabajo principal.

Para cerrar todos los diagramas menos el diagrama activo:

- Haga clic con el botón derecho en la pestaña del diagrama que debe seguir abierto y seleccione la opción **Cerrar ventanas inactivas**.

Para eliminar un diagrama:

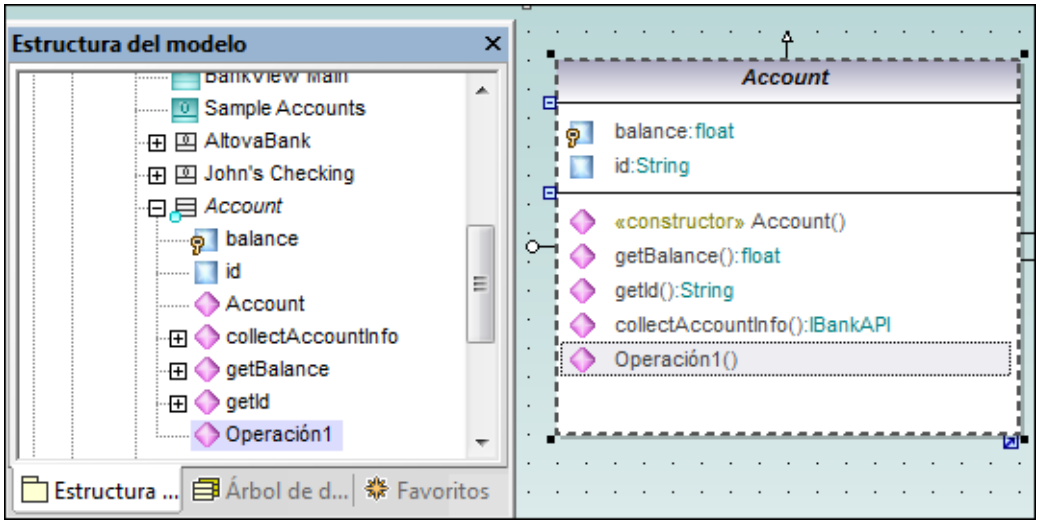
- En la *Estructura del modelo* haga clic en el icono del diagrama que desea eliminar y pulse la tecla **Supr**.

Para mover los diagramas:

- En la *Estructura del modelo* haga clic en el icono del diagrama y arrástrelo a otro paquete.
  A veces es necesario habilitar la opción **No ordenar** para poder mover el diagrama.

Para buscar (eliminar) propiedades de clase y opciones desde la *Estructura del modelo*:
Las propiedades y opciones también se pueden eliminar desde la *Estructura del modelo* directamente. Para hacerlo bien es importante buscar primero la propiedad que deseamos eliminar. Imaginemos que insertó la `Operación1` en la clase **`Account`** (pulse **F8** y **Entrar** para insertarla):

1. Ahora haga clic con el botón derecho en `Operación1` de la clase **`Account`**.
2. Seleccione la opción **Seleccionar en la estructura del modelo** (o pulse **F4**).
   El componente `Operación1` aparece resaltado debajo de *Account* en la pestaña *Estructura del modelo*.



3. Pulse la tecla **Supr** para eliminar la operación de la clase y del proyecto.
   Recuerde que casi todos los elementos de modelado aparecen en la *Estructura del modelo* cuando se pulsa **F4**.

Nota: también puede navegar hasta la *Estructura del modelo* desde el panel *Propiedades*. Para más información consulte el apartado Propiedades de la sección *Interfaz del usuario*.

Para eliminar elementos de un diagrama y del proyecto:

- Seleccione el elemento y pulse la tecla **Supr**.

Para eliminar elementos en el diagrama (y no del proyecto):

- Seleccione el elemento que desea eliminar en el diagrama y pulse **Ctrl+Supr**.

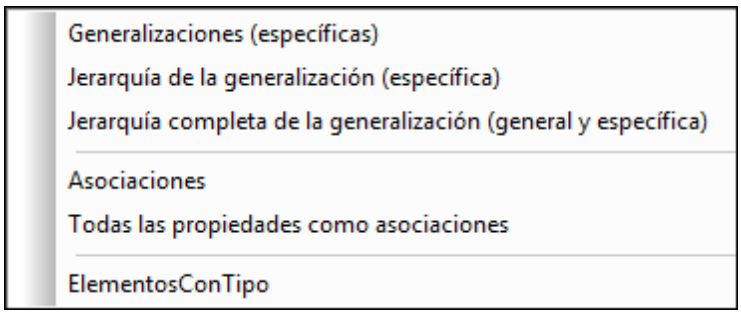
UModel también ofrece una función de diseño automático con la que puede definir la estructura visual de los diagramas. Haga clic con el botón derecho en el fondo del diagrama y en el menú contextual seleccione uno de estos comandos:

- **Aplicar diseño automática a todo | Diseño dirigido por fuerzas**

- **Aplicar diseño automática a todo | Diseño jerárquico**
- **Aplicar diseño automática a todo | Diseño por bloques**

Para ver las relaciones que existen entre los elementos de modelado:

Haga clic con el botón derecho en el elemento que le interesa y seleccione **Mostrar**. Este submenú ofrece diferentes opciones dependiendo del elemento seleccionado.

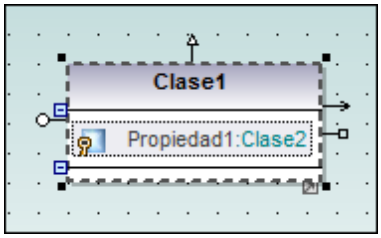


Para ver / ocultar las etiquetas de texto:

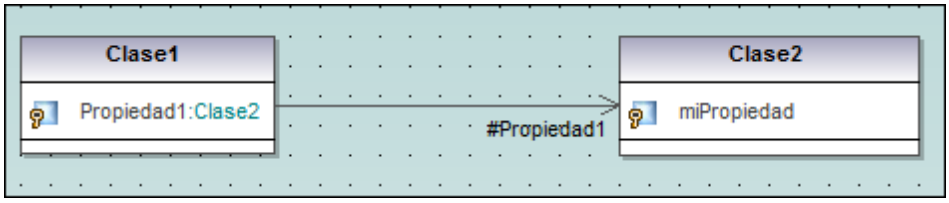
Haga clic con el botón derecho en una clase o en una flecha de asociación y seleccione **Etiquetas de texto | Mostrar (ocultar) todas las etiquetas de texto**.

Para ver un atributo / una propiedad de una clase en forma de asociación:

1. Haga clic con el botón derecho en la propiedad de la clase.



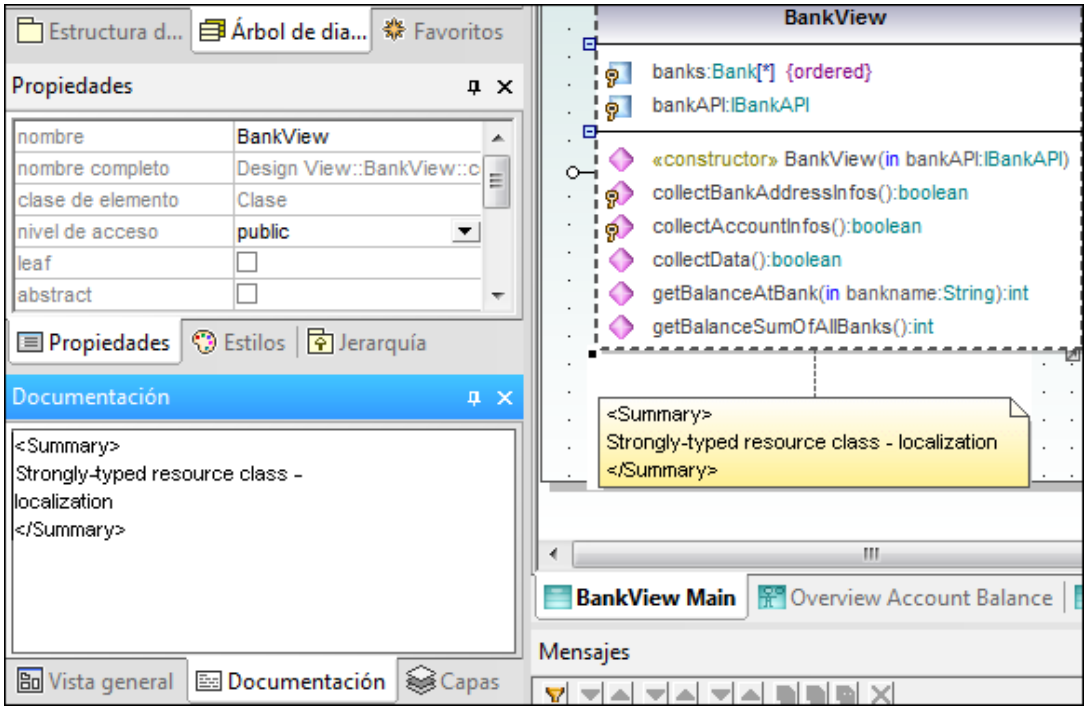
2. Seleccione la opción **Mostrar | <nombre de la propiedad> como asociación**. Esto inserta/abre la clase a la que se hace referencia y muestra la correspondiente asociación.



Para ver los comentarios del código fuente en el diagrama:

- Haga clic con el botón derecho en la clase en el panel del diagrama y seleccione **Mostrar | Comentarios de anotación**.

  Esto permite editar los `DocComment` del código fuente desde el diagrama directamente (los `DocComment` también se pueden ver en la pestaña *Documentación*).



Para ajustar el grosor (espesor) de las líneas en un proyecto:

1. Haga clic en cualquier diagrama y después en la pestaña *Estilos*.
   Compruebe que en el cuadro combinado superior está activa la opción **`Estilos del proyecto`**.
2. Desplácese hasta la entrada **`Grosor de la línea`** y seleccione un valor.
   Este estilo se aplica a las líneas de asociación, agregación, generalización, etc. de todos los diagramas del proyecto actual.

4.10.1 Propiedades de los diagramas

Para configurar las propiedades de los diagramas:

1. Haga clic en el fondo del área de trabajo.
2. Seleccione uno de los estilos de la pestaña *Estilos*.
   Para más información consulte el apartado *Estilos*.

Para aumentar el tamaño del diagrama:

El tamaño de la pestaña del diagrama depende de los elementos del diagrama y de su posición.

- Arrastre un elemento hasta el borde del diagrama para aumentar el tamaño del diagrama automáticamente.

Para cambiar los elementos de posición con ayuda de la cuadrícula:

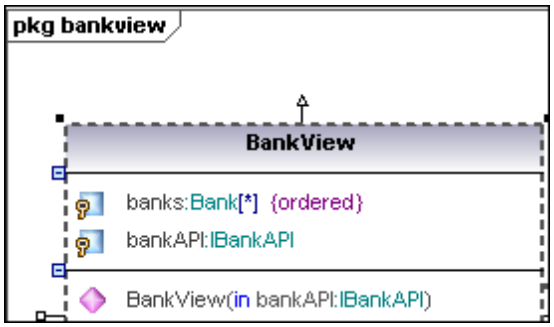
Puede cambiar a mano la posición de los elementos de modelado o puede usar una opción para colocarlos sobre las líneas de la cuadrícula automáticamente.

 Este icono de la barra de herramientas sirve para mostrar/ocultar la cuadrícula.

 Este icono de la barra de herramientas sirve para ajustar o no los elementos a la cuadrícula visible/invisible automáticamente.

Para ver el título del diagrama UML:

 Este comando muestra/oculta el título del diagrama UML, es decir, el marco que rodea el diagrama y que incluye el nombre del diagrama en la esquina superior izquierda.



Para alinear los elementos de modelado:

Puede alinear los elementos y ajustar su tamaño en relación a otros elementos. P. ej. puede centrar todos los elementos o alinearlos a la izquierda o a la derecha.

Nota: cuando marque varios objetos, el comando toma como referencia el **último** elemento que seleccionó. Por ejemplo, si marca tres elementos `clase` y hace clic en el comando **Igualar ancho**, las tres clases cobrarán el ancho de la última clase que seleccionó. Lo mismo ocurre cuando se marcan varios objetos creando un recuadro con el puntero del ratón. El último elemento que se selecciona con el recuadro se toma como referencia para los demás objetos.

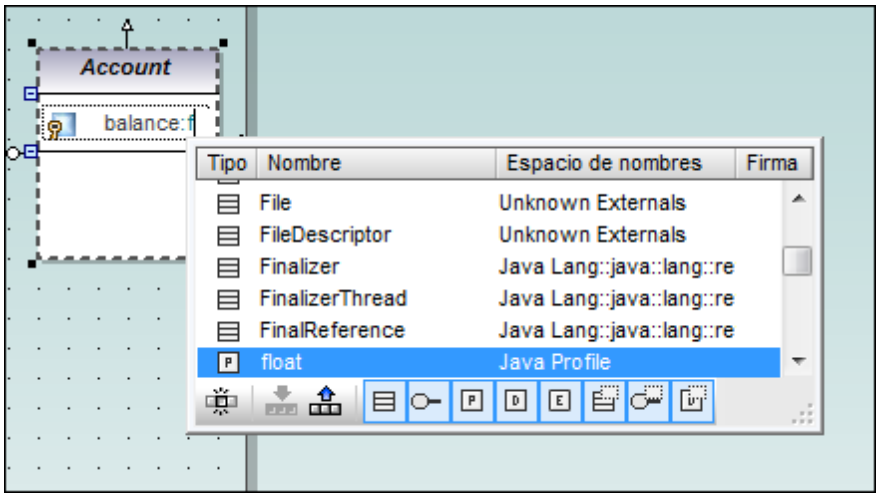
Estos son los iconos de la barra de herramientas anterior (de izquierda a derecha):

- **Alinear a la izquierda**
- **Alinear a la derecha**
- **Alinear arriba**
- **Alinear abajo**
- **Centrar verticalmente**
- **Centrar horizontalmente**
- **Espaciar en horizontal**
- **Espaciar en vertical**
- **Poner en fila horizontal**
- **Poner en fila vertical**
- **Igualar ancho**
- **Igualar alto**
- **Igualar tamaño**

Ventana de finalización automática: cómo seleccionar tipos de datos

Cuando necesite insertar tipos de datos para operaciones o propiedades, la ventana de finalización automática se abre automáticamente. Esta ventana ofrece estas características y opciones:

- Para ordenar la ventana en orden ascendente/descendente: haga clic en el nombre de una columna.
- Para ajustar el tamaño de la ventana: arrastre la esquina inferior derecha de la ventana.
- Para filtrar el contenido de la ventana: sírvase de los iconos situados en la parte inferior de la ventana.
- Para trabajar con cada categoría, una por una, active el icono **Modo único** . Desactive el modo único si prefiere trabajar con diferentes categorías a la vez. Por ejemplo, en la imagen siguiente, el modo único no está seleccionado.
- Puede activar/desactivar todas las categorías con los iconos **Activar todas las categorías** y **Desactivar todas las categorías**.
- Si la finalización automática está deshabilitada, pulse **Ctrl+Barra espaciadora** para abrir la ventana de finalización automática.



Categorías:
Clase, Interfaz, TipoPrimitivo, TipoDeDatos, Enumeración, Plantilla de clase, Plantilla de interfaz, Plantilla de tipo de datos.

Nota: la función de finalización automática se puede habilitar/deshabilitar en la pestaña *Edición de diagramas* del cuadro de diálogo "Opciones locales" (**Herramientas | Opciones**), activando/desactivando la casilla *Habilitar ayudante de entrada automática*.

4.10.2 Cortar, copiar y pegar en los diagramas de UModel

Cortar, copiar y pegar elementos dentro del panel del diagrama

Todos los elementos de los diagramas de UModel se pueden cortar, copiar y pegar dentro de elementos del mismo tipo e incluso de otro. Para ello puede usar también teclas de acceso rápido.

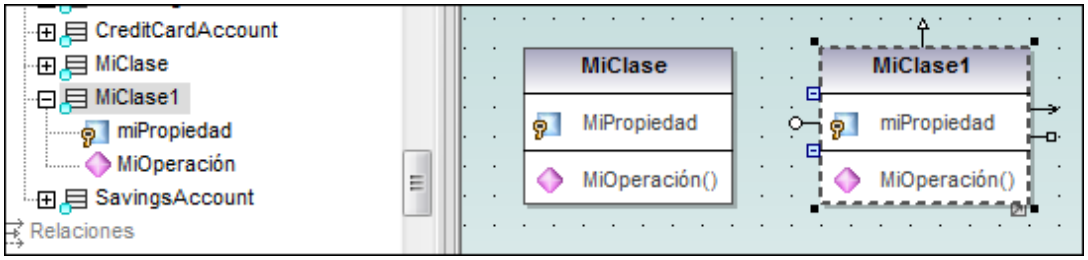
Tras copiar el elemento:

- Puede **pegarlo** usando la combinación de teclas **Ctrl+V**, el comando **Pegar** del menú contextual o el comando **Pegar** del menú **Edición**. Esto añade un **elemento nuevo** al diagrama y a la *Estructura del modelo*.
- Puede **pegarlo sólo en el diagrama** usando el menú contextual (clic con el botón derecho en el fondo del diagrama y después en **Pegar sólo en el diagrama**). Esto añade una vista (o vínculo) del elemento ya existente en el diagrama actual y no en la *Estructura del modelo*.

Por ejemplo, imaginemos que estamos trabajando con un diagrama de clases:

Para pegar (Ctrl+V) una clase copiada previamente:

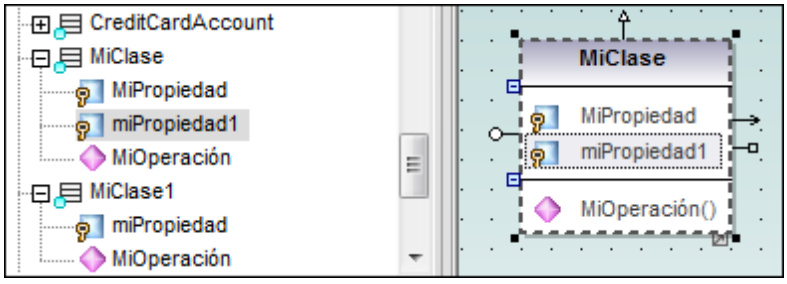
- Si pega la clase copiada **en el mismo diagrama o paquete**, se inserta una **clase nueva** con el nombre de clase original más un número secuencial. Por ejemplo, si el nombre original era `MiClase`, la clase pegada se llamará `MiClase1`. En la clase nueva también se copian todas las operaciones y propiedades de la clase original.



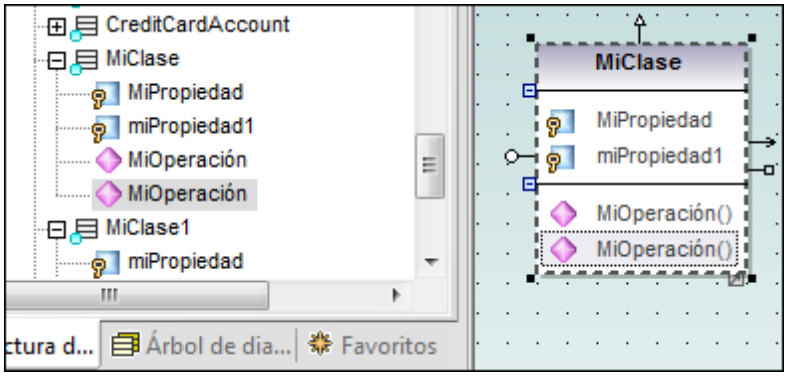
- Si pega la clase copiada **en un paquete distinto**, se inserta una clase nueva que se llama igual que la clase original.
- En ambos casos la clase nueva también se añade a la *Estructura del modelo*.

Para pegar (Ctrl+V) propiedades y operaciones copiadas previamente:

- Si pega la propiedad copiada **en la misma clase**, se inserta una **propiedad nueva** con el nombre de propiedad original más un número secuencial (p. ej. `MiPropiedad1`).



- Si pega la operación copiada **en la misma clase**, se inserta una **operación nueva** que se llama igual que la operación original.

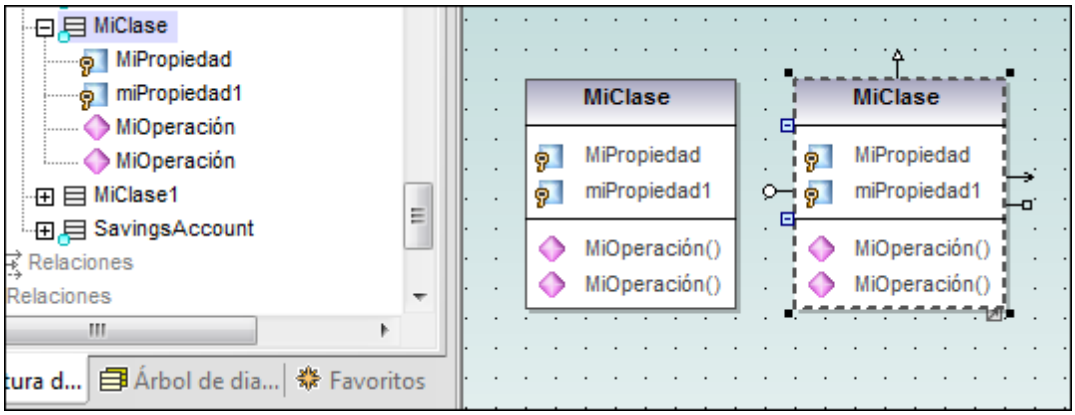


- En ambos casos la operación/propiedad nueva también se añade a la *Estructura del modelo*.

Pegar sólo en el diagrama

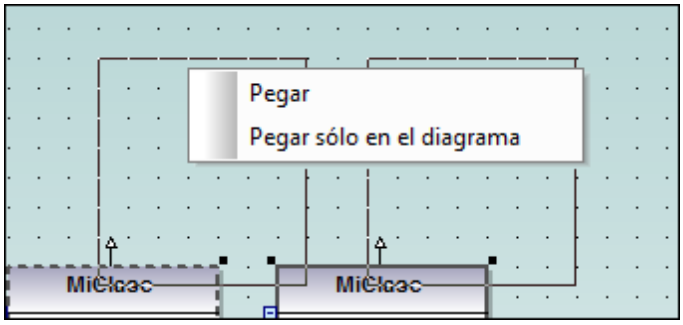
Cuando use este comando del menú contextual, se crea un *enlace* o una *vista* del elemento en el **diagrama** donde se ejecuta la operación. Sigamos con el ejemplo de nuestro diagrama de clases:

- La opción **Pegar sólo en el diagrama** del menú contextual crea una vista de la clase original.
- La clase se inserta en el diagrama y aparece exactamente igual que la clase original.
- En la *Estructura de modelo* no se añade ninguna clase nueva.
- El nombre de la clase y de sus propiedades/operaciones no cambia.
- Si cambia las propiedades en una de las *vistas*, las propiedades cambian en la otra *vista* automáticamente.



Para copiar y pegar elementos con el ratón:

1. Haga clic en el elemento de modelado que desea copiar.
2. Arrastre el puntero del ratón hasta la posición donde quiere colocar el nuevo elemento.
3. Pulse la tecla **Ctrl**. Aparece un signo **+** bajo el puntero del ratón. Esto significa que se trata de una operación de copia.
4. Suelte el botón del ratón.



Aparece un menú contextual donde puede elegir entre **Pegar** y **Pegar sólo en el diagrama**.

5. Elija una opción.

Nota: utilice este método (botón del ratón + tecla **Ctrl**) para copiar o mover propiedades y operaciones dentro de una clase.

4.11 Agregar/insertar elementos en los modelos

En UModel los elementos se pueden crear e insertar en los diagramas de varias maneras:

- Añadiendo elementos a determinados paquetes, desde la pestaña *Estructura del modelo.*
- Arrastrando elementos ya existentes desde la pestaña *Estructura del modelo.*
- Haciendo clic en un icono determinado e insertando el tipo de elemento en el diagrama.
- Usando el menú contextual para agregar elementos al diagrama (y a la *Estructura del modelo* automáticamente).

  Recuerde que en la *Estructura del modelo* puede seleccionar varios elementos con las teclas **Mayús+Clic** o **Ctrl+Clic**.

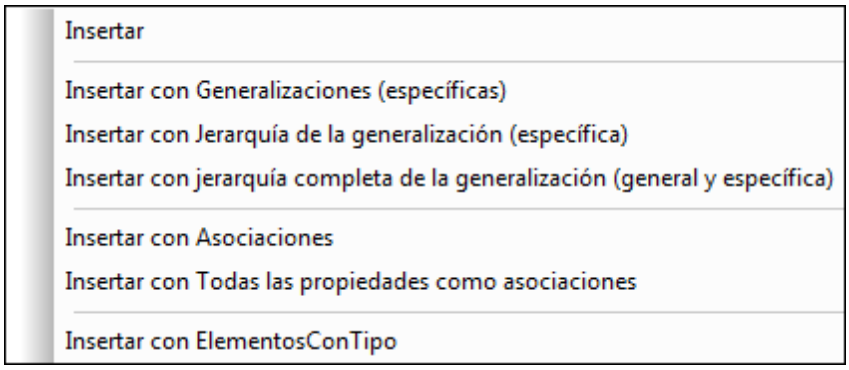
Para agregar elementos en la pestaña *Estructura del modelo* y *Favoritos*:

- Haga clic con el botón derecho en el paquete, seleccione **Nuevo/a** y elija un elemento del submenú.
  El elemento nuevo se añade en la pestaña *Estructura del modelo* en el proyecto actual.

Para insertar elementos de la vista *Estructura del modelo* en un diagrama:

Puede insertar elementos de modelado por separado o en grupo. Para marcar varios elementos haga clic en los elementos mientras pulsa la tecla **Ctrl**. Hay dos maneras de insertar los elementos en el diagrama: arrastrándolos mientras pulsa el botón principal del ratón y arrastrándolos mientras pulsa el botón derecho.

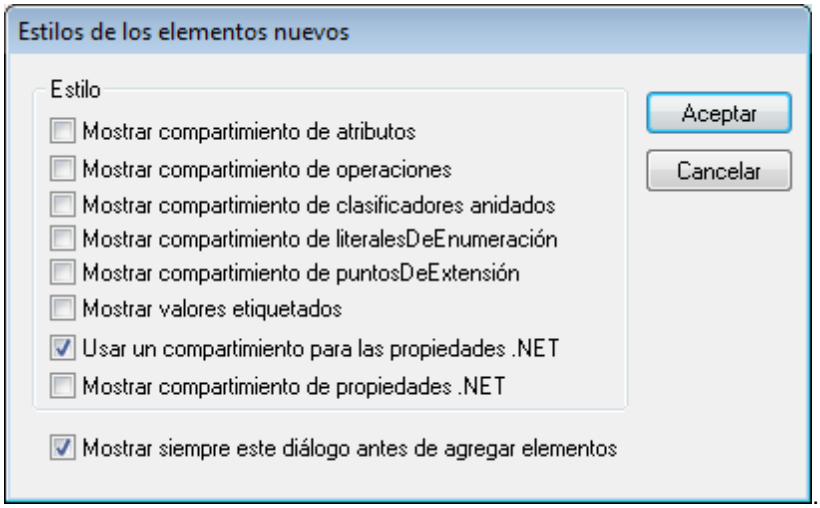
- **Arrastre el elemento mientras pulsa el botón principal** (operación arrastrar y colocar normal) para insertar los elementos inmediatamente en la posición del cursor (las asociaciones, dependencias, etc. que existan entre los elementos insertados y el nuevo se muestran automáticamente).
- **Arrastre el elemento mientras pulsa el botón derecho** para abrir un menú contextual donde puede elegir qué asociaciones y generalizaciones se muestran.



En nuestro ejemplo vamos a intentar crear un duplicado del diagrama **`Account Hierarchy`**.

1. Haga clic con el botón derecho en el paquete **`bankview`** y seleccione **Nuevo diagrama | Diagrama de clases**.
2. Busque la clase abstracta `Account` en la *Estructura del modelo* y arrástrela **mientras pulsa el botón derecho** hasta el nuevo diagrama.
   Aparece el menú contextual de la imagen anterior.

3. Seleccione la opción **Insertar con jerarquía de la generalización (específica)**.



4. Desactive las casillas de los componentes que deben aparecer en los elementos (en este caso, las propiedades y las operaciones).
5. Haga clic en **Aceptar**.
   La clase *Account* y sus tres clases subordinadas se insertan en la pestaña del diagrama. Las flechas de Generalización se muestran automáticamente.

Para agregar elementos a un diagrama con los iconos de la barra de herramientas:

1. Seleccione el icono del elemento que desea insertar en la barra de herramientas.
2. Haga clic en el área de trabajo para insertar el elemento.

Nota: pulse la tecla **Ctrl** mientras hace clic en la pestaña del diagrama para insertar varios elementos del mismo tiempo con cada clic.

Para agregar elementos a un diagrama con el menú contextual:

- Haga clic con el botón derecho en el fondo del diagrama y seleccione la opción **Nuevo/a | <nombre de elemento>**.

Nota: si añade elementos nuevos en el área de trabajo directamente, el mismo elemento se añade automáticamente en la pestaña *Estructura del modelo*. El elemento se añade al paquete que contiene el diagrama UML en la *Estructura del modelo*.

- Haga clic con el botón derecho en un elemento y seleccione **Mostrar | xxxx**
  Por, ejemplo, haga clic con el botón derecho en la clase *Account* y seleccione **Mostrar | Jerarquía de la generalización**. Esto también inserta las clases derivadas en el diagrama.

4.12 Crear hipervínculos entre los elementos de modelado

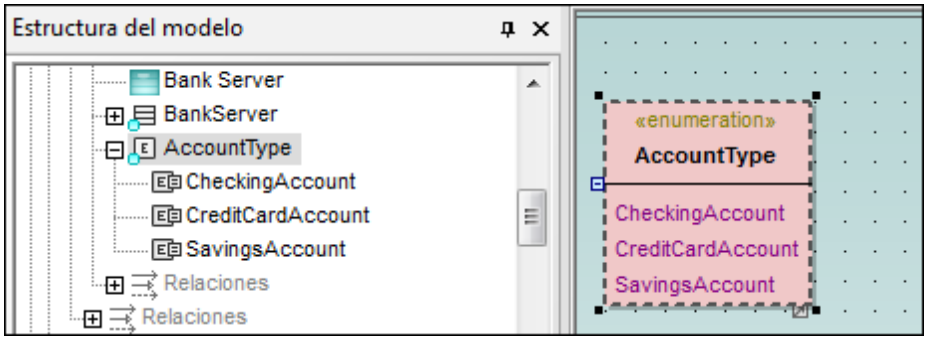
UModel ofrece opciones para crear hipervínculos entre elementos de modelado, tanto automática como manualmente. Los hipervínculos automáticos se crean al importar código fuente o archivos binarios en el modelo.

Los hipervínculos manuales se pueden crear entre casi todos los elementos de modelado (excepto entre las líneas) y:

- cualquier diagrama del proyecto `*.ump` actual
- cualquier elemento de un diagrama
- cualquier elemento de la Estructura del modelo
- documentos externos (p. ej. documentos PDF, Excel o Word)
- páginas web

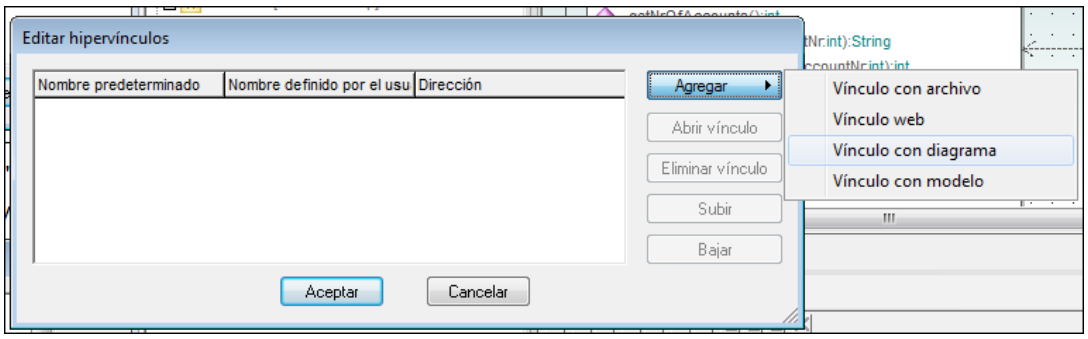
Note: todos los hipervínculos creados manualmente también están disponibles en la documentación HTML del proyecto cuando se genera documentación HTML.

Al abrir el diagrama **`Bank Server`** situado bajo el paquete **`Bank Server`** se puede ver la interfaz **`IBankAPI`**, así como la clase **`BankServer`**. También se puede ver un elemento **`enumeration`** con los nombres de los **`LiteralesDeEnumeración`**. Nuestro objetivo es crear un hipervínculo entre el elemento **`enumeration`** y el diagrama de clases **`Account Hierarchy`**.



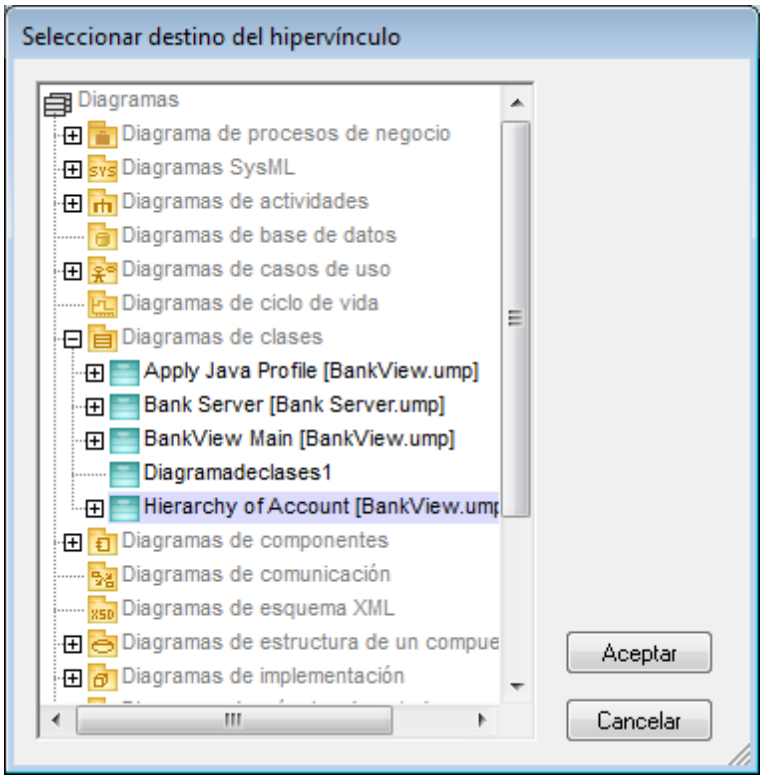
Para crear un hipervínculo de diagrama:

1. Haga clic con el botón derecho en el elemento y seleccione **Hipervínculos | Insertar o editar hipervínculos**.

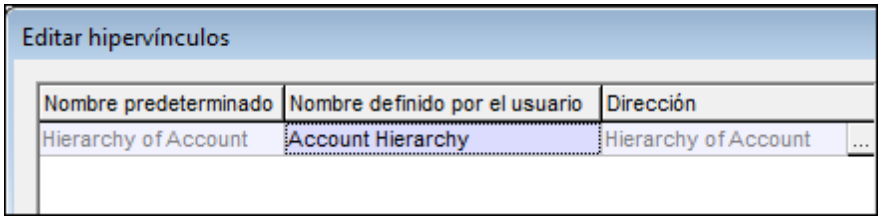


Esto abre el cuadro de diálogo "Editar hipervínculos", que sirve para gestionar los hipervínculos.

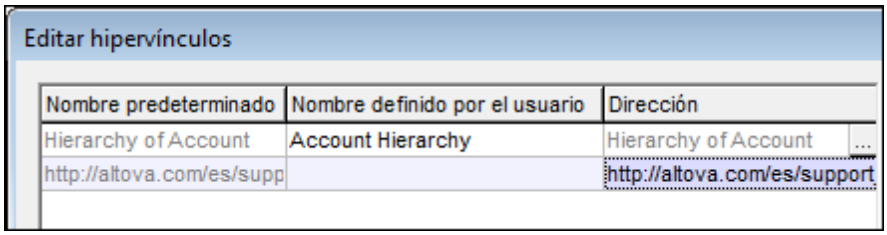
2. Haga clic en el botón **Agregar** y seleccione **Vínculo con diagrama** para definir un vínculo con un diagrama ya existente.



3. Seleccione el destino del hipervínculo (p. ej. el diagrama **`Hierarchy of Account`**) y haga clic en **Aceptar**.

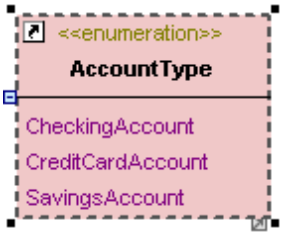


Haga doble clic en la columna *Nombre definido por el usuario* para definir el nombre de vínculo que desea usar.
Recuerde que puede agregar varios vínculos, incluso de tipos diferentes, en el mismo elemento de modelado (p. ej. también puede añadir un vínculo web a http://altova.com/es/support_help.html haciendo clic en **Agregar | Vínculo web**).

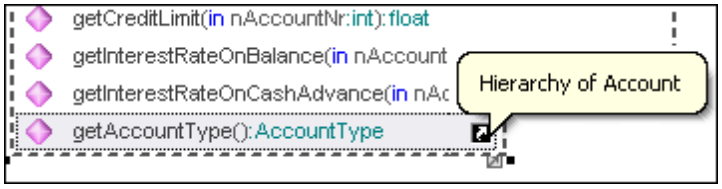


4. Cuando termine de definir los hipervínculos haga clic en **Aceptar**.
Observe que en la esquina superior izquierda del elemento **`enumeration`** se añadió un

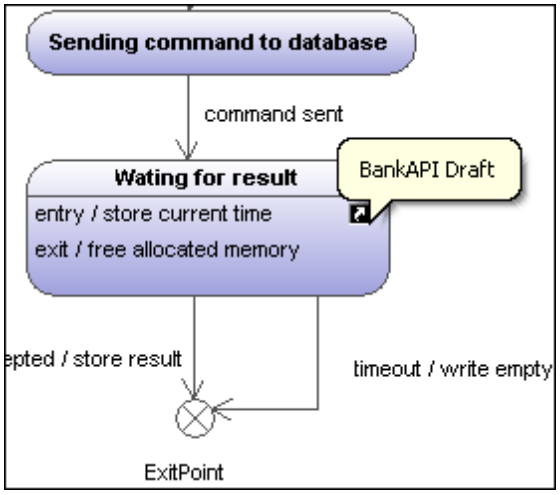
icono en forma de flecha. Pase el cursor por encima del icono para ver el nombre del elemento de destino.



Hipervínculos en operaciones

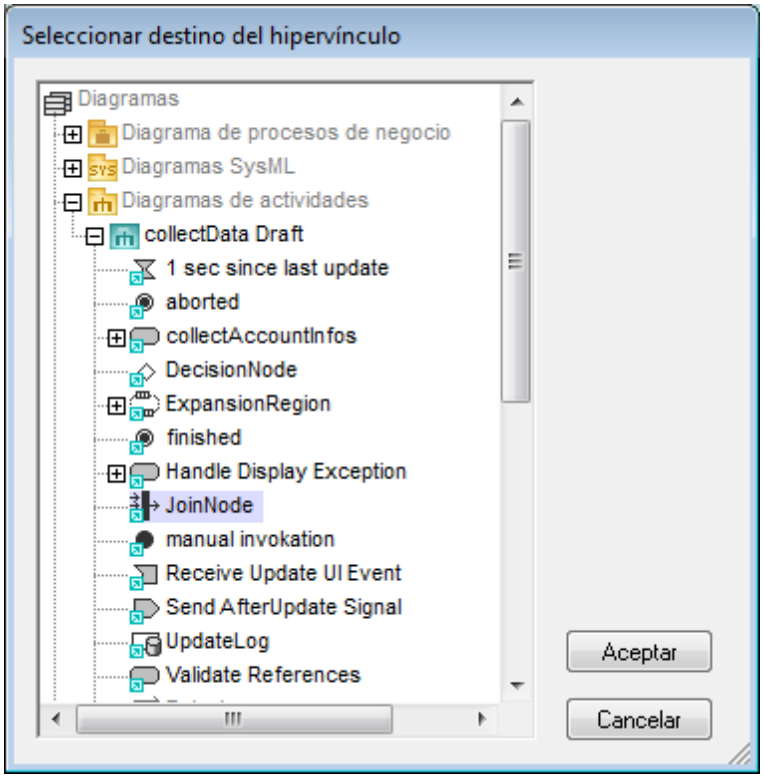


Hipervínculo desde una máquina de estados



Para crear un vínculo a un elemento concreto de un diagrama (vínculo con diagrama):

1. Cree el hipervínculo igual que antes pero esta vez haga clic en el signo **+** para expandir el contenido del diagrama.



2. Seleccione el elemento de modelado al que debe apuntar el vínculo y haga clic en **Aceptar**.

   Al hacer clic en el icono del hipervínculo se abre el diagrama y observará que en él está seleccionado el elemento designado.

Para crear un vínculo a un elemento en la Estructura del modelo (vínculo con modelo):

- Haga lo mismo que en el ejemplo anterior pero esta vez haga clic en **Agregar | Vínculo con modelo** en el cuadro de diálogo.

  Una vez insertado, el hipervínculo conduce al elemento seleccionado en la *Estructura del modelo* desde el cuadro de diálogo.

  Nota: cuando genere documentación del proyecto, los hipervínculos a los elementos de la Estructura del modelo se ajustan para que apunten a las definiciones correctas en la documentación generada.

Para crear un vínculo a un documento:

1. En el cuadro de diálogo "Editar hipervínculos" haga clic en el botón **Agregar | Vínculo con archivo**.
2. Seleccione el documento al que debe apuntar el vínculo (p. ej. `*.doc`, `*.xls`, `*.pdf`, etc.).

Para crear un hipervínculo desde una nota:

1. Seleccione el texto de la nota que debe servir de hipervínculo.
2. Haga clic con el botón derecho en el texto seleccionado y seleccione la opción **Insertar**

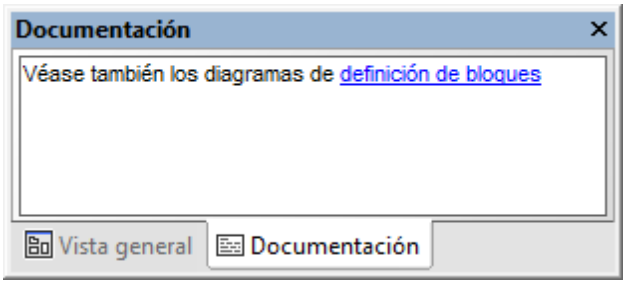
o editar hipervínculos.

3. En el cuadro de diálogo "Editar hipervínculos" cree el vínculo a un diagrama.

Nota: puede usar el mismo método para crear hipervínculos desde elementos `Comentario`.

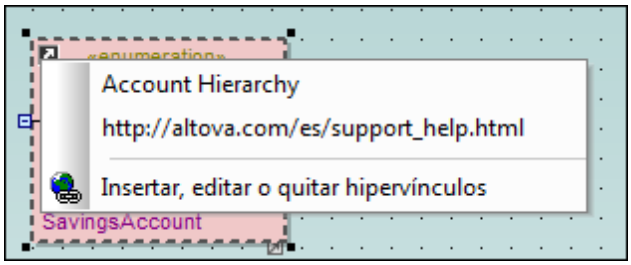
Para crear un vínculo desde la pestaña *Documentación*:

1. Escriba el texto descriptivo en la pestaña *Documentación*.
2. Seleccione el texto que debe servir de hipervínculo.
3. Haga clic con el botón derecho y seleccione qué tipo de vínculo desea crear.



Para navegar hasta el destino del hipervínculo:

1. Haga clic en el icono del hipervínculo del elemento de modelado.
   Si solo definió un destino, el diagrama o la web de destino, por ejemplo, aparecerá inmediatamente.
   Si definió varios destinos, aparece un menú contextual donde puede seleccionar uno de ellos.



   Por ejemplo, en el ejemplo de la imagen anterior la primera opción abre el diagrama `Hierarchy of Account`.

Para navegar por los hipervínculos:

- Haga clic en los iconos **Anterior** y **Siguiente** de la barra de herramientas principal para navegar por el origen y el destino de los vínculos.

Para editar/cambiar el destino de un hipervínculo:

1. Haga clic con el botón derecho en el icono del vínculo y seleccione la opción **Insertar, editar o quitar hipervínculos**.

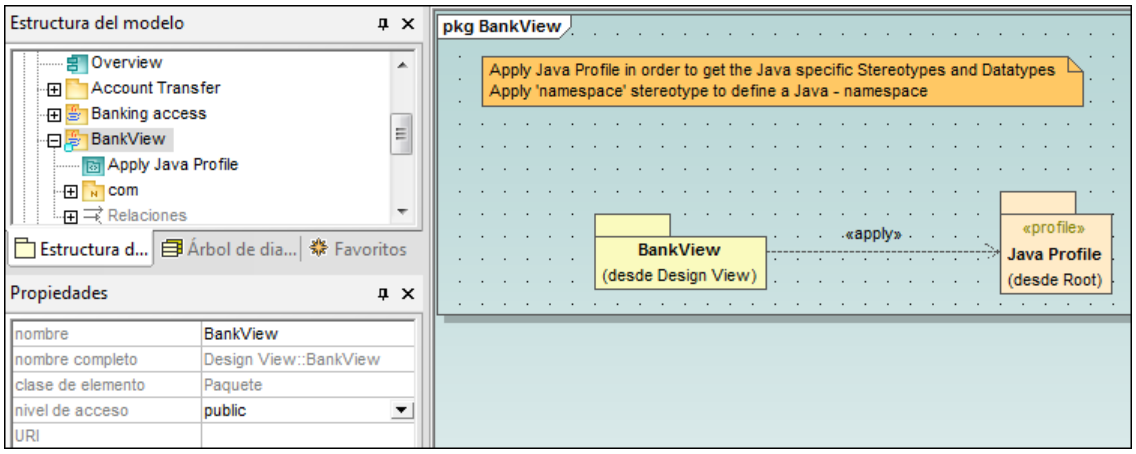
2. Realice los cambios necesarios en el cuadro de diálogo "Editar hipervínculos".

4.13 Ejemplos Bank

La carpeta ...**UModelExamples** contiene archivos de ejemplo que muestran diferentes aspectos del modelado UML en UModel. Estos archivos están diseñados para ilustrar modelos propios de los lenguajes Java y C# y de una mezcla de ambos.

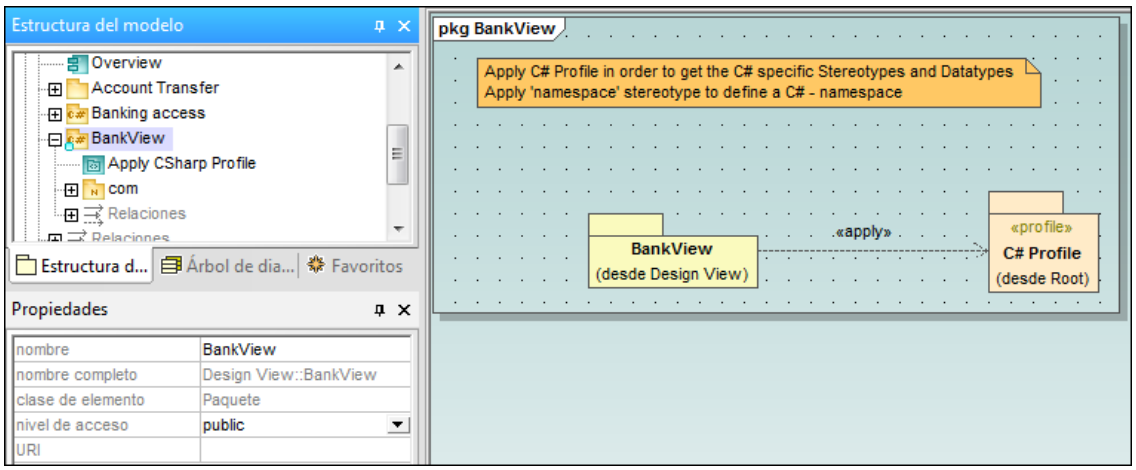
El archivo de ejemplo **Bank_Java.ump** (*imagen siguiente*) tiene estas características:

- El perfil **`Java`** está asignado al paquete **`Bankview`**.
- La raíz de espacio de nombres Java está asignada a los paquetes **`Banking access`** y **`BankView`**.
- El paquete **`Interaction View`** contiene dos elementos de interacción, con sendos diagramas de secuencia.



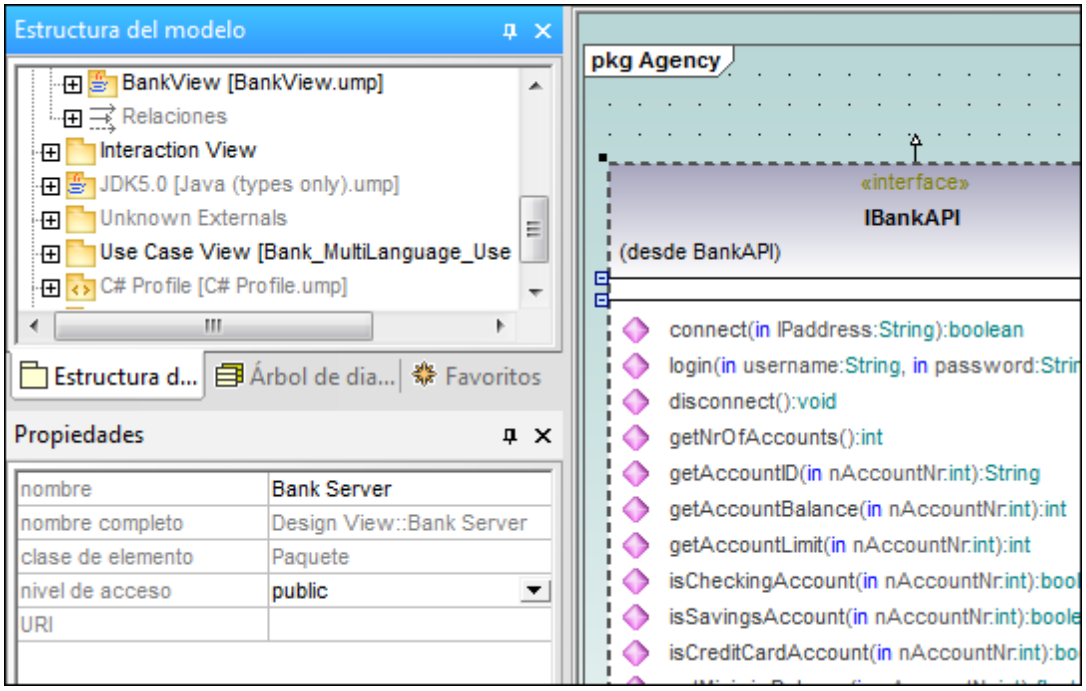
El archivo de ejemplo **Bank_CSharp.ump** (*imagen siguiente*) tiene estas características:

- El perfil C# está asignado al paquete **`BankView`**.
- La raíz de espacio de nombres C# está asignada a los paquetes **`Banking access`** y **`BankView`**.
- El paquete **`Interaction View`** contiene dos elementos de interacción, con sendos diagramas de secuencia.



El archivo de ejemplo **Bank_MultiLanguage.ump** (*imagen siguiente*) tiene estas características:

- El perfil Java está asignado al paquete **`BankView`**.
- La raíz de espacio de nombres C# está asignada al paquete **`Bank Server`**.
- La raíz de espacio de nombres Java está asignada al paquete **`BankView`**.
- El paquete **`Interaction View`** contiene dos elementos de interacción, con sendos diagramas de secuencia.
- El proyecto está dividido en 4 subproyectos editables: **`Bank Server.ump`**, **`Banking access.ump`**, **`BankView.ump`** y **`Bank_Multilanguage_Use Case View.ump`**.



Chapter 5

Interfaz de la línea de comandos

5 Interfaz de la línea de comandos

Además de una interfaz gráfica del usuario, UModel ofrece una interfaz de la línea de comandos. Para abrirla debe ejecutar el archivo `UModelBatch.exe` que está en el directorio `C:\Archivos de programa\Altova\UModel2017`. Si trabaja con la versión de 32 bits de UModel en un sistema operativo de 64 bits, entonces la ruta de acceso del archivo `UModelBatch.exe` es `C:\Archivos de programa (x86)\Altova\UModel2017`.

En este apartado puede consultar la sintaxis de parámetros de la línea de comandos. Para ver la sintaxis en una ventana del símbolo del sistema, escriba `umodelbatch /?` en la línea de comandos.

Nota: si contiene espacios, la ruta de acceso o el nombre de archivo debe ir entre comillas (p. ej. `"c:\Archivos de programa\...\MiProyecto.ump"`)

```
utilización: umodelbatch [proyecto] [opciones]

/? o /help ... mostrar esta información de ayuda

proyecto                ... archivo de proyecto (*.ump). Ver también
Archivo: Nuevo / Cargar / Opciones al guardar
/new[=archivo]    ... crear, guardar o guardar como proyecto nuevo
/set                    ... establecer operaciones permanentes
/gui                    ... mostrar la interfaz de usuario de UModel


comandos (ejecutados en este orden):
/chk                    ... revisar la sintaxis del proyecto
/isd=ruta               ... importar directorio de código fuente
/isp=archivo      ... importar archivo de proyecto de código fuente
                 (*.project,*.xml,*.jpx,*.csproj,*.csdproj,*.vbproj,*.vbdpro
j,*.sln,*.bdsproj)
/ibt=lista              ... importar tipos binarios (especificar lista de
[nombres de tipos] binarios)
                (";"=separador, "*"=todos los tipos, "#" antes de los
nombres de ensamblado)
/ixd=ruta               ... importar directorio del esquema XML
/ixs=archivo      ... importar archivo de esquema XML (*.xsd)
/m2c                    ... actualizar código de programa con el modelo
(exportar/ingeniería directa)
/c2m                    ... actualizar modelo con el código de programa
(importar/ingeniería inversa)
/ixf=archivo      ... importar archivo XMI
/exf=archivo      ... exportar a archivo XMI
/inc=archivo      ... incluir archivo
/mrg=archivo      ... combinar archivo
/doc=archivo      ... escribir documentación en el archivo especificado
/lue[=cpri]             ... mostrar una lista de los elementos que no se
utilizan en ningún diagrama (es decir, no utilizados)
/ldg                    ... mostrar una lista de todos los diagramas
/lcl                    ... mostrar una lista de todas las clases
/lsp                    ... mostrar una lista de todos los paquetes
```

```
compartidos
/lip                    ... mostrar una lista de todos los paquetes
incluidos


opciones para guardar como proyecto nuevo:
/npad=opc   ... ajustar rutas de acceso relativas (Yes | No |
MakeAbsolute), es decir Sí, No o Convertir en absolutas


opciones para comandos de importación:
/iclg=leng  ... lenguaje de código (Java1.4 | Java5.0 | Java6.0 | Java7.0 |
Java8.0 | C#1.2 | C#2.0 | C#3.0 | C#4.0 | C#5.0 | C#6.0 | C#7.0 | VB7.1 |
VB8.0 | VB9.0)
/ipsd[=0|1] ... procesar subdirectorios (recursivo)
/irpf[=0|1] ... importar relativos al archivo de proyecto de UModel
/ijdc[=0|1] ... JavaDocs como comentarios de Java
/icdc[=0|1] ... DocComments como comentarios de C#
/icds[=lst] ... Símbolos definidos de C#
/ivdc[=0|1] ... DocComments como comentarios de VB
/ivds[=lst] ... Símbolos definidos de VB (constantes personalizadas)
/imrg[=0|1] ... sincronizar combinados
/iudf[=0|1] ... utilizar filtro de directorios
/iflt[=lst] ... filtro de directorios (restablece /iudf)


opciones para importar tipos binarios (después de /iclg):
/ibrt=vers  ... versión en tiempo de ejecución
/ibpv=path  ... reemplazo de la variable PATH para buscar bibliotecas de
código nativo
/ibro[=0|1] ... usar el contexto de sólo reflexión
/ibua[=0|1] ... usar la opción de agregar tipos referenciados con filtro de
paquetes
/ibar[=flt] ... agregar el filtro de paquetes de tipos referenciados
(restablece /ibua)
/ibot[=0|1] ... sólo importar los tipos
/ibuv[=0|1] ... utilizar el filtro de nivel de acceso mínimo
/ibmv[=key] ... palabra clave para el nivel de acceso mínimo necesario
(restablece /ibuv)
/ibsa[=0|1] ... omitir secciones de atributo o modificadores de anotación
/iboa[=0|1] ... crear un solo atributo por sección de atributos
/ibss[=0|1] ... omitir el sufijo "Attribute" en los nombres de tipo de
atributo


opciones para la generación de diagramas:
/dgen[=0|1] ... generar diagramas
/dopn[=0|1] ... abrir diagramas generados
/dsac[=0|1] ... mostrar compartimento de atributos
/dsoc[=0|1] ... mostrar compartimento de operaciones
/dscc[=0|1] ... mostrar compartimento de clasificadores anidados
/dstv[=0|1] ... mostrar valores etiquetados
/dudp[=0|1] ... usar compartimento de la propiedad .NET
/dspd[=0|1] ... mostrar compartimento de la propiedad .NET
```

```
opciones par comandos de exportación:
/ejdc[=0|1] ... comentarios de Java como JavaDocs
/ecdc[=0|1] ... comentarios de C# como DocComments
/evdc[=0|1] ... comentarios de VB como DocComments
/espl[=0|1] ... utilizar plantillas SPL definidas por el usuario
/ecod[=0|1] ... convertir código eliminado en comentario
/emrg[=0|1] ... sincronizar combinados
/egfn[=0|1] ... generar los nombres de archivo que falten
/eusc[=0|1] ... usar revisión de sintaxis


opciones para la exportación de XMI:
/exid[=0|1] ... exportar identificadores UUID
/exex[=0|1] ... exportar extensiones específicas de UModel
/exdg[=0|1] ... exportar diagramas (restablece /exex)
/exuv[=ver] ... versión de UML (UML2.0 | UML2.1.2 | UML2.2 | UML2.3)


opciones para combinar archivos:
/mcan=archivo  ... archivo antecesor común


opciones para la generación de documentación:
/doof=fmt   ... formato de salida (HTML | RTF | MSWORD | PDF)
/dsps=archivo  ... archivo de diseño SPS
```

En la sección de proyectos:

- el parámetro **`/new`** define la ruta de acceso y el nombre de archivo de proyecto nuevo (*.ump). También se puede usar para guardar un proyecto con otro nombre distinto (p. ej. `UModelBatch.exe MiArchivo.ump /new=MiArchivoBackup.ump`). Para más información consulte el apartado Archivo: Nuevo / Cargar / Opciones al guardar.
- el parámetro **`/set`** reemplaza las opciones de configuración predeterminadas del registro con las opciones definidas aquí.
- el parámetro **`/gui`** muestra la interfaz de UModel durante el procesamiento por lotes.

Ejemplo nº1

Este comando importa código fuente y crea un archivo de proyecto nuevo. Observe que la ruta de acceso del proyecto contiene espacios y, por tanto, está entre comillas.

```
"C:\Archivos de programa\Altova\UModel2017\UModelBatch.exe" /new="C:\Archivos
de programa\Altova\UModel2017\UModelBatchOut\Fred.ump" /isd="X:TestCases
\UModel\Fred" /set /gui /iclg=Java5.0 /ipsd=1 /ijdc=1 /dgen=1 /dopn=1 /
dmax=5 /chk
```

/new:	Indica que el archivo de proyecto nuevo debe llamarse "Fred.ump" y se debe guardar en `C:\Archivos de programa\Altova\UModel2017\UModelBatchOut\`
/isd=	Indica que el directorio raíz en el que se debe importar es `X:\TestCases\UModel\Fred`

/**set**:	Indica que las opciones utilizadas en la línea de comandos se guardarán en el registro (y estas opciones formarán la configuración predeterminada la próxima vez que se abra UModel).
/**gui**:	Mostrar la interfaz gráfica de UModel durante el procesamiento por lotes
/**iclg**:	UModel importará el código como Java 5.0
/**ipsd**=1:	Procesar recursivamente todos los subdirectorios del directorio raíz suministrado con el parámetro **/isd**
/**pfd**=1:	Crear un paquete por cada directorio importado en el proyecto de UModel
/**ijdc**=1:	Crear JavaDocs donde corresponda a partir de los comentarios
/**dgen**=1:	Generar diagramas
/**dopn**=1:	Abrir los diagramas generados
/**chk**:	Revisar la sintaxis

Ejemplo nº2

Este comando importa código fuente desde `X:\TestCases\UModel` y guarda el archivo de proyecto resultante en `C:\Archivos de programa\Altova\UModel2017\UModelBatchOut\finalclass.ump`.

```
"C:\Archivos de programa\Altova\UModel2017\UModelBatch.exe" /new="C:\Archivos de programa\Altova\UModel2017\UModelBatchOut\finalclass.ump" /isd="X:\TestCases\UModel\" /iclg=Java5.0 /ipsd=1 /ijdc=1 /dgen=1 /dopn=1 /dmax=5 /dsat=1 /dsnc=1 /chk
```

/**dsat**=1	Suprimir los atributos en los diagramas generados
/**dsnc**=1	Suprimir los clasificadores anidados en los diagramas generados

Ejemplo nº3

Este comando sincroniza código usando un archivo de proyecto ya existente (`"C:\Archivos de programa\Altova\UModel2017\UModelBatchOut\Fred.ump"`).

```
"C:\Archivos de programa\Altova\UModel2017\UModelBatch.exe" "C:\Program Files\Altova\UModel2017\UModelBatchOut\Fred.ump" /m2c /ejdc=1 /ecod=1 /emrg=1 /egfn=1 /eusc=1
```

/**m2c**	Actualizar el código con el modelo
/**ejdc**:	Generar JavaDoc a partir de los comentarios del modelo del proyecto
/**ecod**=1	Convertir el código eliminado en comentarios
/**emrg**=1	Sincronizar el código combinado
/**egfn**=1	Generar en el proyecto los nombres de archivo que falten
/**eusc**=1	Revisar la sintaxis

5.1 Archivo: Nuevo / Cargar / Opciones al guardar

Modo por lotes total (es decir, sin usar el parámetro **/gui**).

new

UModelBatch /new=xxx.ump (opciones)
crea un proyecto nuevo, ejecuta las opciones y xxx.ump se guarda **siempre** (independientemente de las opciones utilizadas)

auto save

UModelBatch xxx.ump (opciones)
carga el proyecto xxx.ump, ejecuta las opciones y xxx.ump **solo** se guarda si hubo cambios en el documento (como **/ibt**)

save

UModelBatch xxx.ump (opciones) /new
carga el proyecto xxx.ump, ejecuta las opciones y xxx.ump se guarda **siempre** (independientemente de las opciones utilizadas)

save as

UModelBatch xxx.ump (opciones) /new=yyy.ump
carga el proyecto xxx.ump, ejecuta las opciones y xxx.ump se guarda siempre como yyy.ump (independientemente de las opciones utilizadas)

Modo por lotes con la interfaz de UModel (es decir, usando el parámetro **/gui**).

new

UModelBatch /gui /new (opciones)
crea un proyecto nuevo, ejecuta las opciones y no se guarda nada. La interfaz gráfica permanece abierta.

save new

UModelBatch /gui /new=xxx.ump (opciones)
crea un proyecto nuevo, ejecuta las opciones y se guarda el archivo xxx.ump. La interfaz gráfica permanece abierta.

user mode

UModelBatch /gui xxx.ump (opciones)
carga el proyecto xxx.ump, ejecuta las opciones y no se guarda nada. La interfaz gráfica permanece abierta.

save

UModelBatch /gui xxx.ump (opciones) /new
carga el proyecto xxx.ump, ejecuta las opciones y se guarda el archivo xxx.ump. La interfaz gráfica permanece abierta.

save as

UModelBatch /gui xxx.ump (opciones) /new=yyy.ump
carga el proyecto xxx.ump, ejecuta las opciones y el archivo xxx.ump se guarda como yyy.ump. La interfaz gráfica permanece abierta.

El proyecto se guardará correctamente siempre y cuando no se produzcan errores graves durante la ejecución de las opciones.

Chapter 6

Proyectos e ingeniería de código

6 Proyectos e ingeniería de código

UModel admite todas las construcciones propias de Java, como por ejemplo:

- anotaciones Java
- atributos, operaciones y calificadores anidados para LiteralesDeEnumeración
- las Enumeraciones pueden realizar interfaces
- archivos de proyecto Netbeans

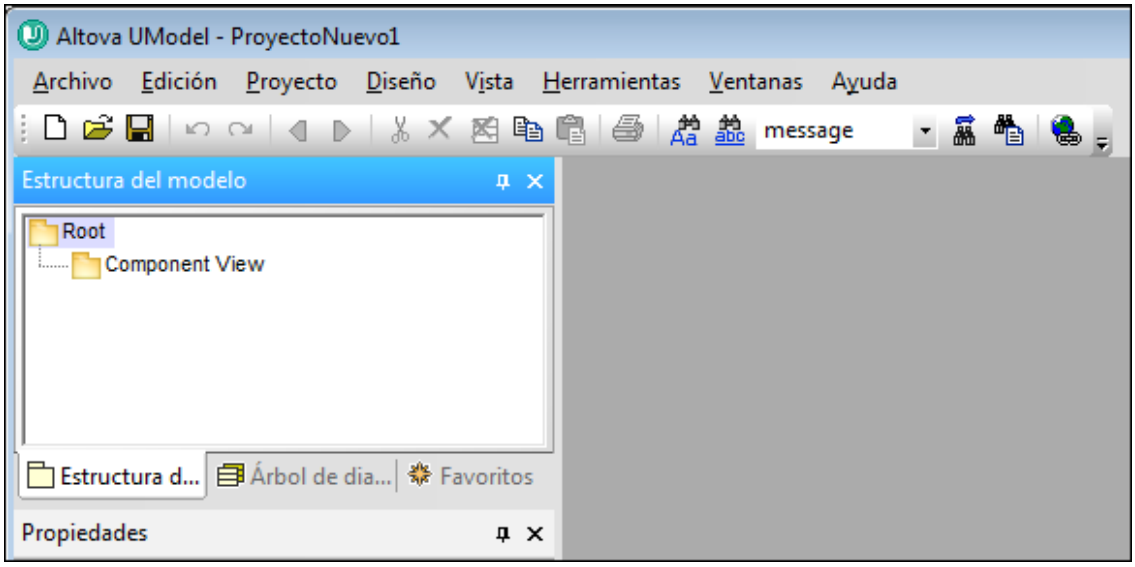
La función de ingeniería inversa ofrece:

- la posibilidad de generar un solo diagrama para todos los elementos creados con ingeniería inversa.
- la posibilidad de mostrar/ocultar elementos enlazados anónimos en diagramas.
- la posibilidad de crear hipervínculos automáticamente desde los paquetes hasta los correspondientes diagramas de contenido del paquete durante el proceso de importación.
- la posibilidad de resolver alias.
- la posibilidad de escribir caracteres Unicode en archivos de código fuente nuevos.
- la posibilidad de crear asociaciones a partir de propiedades .NET.

Para crear un proyecto nuevo:

1. Haga clic en el icono **Nuevo** de la barra de herramientas (o seleccione el comando **Archivo | Nuevo**).

Los paquetes `Root` y `Component` se insertan automáticamente cuando se crea un proyecto nuevo y están visibles en el panel Estructura del modelo. El proyecto nuevo se crea con el nombre predeterminado `ProyectoNuevo1`. Recuerde que UModel se abre con un proyecto nuevo automáticamente.



Los proyectos de UModel nuevos están formados por estos dos paquetes:

- `Root`
- `Component View`

Estos dos paquetes son los únicos paquetes que no se pueden renombrar ni eliminar.

Todos los datos sobre el proyecto se almacenan en el archivo de proyecto de UModel, que tiene la extensión `*.ump`. En la Estructura del modelo cada icono en forma de carpeta representa un paquete UML.

Proceso de trabajo con proyectos de UModel:

En primer lugar, UModel no obliga a seguir ninguna secuencia de modelado predeterminada.

Puede añadir al proyecto cualquier tipo de elemento de modelado (diagramas UML, paquetes, actores, etc.) en el orden que quiera y en la posición que quiera. No olvide que puede insertar, renombrar y eliminar todos los elementos de modelado en el panel Estructura de modelado directamente. Es decir, no hace falta crearlos dentro del diagrama.

Para insertar un paquete nuevo:

1. Haga clic con el botón derecho en el paquete donde desea que aparezca el paquete nuevo (en `Root` o en `Component View`).
2. Seleccione **Elemento nuevo | Paquete** en el menú contextual.
   Se crea un paquete nuevo dentro del paquete elegido. El campo del nombre del paquete se resalta para que pueda escribir inmediatamente el nombre del paquete.

- Los paquetes son contenedores para todos los demás elementos de modelado UML (diagramas, clases, instancias, etc.).
- Los paquetes se pueden **crear** en cualquier posición de la Estructura del modelo.
- Los paquetes (el contenido) se pueden **mover/copiar** en otros paquetes de la Estructura del modelo (y en diagramas válidos que estén abiertos en el área de trabajo).
- Los paquetes y su contenido se pueden **ordenar** (en la Estructura del modelo) en función de varios criterios.
- Los paquetes se pueden **colocar** dentro de otros paquetes.
- Los paquetes se pueden usar como elementos de **origen** o **destino** cuando se combina o sincroniza el código.

Para que los elementos aparezcan en un diagrama UML:

1. Inserte un diagrama UML nuevo, haciendo clic con el botón derecho y seleccionando **Diagrama nuevo | Diagrama de clases**.
2. Arrastre un elemento de modelado desde la Estructura del modelo hasta el diagrama que acaba de crear.
3. Utilice el menú contextual del área de trabajo del diagrama para añadir elementos nuevos en el diagrama.

Para guardar un proyecto:

Seleccione la opción de menú **Archivo | Guardar como...** (o **Archivo | Guardar**).

Nota: en la pestaña *Archivo* del cuadro de diálogo "Opciones locales" (**Herramientas | Opciones**) puede configurar la aplicación para que los archivos `*.ump` se guarden en formato pretty-print.

Para abrir un proyecto:

Seleccione la opción de menú **Archivo | Abrir** o seleccione un archivo de la lista de archivos.

Nota: los cambios realizados en el archivo de proyecto o en sus archivos incluidos desde una aplicación externa se registran automáticamente y generan un aviso. Puede elegir si el proyecto se carga otra vez o no.

Para mover un proyecto:

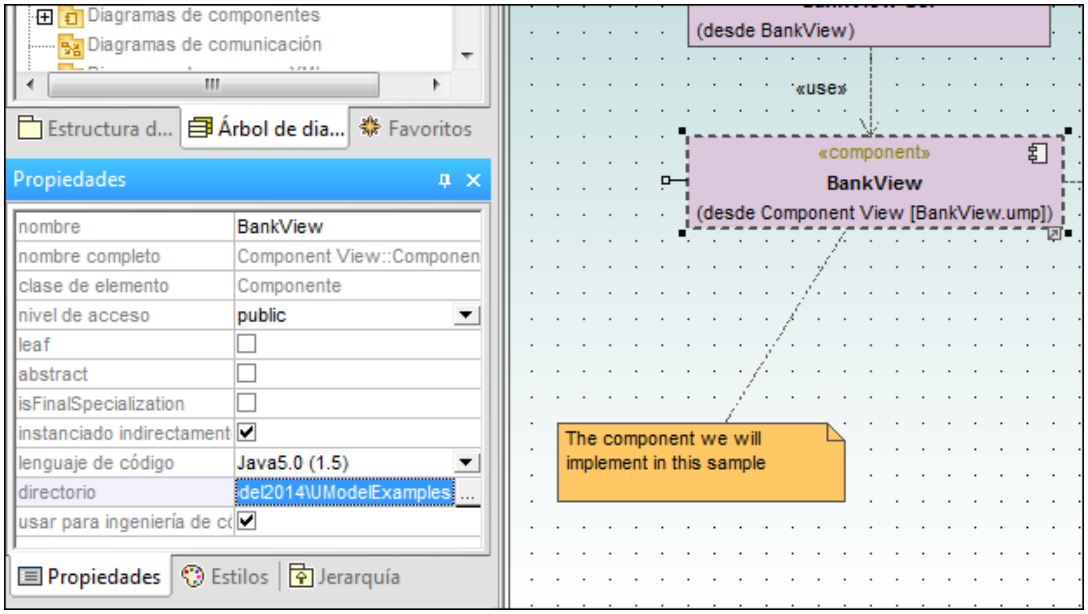
Los proyectos de UModel y el código generado se pueden mover a otro directorio (o a otro equipo) y volver a sincronizarse en el directorio (o equipo) de destino.

Esto se puede hacer de dos maneras:

- Con el comando **Archivo | Guardar como...** y haciendo clic en **Sí** cuando la aplicación solicita adaptar las rutas de acceso del archivo a la nueva ubicación del proyecto.
- Copiando el proyecto de UModel (`*.ump`) a una ubicación nueva y adaptando las rutas de acceso de los paquetes de generación de código de **`Component View`** a la ubicación nueva.

Por ejemplo, tomemos el archivo `BankMultilanguage.ump`:

1. Abra el diagrama de componentes **`Overview`** y del paquete **`Design View`**. Ahora seleccione el componente **`BankView`** del diagrama.
2. En el panel Propiedades actualice la ruta de acceso del campo `directorio` con la nueva ubicación del proyecto.
3. Sincronice el modelo y el código.



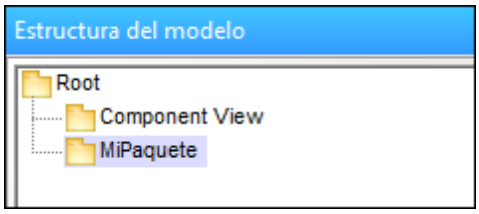
6.1 Crear un proyecto de UModel desde cero

Este apartado describe los pasos necesarios para crear un proyecto desde cero y generar código para una sola clase. Aunque es un proyecto muy básico, utiliza varios diagramas para explicar cómo agregar métodos, etc. Todas las operaciones de los ejemplos se pueden conseguir desde el panel Estructura del modelo directamente, sin necesidad de usar diagramas de modelado para crear clases ni métodos.

Crear un proyecto nuevo y definir una raíz de espacio de nombres

En el panel Estructura del modelo:

1. Seleccione **Archivo | Nuevo** para crear un proyecto de modelado nuevo.
2. Haga clic con el botón derecho en el paquete `Root`. Seleccione **Elemento nuevo | Paquete** y cree un paquete nuevo llamado `MiPaquete`.



3. Haga clic con el botón derecho en `MiPaquete` y seleccione **Ingeniería de código | Establecer como raíz de espacio de nombres Java**. Aparece un aviso. Haga clic en **Sí** para aplicar el perfil Java de UModel al paquete. El paquete `Java Profile` se añade a la Estructura del modelo.

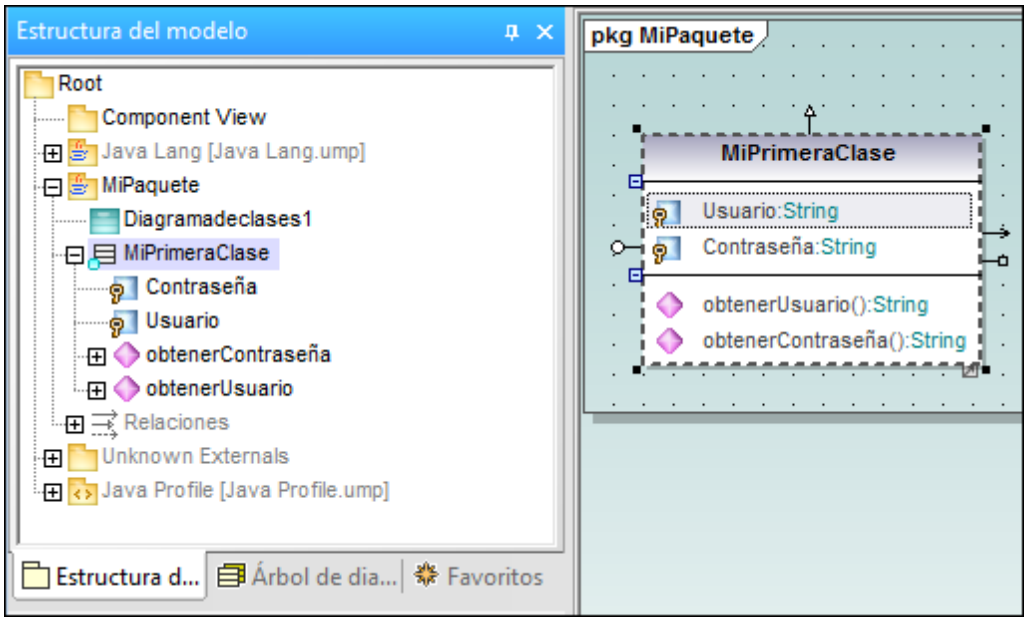
Incluir Java Lang para aportar tipos de datos JDK

1. Haga clic en el paquete `Root` y seleccione el comando de menú **Proyecto | Incluir un subproyecto**.
2. En el cuadro de diálogo que aparece seleccione la pestaña *Java 1.4* y después el paquete `Java Lang.ump`. En el siguiente cuadro de diálogo haga clic en **Aceptar** para usar la opción predeterminada *Incluir mediante referencia*.

Los paquetes **`Java Lang`** y **`Unknown Externals`** se añaden a la Estructura del modelo (*imagen anterior*).

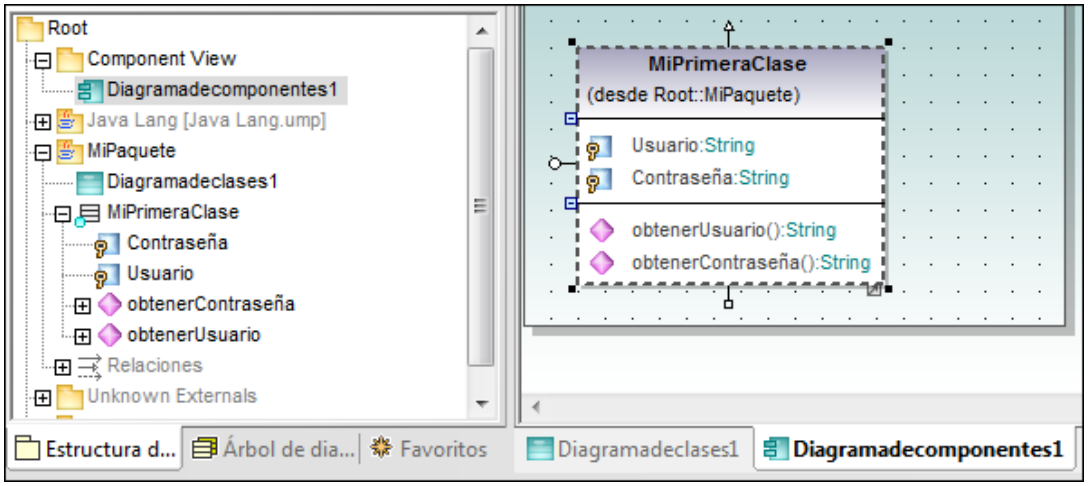
Crear las propiedades y los métodos de las clases

1. Haga clic con el botón derecho en **`MiPaquete`** y seleccione **Diagrama nuevo | Diagrama de clases** en el menú contextual.
2. Haga clic con el botón derecho en el diagrama de clases y seleccione **Elemento nuevo | Clase** para crear una clase nueva en el diagrama (p. ej. **`MiClase`**).
3. Pulse **F7** y añada algunos atributos (p. ej. `Usuario:String` y `Contraseña:String`).
4. Pulse **F8** y añada algunas operaciones (p. ej. `obtenerUsuario():String` y `obtenerContraseña():String`).

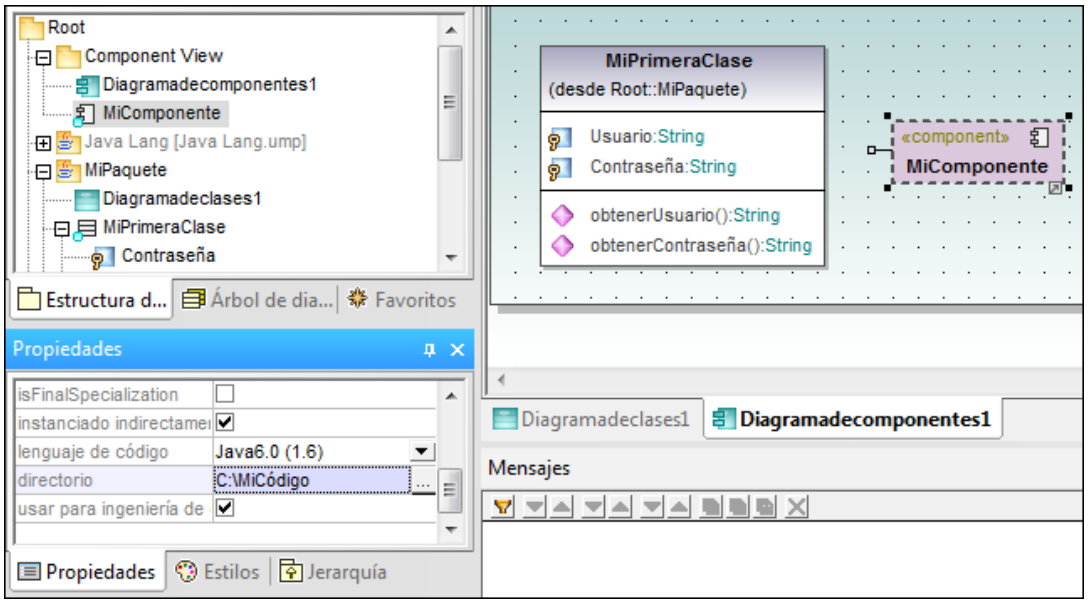


Crear un componente y definir el directorio del código

1. Haga clic con el botón derecho en el paquete **`Component View`** y añada un diagrama de componentes nuevo.
2. Arrastre la clase **`MiPrimeraClase`** desde la Estructura del modelo hasta el diagrama de componentes.

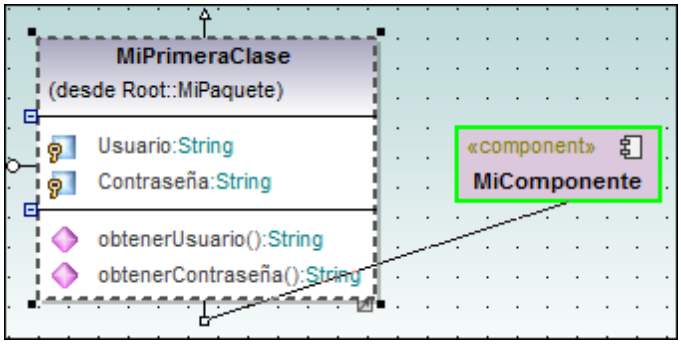


3. Añada un componente nuevo al diagrama (p. ej. **`MiComponente`**).
4. Haga clic en el componente y después en el campo directorio del panel Propiedades. Ahora inserte el directorio donde desea guardar el código (p. ej. `c:\MiCódigo`).



Realizar la clase

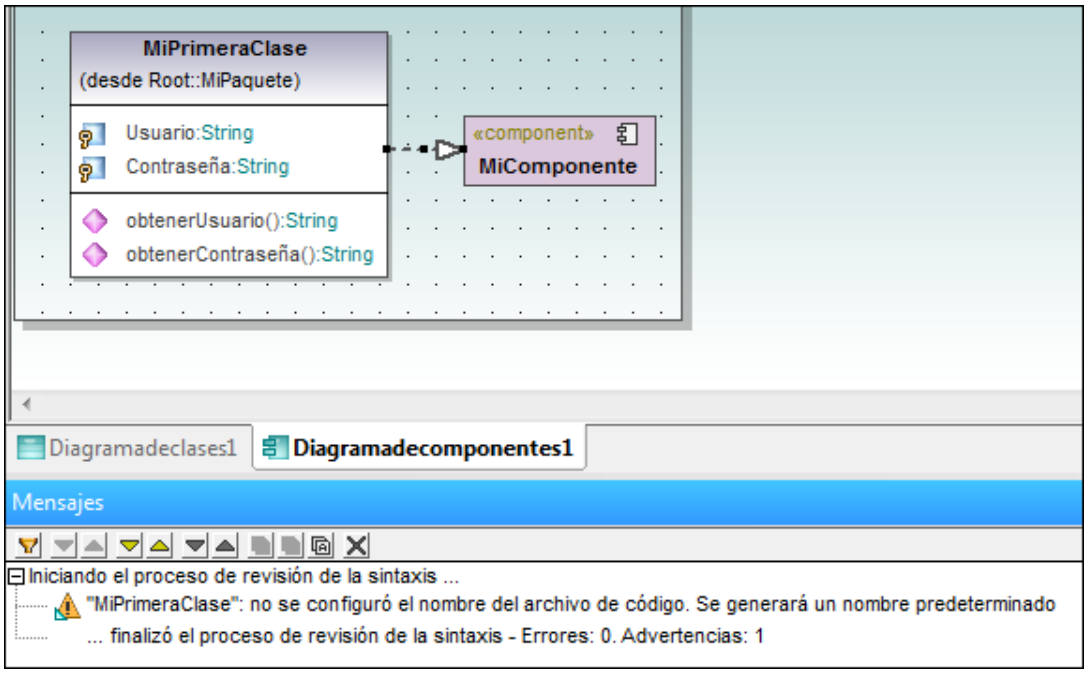
1. Haga clic en **`MiClase`** y arrastre el controlador **`RealizaciónDeComponente`** (situado en la parte inferior del recuadro) hasta el componente **`MiComponente`**.



Para poder generar código es necesario realizar las clases. No olvide que también puede arrastrar la clase y colocarla en el componente desde la Estructura del modelo directamente.

Revisar la sintaxis y generar código

1. Seleccione el comando de menú **Proyecto | Revisar la sintaxis del proyecto** para comprobar que todo es correcto.



No se genera ningún error pero hay un mensaje de advertencia. Aunque no definió un nombre para el archivo de código, UModel generará un nombre predeterminado automáticamente.

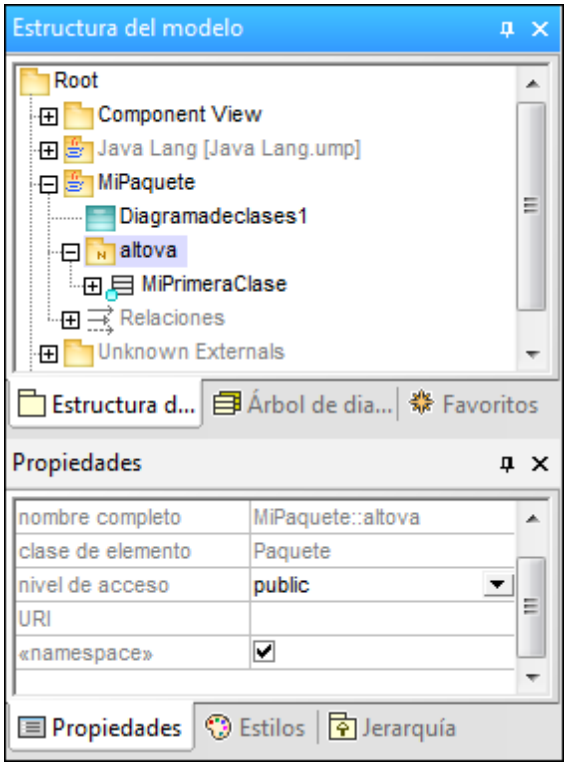
2. Seleccione el comando de menú **Proyecto | Combinar el código de programa con el proyecto de UModel...** para generar el código Java.

Crear un espacio de nombres

Si quiere generar la **clase** dentro de un espacio de nombres determinado:

1. Añada un paquete nuevo dentro de `MiPaquete` (p. ej. `altova`).
2. Haga clic en el paquete `altova` y marque la casilla <<**namespace**>> del panel

Propiedades.



3. Ahora, en la Estructura del modelo, arrastre la clase **`MiPrimeraClase`** hasta el paquete **`altova`**.
   Cuando se genere el código, la clase estará en el espacio de nombres **`altova`**.

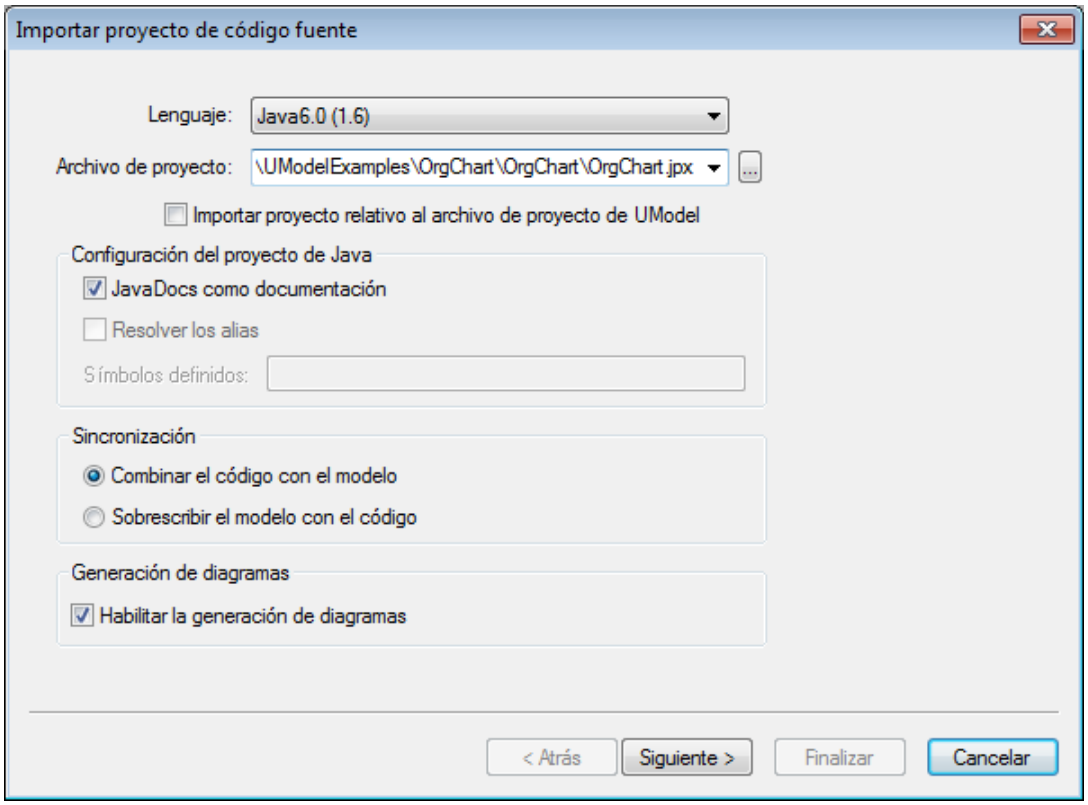
6.2 Importar código fuente a los proyectos

Puede importar código fuente como proyecto de código fuente o como directorio de código fuente. Para ver un ejemplo de cómo importar código como **directorio de código fuente**, consulte el apartado Ingeniería de ida y vuelta (código - modelo - código) del tutorial.

- Puede importar archivos JBuilder `.jpx`, archivos de proyecto Eclipse `.project` y archivos NetBeans (`proyecto.xml`).
- También puede importar proyectos en C# / Visual Basic (archivos de proyecto Visual Studio `sln, csproj, csdprj..., vbproj, vbp` y Borland `.bdsproj`)

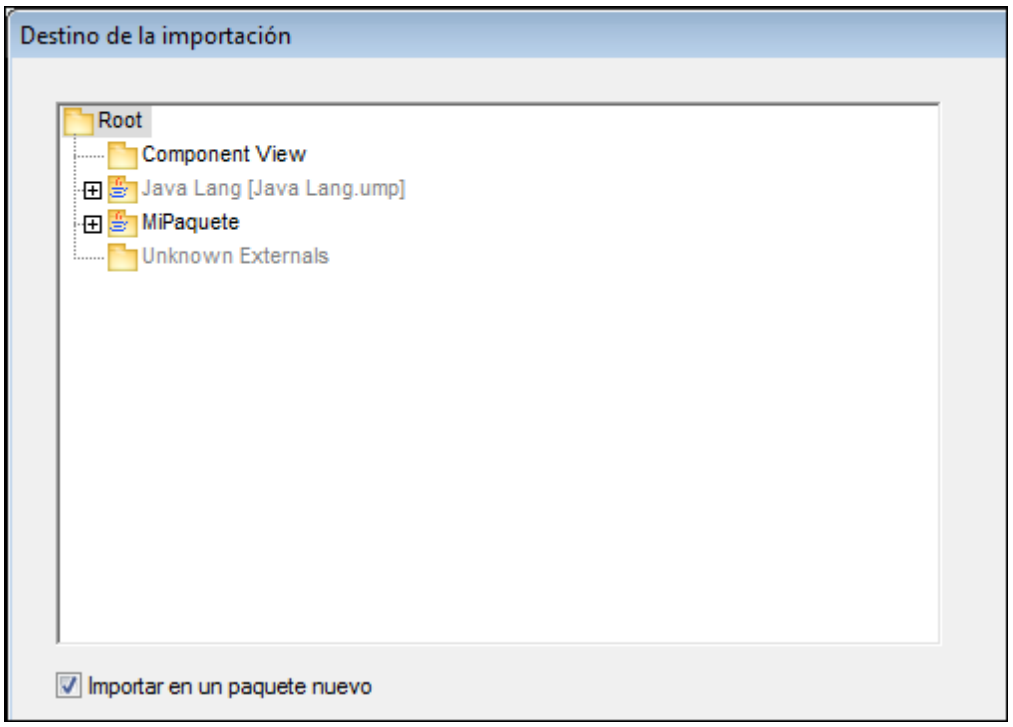
Para importar a UModel un proyecto ya existente:

1. Seleccione el comando de menú **Proyecto | Importar proyecto de código fuente**.
2. En el cuadro de diálogo haga clic en el botón **Examinar** ... para seleccionar el archivo de proyecto.

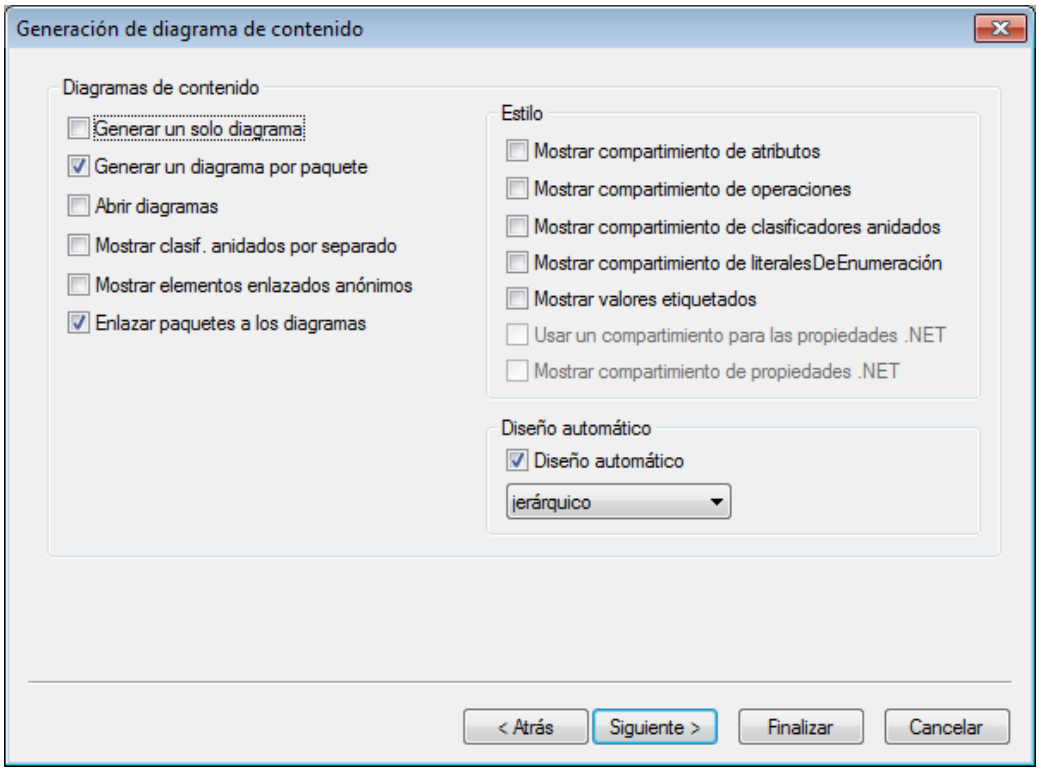


3. Seleccione el tipo de archivo de proyecto (p. ej. `.jpx`) y haga clic en **Abrir**. Este archivo de proyecto JBuilder está disponible en el archivo `OrgChart.zip` de la carpeta `...\UModelExamples`. No olvide que está activa la opción para importar el proyecto relativo al archivo de proyecto de UModel. Para que UModel pueda generar diagramas de clases y diagramas de paquetes a partir del código fuente, marque la casilla *Habilitar la generación de diagramas* y haga clic en **Siguiente**.

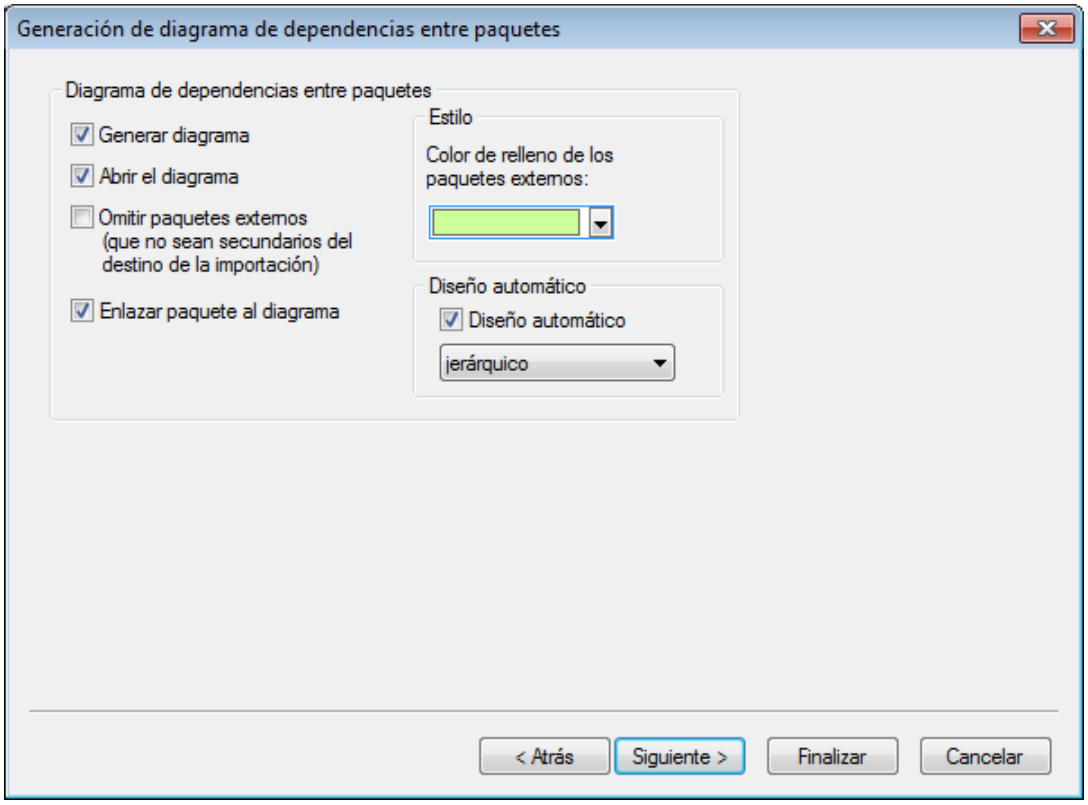
4. En la siguiente pantalla, marque la casilla *Importar en un paquete nuevo* (o haga clic en el paquete donde desea importar el código).



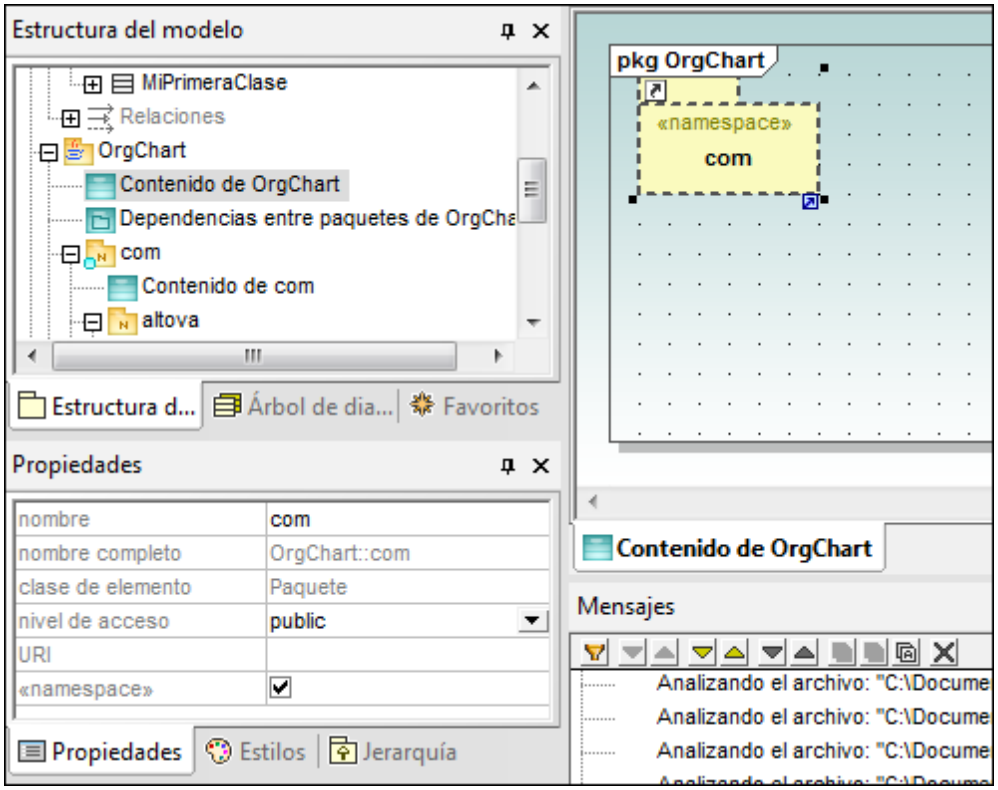
5. No olvide que UModel puede generar un solo diagrama global o un diagrama por cada paquete. Las opciones de configuración que aparecen a continuación son las opciones predeterminadas.



6. Haga clic en **Siguiente** para continuar. En la siguiente pantalla puede configurar la generación del diagrama de dependencias entre paquetes.



7. Haga clic en **Finalizar** para usar la configuración predeterminada. El proyecto se analiza y el modelo de UModel se genera.



Nota: si importa el código en un proyecto ya disponible, UModel le pregunta en qué paquete se debe importar. Si usa un proyecto nuevo, se crea automáticamente la carpeta **`OrgChart`**.

Resolver alias:
Cuando aplique ingeniería inversa a código que contiene alias de espacio de nombres o alias de clase, podrá elegir si el alias se resuelve o no (activando/desactivando la casilla *Resolver alias* del cuadro de diálogo "Importar proyecto/directorio de código fuente").

Cuando actualice el código (con el modelo), las definiciones de alias se retienen en el código tal y como están. El ámbito de las definiciones de alias son los archivos en los que aparecen.

Por ejemplo:

```
using Q = System.Collections.Generic.Queue<String>;
Q myQueue;
```

Los alias que puedan estar en conflicto porque su uso sea ambiguo se añaden a **`Unknown Externals`**.

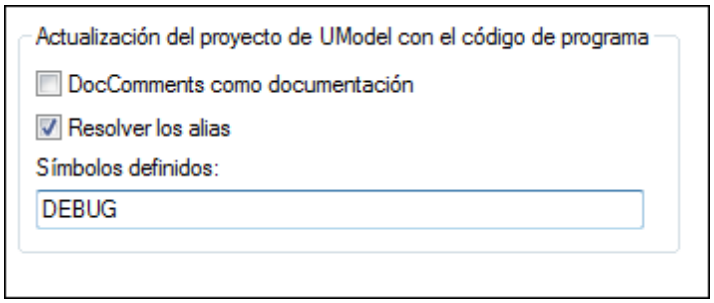
Nota: la configuración de la opción *Resolver alias* puede cambiarse en cualquier momento en el cuadro de diálogo "Configuración del proyecto".

Símbolos definidos

El código C# o Visual Basic permite insertar una lista de símbolos definidos en el campo *Símbolos definidos* del cuadro de diálogo "Configuración del proyecto". Esta lista se usa para compilar secciones del código de forma condicional. En la lista los símbolos deben ir separados por punto y coma. UModel tiene en cuenta los símbolos definidos durante el proceso de

ingeniería de código.

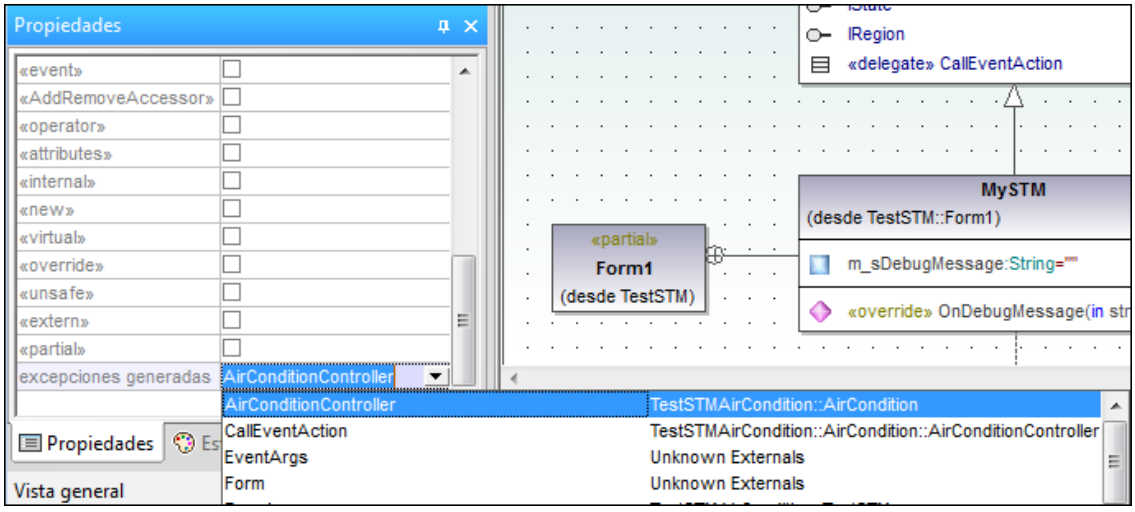
Una vez finalizado el proceso de ingeniería inversa, UModel da salida a todos los símbolos utilizados en el código fuente en la ventana Mensajes.



Por ejemplo: #If DEBUG Then
Dim i = 10
#Else
dim a = 20
#End If

Excepciones generadas

Si hace clic en una operación de una clase y después hace clic en el cuadro combinado `excepciones generadas` (panel Propiedades), podrá ver qué información ofrece la excepción que se genera para una operación.



6.3 Importar binarios Java, C# y VB

UModel permite la importación de binarios C#, Java y VB, lo cual es de gran utilidad cuando se trabaja con binarios de terceros o cuando el código fuente original ya no está disponible.

Si tiene pensado importar archivos binarios Java, es necesario tener instalado Sun Java Runtime Environment (JRE) o el kit de desarrollo JDK en las versiones 1.4, 1.5, 1.6, 1.7 o 1.8. La importación de tipos es compatible con todos los archivos de clases que apunten a estos entornos (es decir, que cumplan la especificación de Java Virtual Machine).

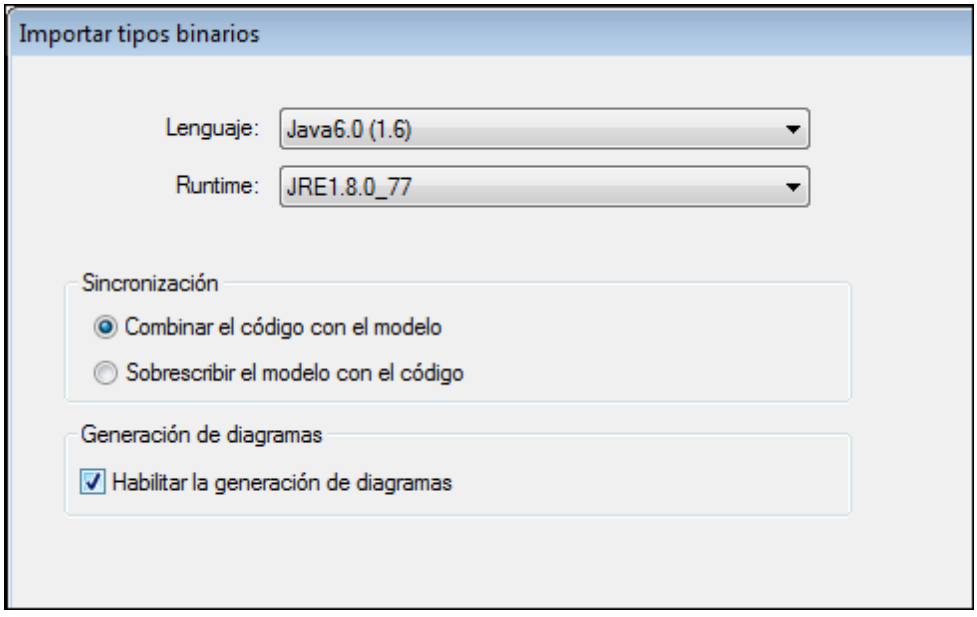
Si tiene pensado importar archivos binarios C# o VB, es necesario tener instalado .NET Framework. La importación de tipos es compatible con todos los ensamblados que apunten a .NET Framework 1.1, 2.0, 3.0, 3.5, 4.0, 4.5, 4.6 y .NET Compact Framework v1.0, v2.0, v3.5 (para PocketPC, Smartphone, WindowsCE).

Estos requisitos solamente son relevantes si tiene pensado importar binarios Java, C# o VB. NET. Si no es así, no hace falta tener instalado Java Runtime Environment ni .NET Framework.

No es posible importar binarios confusos.

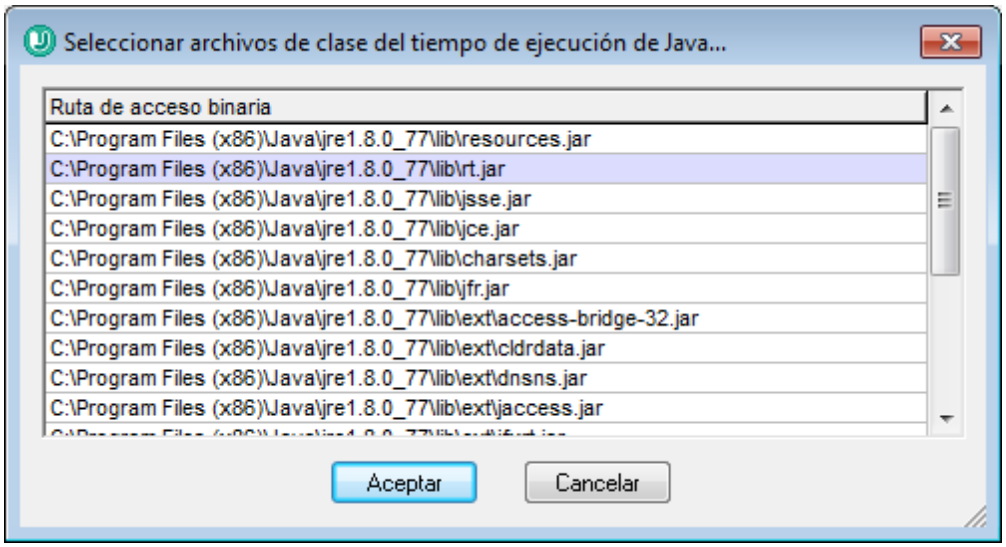
Para importar archivos binarios:

1. Seleccione el comando de menú **Proyecto | Importar tipos binarios**.

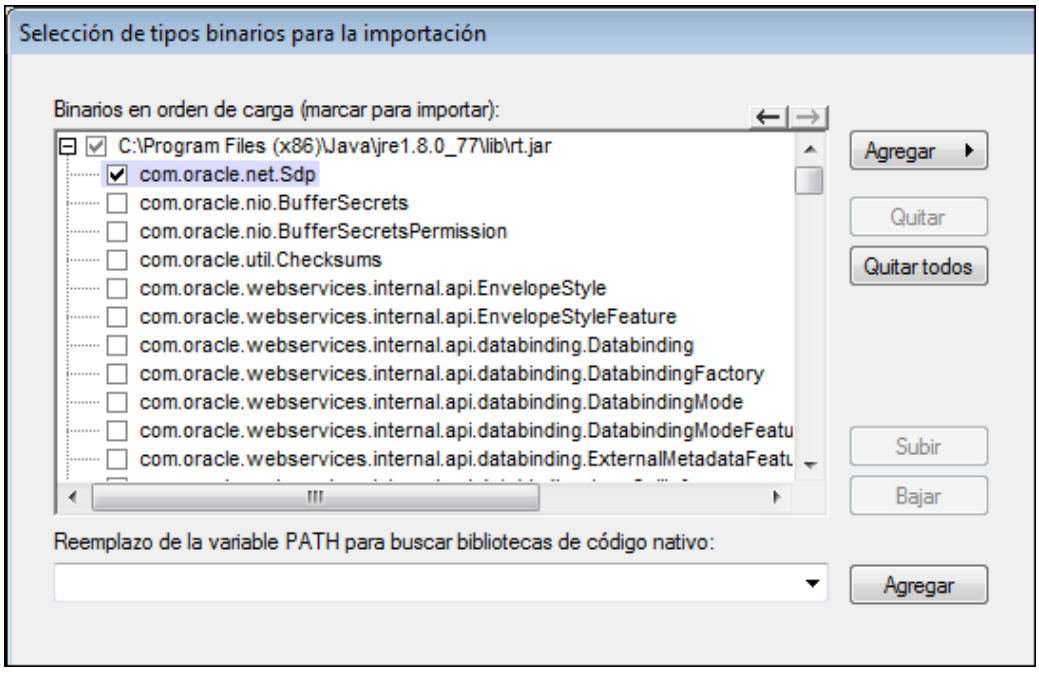


2. Seleccione el lenguaje y la edición de Runtime.
3. Para que UModel pueda generar diagramas de clases y diagramas de paquetes a partir del código fuente, marque la casilla *Habilitar la generación de diagramas* y haga clic en **Siguiente**.
4. Ahora aparece el cuadro de diálogo "Selección de tipos binarios para la importación".
5. Haga clic en **Agregar** y seleccione el archivo de clases en el menú contextual (en este ejemplo añadimos **Archivos de clases de Java Runtime**).

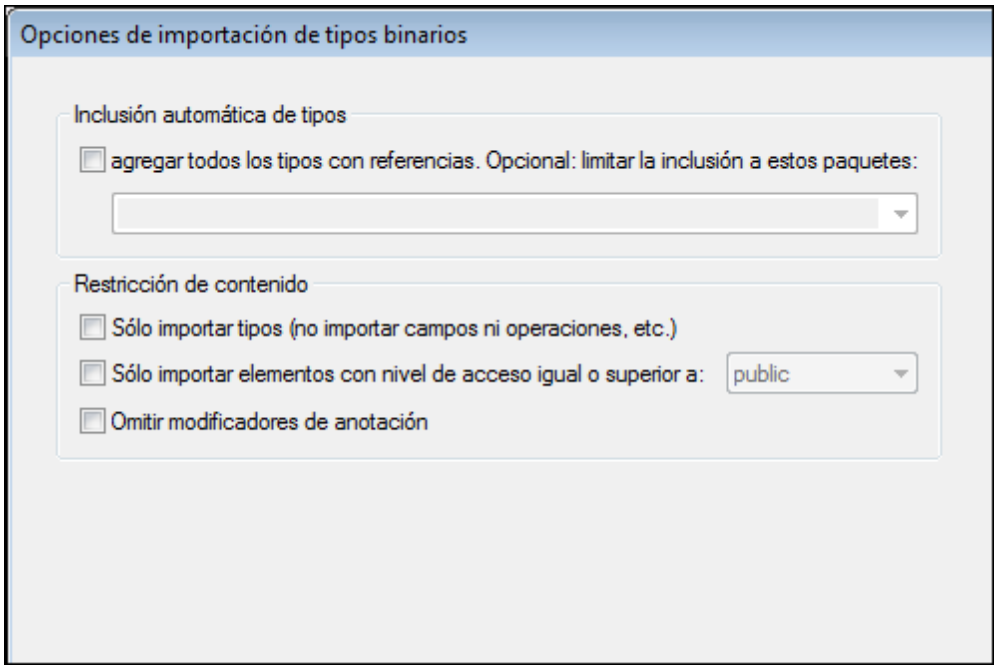
6. Seleccione un binario (en este ejemplo utilizamos `rt.jar`).



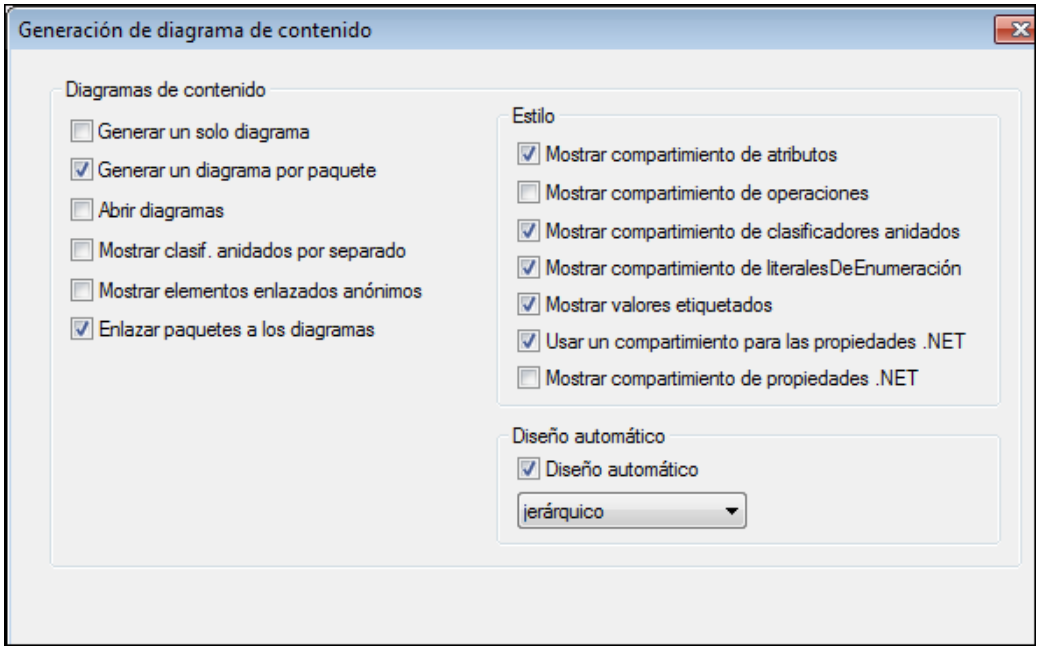
7. Ahora haga clic en el botón + para expandir la lista de binarios y marque las casillas de los binarios que desea importar.



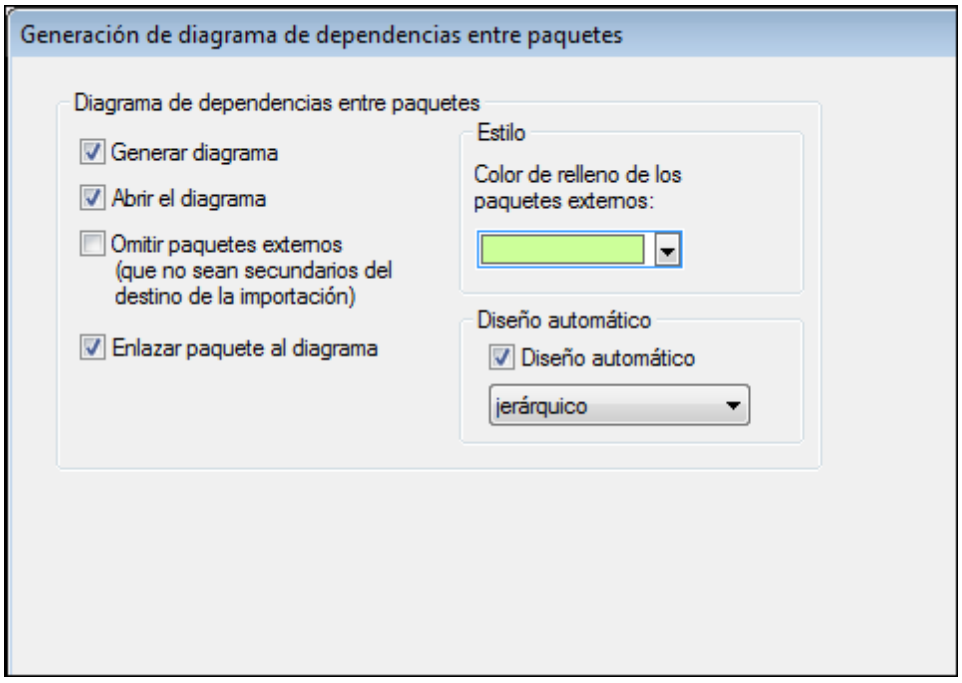
8. Haga clic en **Siguiente**. Esto abre el cuadro de diálogo "Opciones de importación de tipos binarios".



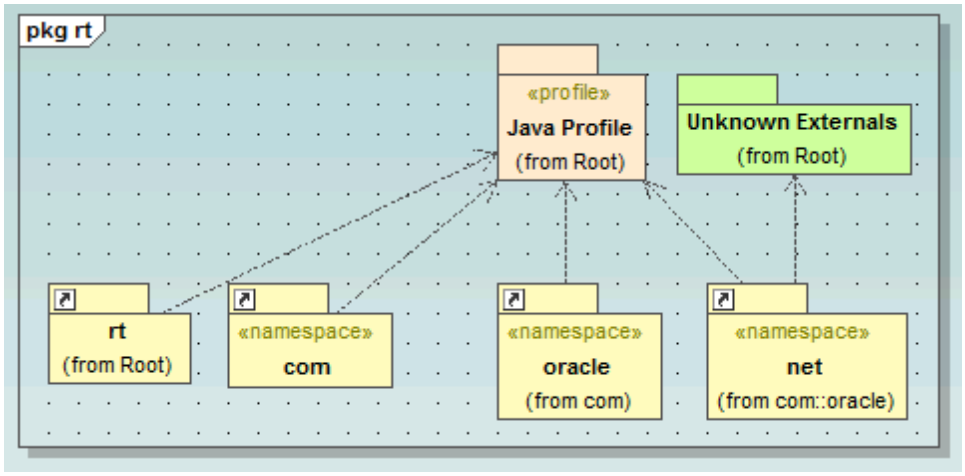
9. Haga clic en **Siguiente**.
10. Si es necesario, defina el destino de la importación o marque la casilla *Importar en un paquete nuevo* y después haga clic en **Siguiente**.



11. Seleccione las propiedades de generación del diagrama de contenido y haga clic en **Siguiente** para continuar. Puede generar un diagrama por cada paquete o un solo diagrama.



12. Seleccione las opciones de dependencias entre paquetes que desea incluir y haga clic en **Finalizar** para terminar el proceso de importación. En la imagen siguiente puede ver el diagrama que contiene las dependencias entre paquetes de los binarios Java.



Nota: al hacer clic en el icono en forma de flecha de una carpeta aparece automáticamente el diagrama al que se hace referencia.

6.4 Sincronizar el modelo y el código fuente

UModel ofrece una función para sincronizar el modelo con el código y viceversa. El código se puede combinar y sincronizar por niveles, tal y como se describe a continuación (a nivel de proyecto, de paquetes o de clases).

> Cuando UModel (Enterprise o Professional) se ejecuta como complemento para Eclipse o Visual Studio, la sincronización entre código y modelo tiene lugar automáticamente. La sincronización manual solamente se puede hacer a nivel de proyecto. La opción para actualizar clases y paquetes por separado no está disponible.

Al hacer clic con el botón derecho dentro del árbol de diagramas (en una clase, por ejemplo), el menú contextual ofrece comandos de combinación o sincronización de código en un submenú llamado **Ingeniería de código**:

- **Combinar código de programa con *** de UModel...**
- **Combinar *** de UModel con el código de programa...**

> *** es en este caso un proyecto, un paquete, un componente, una clase, etc., dependiendo de qué tipo de elemento se seleccionara en el árbol de diagramas.

Dependiendo de las opciones definidas en **Proyecto | Configurar sincronización, estos comandos también se pueden llamar así**:

- **Sobrescribir código de programa con *** de UModel...**
- **Sobrescribir *** de UModel con el código de programa...**

Para actualizar todo el proyecto (pero no las clases, los paquetes ni demás elementos locales), puede utilizar estos comandos del menú **Proyecto**:

- **Combinar (o sobrescribir) el código de programa con el proyecto de UModel**
- **Combinar (o sobrescribir) el proyecto de UModel con el código de programa**

En adelante nos referiremos a estos comandos como *comandos de sincronización de código*.

Para sincronizar el código a nivel de proyecto o paquete raíz tiene dos opciones:

- Haga clic con el botón derecho en el paquete `Raíz` del Árbol de diagramas y seleccione el comando de sincronización de código correspondiente.
- En el menú Proyecto haga clic en el comando de sincronización de código correspondiente.

Para sincronizar el código a nivel de paquete:

1. Mantenga pulsado **Mayús** o **Ctrl** mientras hace clic en los paquetes que desea

sincronizar.
2. Haga clic con el botón derecho en la selección y elija el comando de sincronización de código correspondiente.

Para sincronizar el código a nivel de clase:

1. Mantenga pulsado **Mayús** o **Ctrl** mientras hace clic en las clases que desea sincronizar.
2. Haga clic con el botón derecho en la selección y elija el comando de sincronización de código correspondiente.

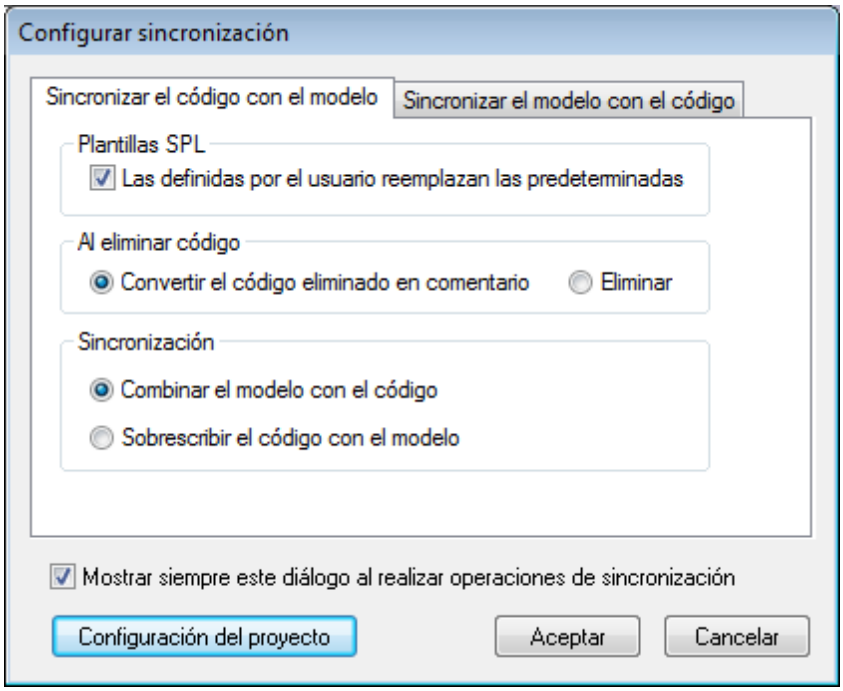
A fin de evitar resultados no deseados a la hora de sincronizar modelo y código, debe tener en cuenta todas estas posibilidades:

En el menú **Proyecto**, si hace clic en **Sobrescribir proyecto de UModel con el código de programa**.	• Esto busca en todos los directorios (archivos de proyecto) de todos los lenguajes de código diferentes que están definidos en el proyecto. • En la ventana Mensajes aparece la entrada `Recopilando archivos fuente en....`
Si hace clic con el botón derecho en una clase o interfaz en el panel "Estructura del modelo" y selecciona **Ingeniería de código | Sobrescribir clase de UModel con el código de programa**.	• Esto actualiza solamente la clase seleccionada (interfaz) del proyecto. • Sin embargo, si el código fuente contiene clases que son nuevas o clases que se modificaron después de la última sincronización, estos cambios no se añadirán al modelo.
Si hace clic con el botón derecho en un componente en el panel "Estructura del modelo" (dentro del paquete `Component View`) y selecciona y selecciona **Ingeniería de código | Sobrescribir componente de UModel con el código de programa**.	• Esto actualiza el directorio correspondiente (o archivo de proyecto) solamente. • Se identifican los archivos nuevos del directorio (archivo de proyecto) y se añaden al proyecto. • En la ventana Mensajes aparece la entrada `Recopilando archivos fuente en....`

Nota: durante la sincronización de código puede recibir un mensaje solicitando que actualice el proyecto de UModel antes de iniciar la sincronización. Esto ocurre cuando se abren proyectos de UModel creados con una versión anterior a la versión más reciente de UModel. Haga clic en **Sí** para actualizar el proyecto y guardarlo en el formato más reciente. El mensaje de notificación ya no aparecerá más.

Configurar sincronización

Las opciones de configuración se pueden modificar con el comando de menú **Proyecto | Configurar sincronización**.



Plantillas SPL
Las plantillas SPL se utilizan durante la generación de código Java, C# y VB.NET, pero solamente se utilizan si se genera código nuevo, es decir, si se añaden nuevas clases, operaciones, etc. al modelo. El código ya existente no accede a las plantillas SPL ni las utiliza.

Para modificar las plantillas SPL que vienen con UModel:

1. Primero debe localizar las plantillas SPL que vienen con UModel. El directorio donde están guardadas es: `...\UModel2017\UModelSPL\Java\Default` (o `...\C#\Default`, `...\VB\Default`.)
2. Copie los archivos SPL que desea editar/modificar en el directorio primario (`...\UModel2017\UModelSPL\Java\`).
3. Realice los cambios y guárdelos.

Para usar las plantillas SPL personalizadas:

1. Seleccione el comando de menú **Proyecto | Configurar sincronización**.
2. Marque la casilla *Las definidas por el usuario reemplazan las predeterminadas*.

6.4.1 Consejos prácticos

Cambiar el nombre de los clasificadores y aplicar ingeniería inversa:
El proceso descrito más abajo tiene lugar durante la ingeniería inversa y la sincronización automática, tanto en la versión independiente de UModel como en los complementos de UModel para Visual Studio y Eclipse.

Si cambia el nombre de un clasificador en la aplicación de programación, el clasificador se elimina o se vuelve a insertar en el panel Estructura del modelo de UModel como clasificador nuevo.

El clasificador nuevo solo se vuelve a insertar en los diagramas de modelado que se crean automáticamente durante el proceso de ingeniería inversa o cuando se genera un diagrama con el comando **Mostrar en un diagrama nuevo de | Contenido**. El clasificador nuevo se inserta en una posición predeterminada en el diagrama, que probablemente no coincida con su ubicación anterior.

Para más información consulte el apartado Refactorizar código y sincronización.

Generación automática de RealizacionesDeComponente

UModel puede generar RealizacionesDeComponente automáticamente durante el proceso de ingeniería de código. Las RealizacionesDeComponente solo se generan cuando está totalmente claro a qué componente se debe asignar una clase, es decir:

- Cuando solo existe un archivo de proyecto de Visual Studio en el archivo `*.ump`.
- Cuando existen varios proyectos de Visual Studio pero sus clases están totalmente separadas en el modelo.

Para habilitar la generación automática de RealizacionesDeComponente:

1. Seleccione el comando **Herramientas | Opciones**.
2. Haga clic en la pestaña *Ingeniería de código* y marque la casilla *Generar las realizacionesDeComponente que falten*.

Las RealizacionesDeComponente automáticas se crean para un clasificador al que se le puede asignar un solo componente. Es decir:

- un clasificador que no tiene ninguna RealizaciónDeComponente o
- un clasificador que está dentro del espacio de nombres de un lenguaje de código

Hay varias maneras de buscar componentes, dependiendo del tipo de componente.

Si se trata de componentes que representan un archivo de proyecto de código (cuando tiene definida la propiedad `projectfile`):

- se busca si hay UN componente que tiene/realiza clasificadores en el paquete que lo contiene.
- se busca si hay UN componente que tiene/realiza clasificadores en un subpaquete del paquete que lo contiene (de arriba a abajo).
- se busca si hay UN componente que tiene/realiza clasificadores en uno de los paquetes primarios (de abajo a arriba).
- se busca si hay UN componente que tiene realiza clasificadores en un subpaquete de uno de los paquetes primarios (de arriba a abajo).

Si se trata de componentes que representan un directorio (cuando tiene definida la propiedad `directory`):

- se busca si hay UN componente que tiene/realiza clasificadores en el paquete que lo contiene
- se busca si hay UN componente que tiene/realiza clasificadores en uno de los paquetes

primarios (de abajo a arriba)

Notas:

- Es necesario activar la opción *Generar las realizacionesDeComponente que falten* (de la pestaña *Ingeniería de código* del cuadro de diálogo "Opciones locales").
- En cuanto UModel encuentra UN componente viable durante los pasos descritos más arriba, el componente se utiliza y se omiten los pasos siguientes.

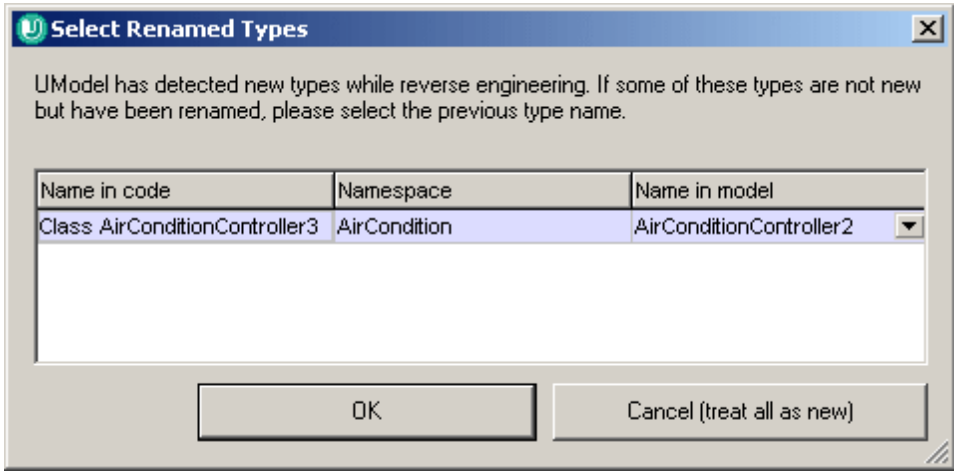
Errores/advertencias:

- Si no se encuentra ningún componente viable, se genera una advertencia.
- Si se encuentran varios componentes viables, se genera un error.

6.4.2 Refactorización de código y sincronización

Cuando se refactoriza código, el nombre de las clases suele cambiar o actualizarse. Las versiones de UModel anteriores a la versión 2009 eliminaban las clases antiguas e insertaban clases nuevas durante el proceso de sincronización del código/modelo.

En UModel 2009 o superior, si se detecta que durante la fase de ingeniería inversa se añadieron tipos nuevos o se cambió el nombre de algunos tipos, aparece el cuadro de diálogo "Seleccionar tipos con nombre nuevo" (*imagen siguiente*). La columna *Nombre en el código* enumera los tipos nuevos, mientras que el nombre original de cada tipo aparece en la columna *Nombre en el modelo*. UModel trata de averiguar cuál era el nombre original del tipo a partir del espacio de nombres, el contenido de la clase, las clases bases y otros datos.



Si se cambió el nombre de una clase, seleccione el nombre antiguo de la clase en la lista desplegable de la columna *Nombre en el modelo* (p. ej. `C1`). Esto permite conservar todos los datos relacionados y que el proceso de ingeniería de código funcione con precisión.

Cambiar el nombre de las clases en el modelo y volver a generar código

Tras crear un modelo y generar código a partir de él, si quiere puede volver a realizar cambios en el modelo antes de iniciar el proceso de sincronización.

Por ejemplo, imagine que quiere cambiar el nombre de las clases antes de generar código por segunda vez. Como previamente asignó un nombre de archivo a cada clase, en el campo `nombre del archivo de código`, la clase nueva y el nombre de archivo no coinciden.

Cuando inicie el proceso de sincronización, UModel le pregunta si quiere que el nombre del archivo de código coincida con el nombre de la clase nueva. Recuerde que también tiene la opción de cambiar los constructores de clase.

Ingeniería de ida y vuelta y relaciones entre elementos de modelado

Cuando se actualiza el modelo con el código, las asociaciones entre elementos de modelado aparecen en pantalla automáticamente si se marcó la opción *Crear asociaciones automáticamente* en la pestaña *Edición de diagramas* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**). UModel muestra las asociaciones de los elementos que tengan configurado el tipo de los atributos y en cuyo mismo diagrama esté el elemento de modelado **`type`**.

Las realizacionesDeInterfaz y las generalizaciones aparecen automáticamente en el diagrama cuando se actualiza el modelo con el código.

6.5 Requisitos para ingeniería directa

Requisitos para generar código para ingeniería directa:

- Un componente es **realizado** por una **clase** o varias o por una **interfaz** o varias.
- El componente debe tener asignada una ubicación física (es decir, un directorio). El código generado se coloca después en este directorio.
- Cada componente debe estar configurado para incluirse en el proceso de generación de código.
- El paquete raíz de espacio de nombres Java, C# o VB debe estar definido.

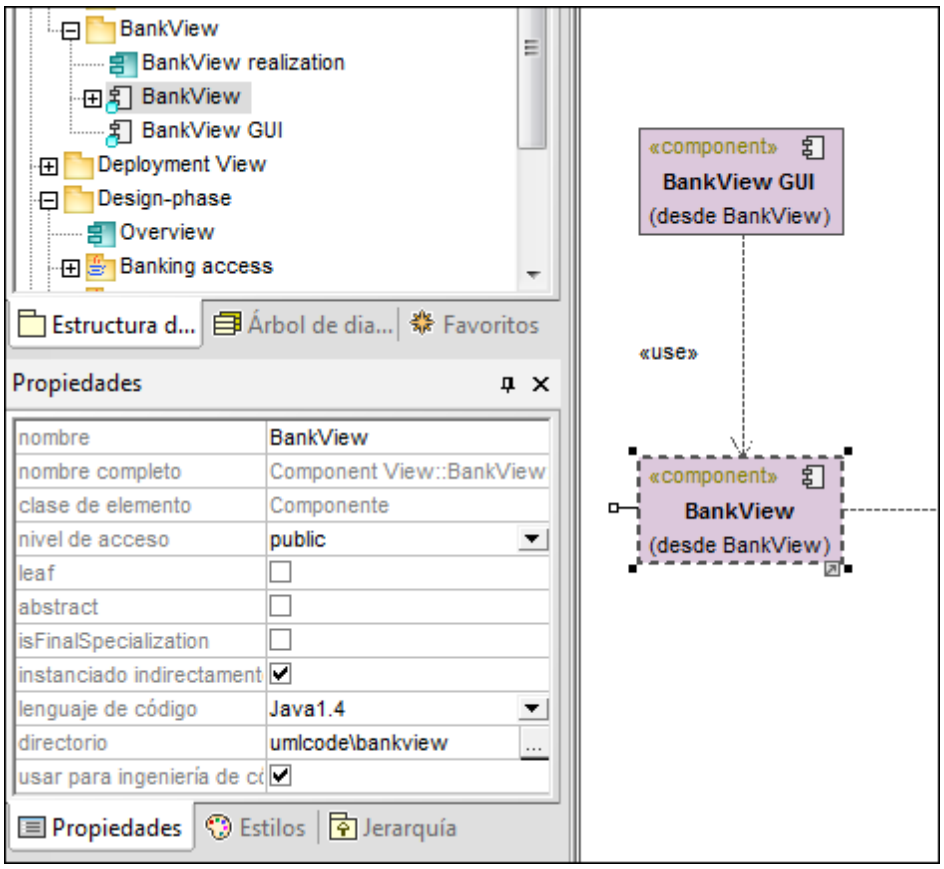
Para crear una realización de componente:

1. Arrastre la clase o interfaz al componente correspondiente en el panel Estructura del modelo.

   También puede crear una realización en un diagrama de componentes usando el icono **Realización** de la barra de herramientas.

Para asignar una ubicación física:

1. Seleccione el componente en la Estructura del modelo o en el diagrama.
2. Haga clic en el botón **Examinar** ... de la propiedad directorio y seleccione un `directorio`.



Para incluir componentes en el proceso de ingeniería de código:

1. Seleccione el componente en la Estructura del modelo o en el diagrama.
2. En el panel Propiedades active la casilla `usar para ingeniería de código`.

Para definir la raíz de espacio de nombres Java:

1. Haga clic con el botón derecho en un paquete y seleccione **Establecer como raíz de espacio de nombres de Java**.

   Esto significa que este paquete y todos los subpaquetes se habilitan durante el proceso de ingeniería de código. En el panel Estructura del modelo la raíz de espacio de nombres Java se señala con el icono  .

- Seleccione el comando **Establecer como raíz de espacio de nombres de Java** otra vez para quitar el espacio de nombres Java del paquete.

6.6 Correspondencia entre código Java y elementos de UModel

La tabla que aparece más abajo muestra la correspondencia entre:

- Elementos de UModel y elementos de código Java cuando se genera código a partir del modelo.
- Elementos de código Java y elementos UModel cuando se actualiza el modelo con el código.

A continuación puede ver un resumen de la tabla de correspondencia. Para ver la versión completa en su explorador web, haga clic aquí.

<table>
<tr><th colspan="5">Java ◁▷ UModel</th></tr>
<tr><th colspan="3">Java</th><th colspan="2">UModel</th></tr>
<tr><td>Project</td><td colspan="2">projectfile
directory</td><td colspan="2">projectfile
directory</td><td>Component</td></tr>
<tr><td>Package</td><td colspan="2">name</td><td colspan="2">name</td><td>Package
<<namespace>></td></tr>
<tr><td rowspan="11"></td><td colspan="2">name</td><td colspan="2">name</td><td rowspan="11"></td></tr>
<tr><td rowspan="6">modifiers</td><td>package</td><td rowspan="4">visibility</td><td>package</td></tr>
<tr><td>public</td><td>public</td></tr>
<tr><td>protected</td><td>protected</td></tr>
<tr><td>private</td><td>private</td></tr>
<tr><td>abstract</td><td colspan="2">abstract</td></tr>
<tr><td>final</td><td colspan="2">isFinalSpecification</td></tr>
<tr><td>strictfp</td><td colspan="2"><<strictfp>></td></tr>
<tr><td colspan="2">filename</td><td colspan="2">code file name</td></tr>
<tr><td colspan="2">associated projectfile/directory</td><td colspan="2">ComponentRealization</td></tr>
<tr><td colspan="2">extends clause</td><td colspan="2">Generalization</td></tr>
<tr><td colspan="2">implements clause</td><td colspan="2">InterfaceRealization(s)</td></tr>
<tr><td colspan="2">java docs</td><td colspan="2">Comment(->Documentation)</td></tr>
</table>

6.7 Correspondencia entre código C# y elementos de UModel

La tabla que aparece más abajo muestra la correspondencia entre:

- Elementos de UModel y elementos de código C# cuando se genera código a partir del modelo.
- Elementos de código C# y elementos UModel cuando se actualiza el modelo con el código.

A continuación puede ver un resumen de la tabla de correspondencia. Para ver la versión completa en su explorador web, haga clic aquí.

C# ◁▷ UModel					
C#			UModel		
Project	projectfile directory		projectfile directory		Component
Namespace	name		name		Package <<namespace>>
	name		name		
	modifiers	internal	visibility	package	
		protected internal		protected <<internal>>	
		public		public	
		protected		protected	
		private		private	
		sealed	isFinalSpecification		
		abstract	abstract		
		static	<<static>>		
		unsafe	<<unsafe>>		
		partial	<<partial>>		
		new	<<new>>		
	filename		code file name		
	associated projectfile/directory		ComponentRealization		
	base types		Generalization, InterfaceRealization(s)		
	attribute sections		<<attributes>>		
	doc comments		Comment(->Documentation)		

6.8 Correspondencia entre elementos de XML Schema y de UModel

La tabla que aparece más abajo muestra la correspondencia entre:

- Elementos de UModel y elementos de XML Schema cuando se genera código a partir del modelo.
- Elementos de XML Schema y elementos UModel cuando se actualiza el modelo con el código.

Los elementos UML/XSD aparecen en **color azul**.

Las propiedades de estereotipo (=valores etiquetados) aparecen en **color verde**.

A continuación puede ver un resumen de la tabla de correspondencia. Para ver la versión completa en su explorador web, haga clic aquí.

XSD ◁▷ UModel						
XSD				**UModel**		
file path				project file		Component
	target namespace			name		Package <<namespace>>
	attributeFormDefault			attributeFormDefault		
	blockDefault			blockDefault		
	elementFormDefault			elementFormDefault		
	finalDefault			finalDefault		
	version			version		
	xml:lang			xml:lang		
	xmlns			xmlns		
	annotation	source		source		
		appinfo			Comment <<appinfo>>	
		documentation	xml:lang	xml:lang	Comment <<documentation>>	

6.9 Correspondencia entre código VB.NET y de elementos de UModel

La tabla que aparece más abajo muestra la correspondencia entre:

- Elementos de UModel y elementos de código VB.NET cuando se genera código a partir del modelo.
- Elementos de código VB.NET y elementos UModel cuando se actualiza el modelo con el código.

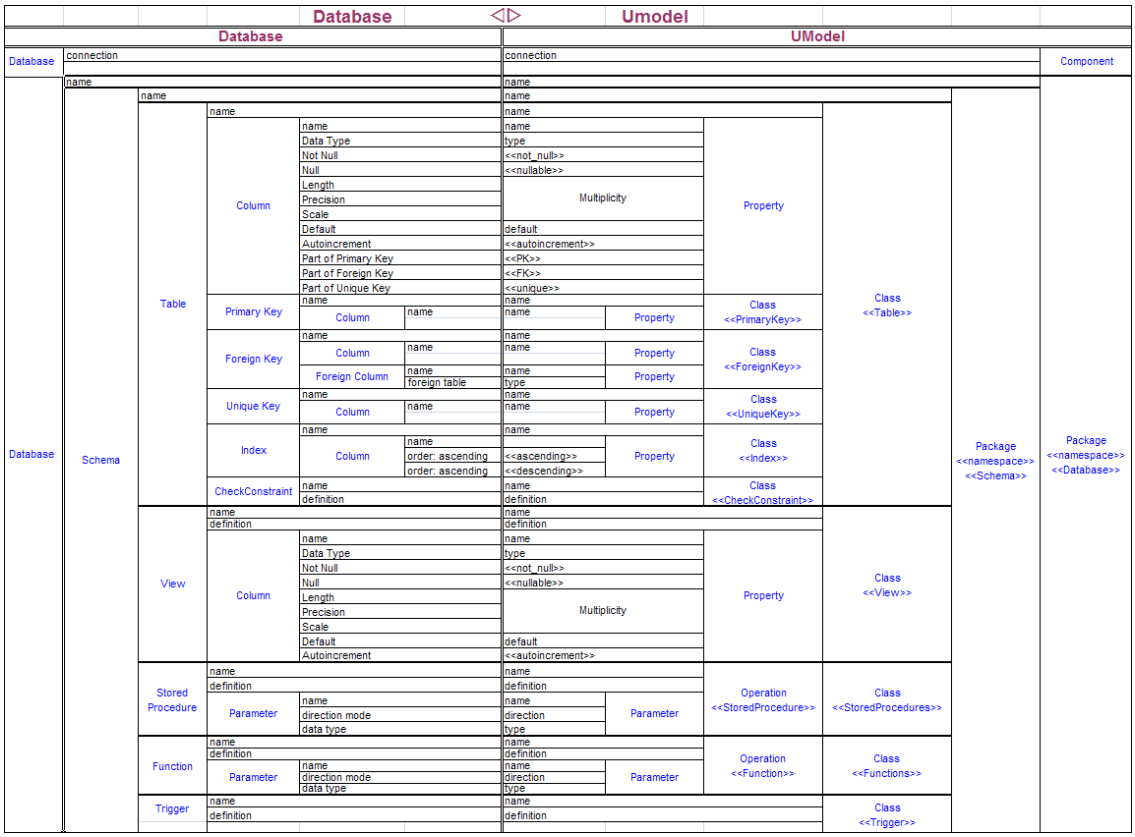
A continuación puede ver un resumen de la tabla de correspondencia. Para ver la versión completa en su explorador web, haga clic aquí.

VB.NET ◁▷ UModel					
VB.NET			UModel		
Project	projectfile		projectfile		Component
	directory		directory		
Namespace	name		name		Package <<namespace>>
	name		name		
	modifiers	Friend	visibility	package	
		Protected Friend		protected <<Friend>>	
		Public		public	
		Protected		protected	
		Private		private	
		NotInheritable	isFinalSpecification		
		MustInherit	abstract		
		Partial	<<Partial>>		
		Shadows	<<Shadows>>		
	filename		code file name		
	associated projectfile/directory		ComponentRealization		
	base types		Generalization, InterfaceRealization(s)		
	attribute sections		<<Attributes>>		
	doc comments		Comment(->Documentation)		

6.10 Correspondencia entre elementos de BD y de UModel

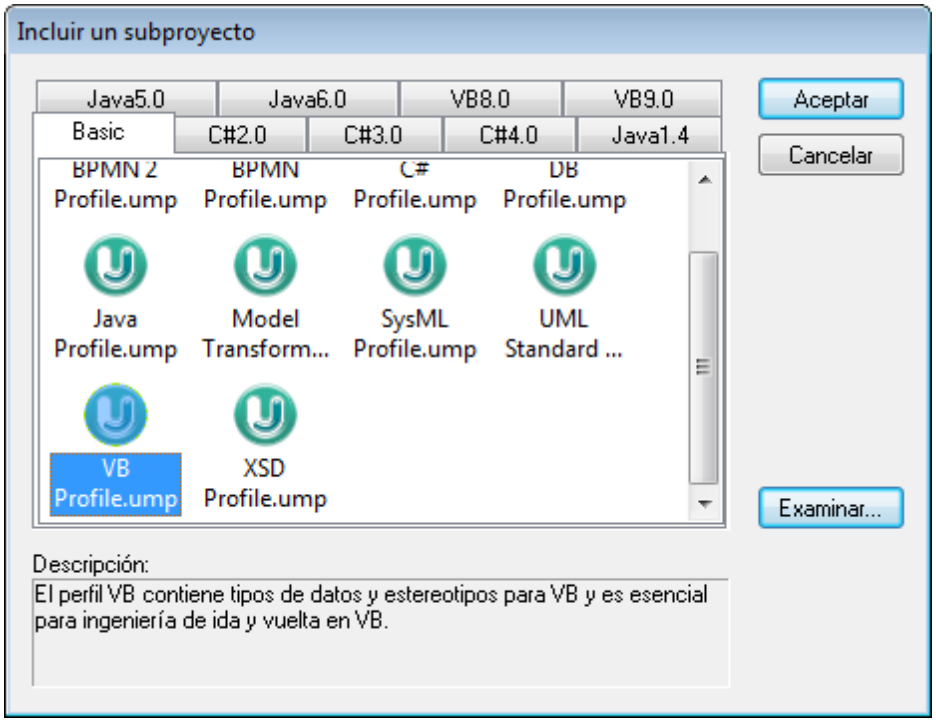
La tabla que aparece más abajo muestra la correspondencia entre:

- Elementos de UModel y elementos de BD cuando se genera código a partir del modelo.
- Elementos de BD y elementos UModel cuando se actualiza el modelo con el código.



6.11 Incluir otros proyectos de UModel

Puede incluir proyectos de UModel dentro de otros con el comando **Proyecto | Incluir un subproyecto**.

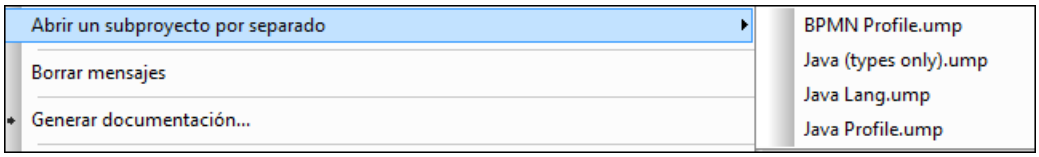


Las pestañas y los proyectos de UModel (archivos `.ump`) que aparecen en el cuadro de diálogo "Incluir un subproyecto" se pueden configurar. UModel obtiene esta información de esta ruta de acceso relativa a la carpeta "`Archivos de programa`" del sistema: `\Altova\UModel2017\UModelInclude`. Recuerde que los archivos de proyecto que aparecen en la pestaña *Basic* existen en la carpeta `UModelInclude` directamente, mientras que los proyectos de las pestañas *Java*, *VB* y *C#* existen en las correspondientes subcarpetas de la carpeta `UModelInclude`.

UModel viene con varios perfiles de proyecto (archivos `.ump`) que pueden ser de gran utilidad para sus proyectos. Por ejemplo, en las pestañas *Java* encontrará paquetes, interfaces y clases del lenguaje Java. Asimismo, en las pestañas *C#* y *VB* encontrará archivos .ump para estos lenguajes.

Para ver todos los proyectos que se importaron al proyecto:

- Seleccione el comando de menú **Proyecto | Abrir un subproyecto por separado**. El menú desplegable enumera todos los subproyectos que forman parte del proyecto.



Para crear una pestaña nueva en el cuadro de diálogo "Incluir un subproyecto":

- Navegue hasta la carpeta `\Altova\UModel2017\UModelInclude` (relativa a su carpeta `Archivos de programa`) y cree la carpeta correspondiente a la pestaña que desea crear en el cuadro de diálogo (p. ej. `\UModelInclude\micarpeta`). El nombre que le asigne a la carpeta será el nombre que toma la pestaña en el cuadro de diálogo "Incluir un subproyecto".
- Copie los archivos `.ump` pertinentes en la carpeta que acaba de crear para que estén disponibles en el cuadro de diálogo "Incluir un subproyecto".

Para crear una descripción para cada archivo de proyecto de UModel:

- Cree un archivo de texto cuyo nombre debe ser el mismo que el del archivo `.ump` y guárdelo en la misma carpeta. Por ejemplo, el archivo `MiModelo.ump` necesita un archivo de texto llamado `MiModelo.txt`. **Nota:** la codificación del archivo debe ser **UTF-8**.

Para eliminar un subproyecto incluido en el proyecto:

1. Haga clic en el subproyecto incluido en la vista Estructura del modelo y pulse la tecla **Supr**.
2. Aparece un aviso preguntando si desea continuar con el proceso de eliminación.
3. Haga clic en **Aceptar** para eliminar el archivo incluido del proyecto.

Para eliminar o quitar un proyecto del cuadro de diálogo "Incluir un subproyecto":

- Elimine o quite el archivo `.ump` que desea eliminar de la carpeta correspondiente en el sistema de archivos.

6.12 Combinar proyectos de UModel

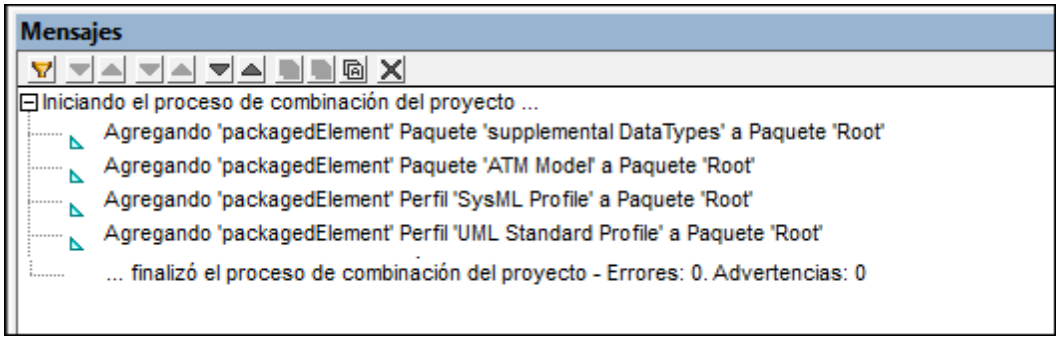
En UModel puede realizar fusiones a 2 y 3 bandas para combinar varios proyectos de UModel distintos en un solo modelo `*.ump`.

Esta función es de gran utilidad si en el mismo proyecto trabajan varias personas a la vez o si quiere reunir todo su trabajo en un solo modelo.

6.12.1 Fusión de proyectos a 2 bandas

Para combinar dos proyectos UML:

1. Abra el archivo UML que debe hacer de archivo de destino (es decir, el archivo con el que se debe combinar el otro modelo).
2. Seleccione el comando **Proyecto | Combinar el proyecto**.
3. Seleccione el proyecto UML que debe combinarse con el proyecto principal.
   En la ventana Mensajes puede comprobar cómo se desarrolla el proceso de combinación.



Nota: si hace clic en una entrada de la ventana Mensajes el elemento de modelado correspondiente aparece resaltado en la Estructura del modelo.

¿Qué efectos tiene la fusión de proyectos a 2 bandas?

- **Elementos de modelado nuevos:** los elementos que no existían en el proyecto principal se añaden al proyecto.
- **Diferencias entre los mismos elementos de modelado:** los elementos del segundo modelo tienen prioridad. Por ejemplo, solo puede haber un valor predeterminado de un atributo, así que se usa el valor predeterminado del segundo modelo.
- **Diferencias entre diagramas**: UModel comprueba si hay diferencias entre los diagramas de los dos modelos.

  - Si hay diferencias, los diagramas nuevos/diferentes se añaden al modelo principal (con un sufijo numérico tipo `actividad1` etc.) y el diagrama original se conserva.

  - Si no hay diferencias, no se realiza ningún cambio. Después puede eliminar los diagramas que quiera.
- El proceso de combinación entero se puede deshacer paso a paso con el comando

Deshacer (**Ctrl+Z** o con el botón de la barra de herramientas).

- Al hacer clic en una entrada de la ventana Mensajes aparece resaltado el elemento de modelado correspondiente en la Estructura del modelo
- El proyecto principal combinado conserva el mismo **nombre de archivo**.

6.12.2 Fusión de proyectos a 3 bandas

UModel también ofrece una función para combinar varios proyectos de UModel editados por programadores diferentes. Esta función se conoce como *fusión de proyectos a 3 bandas*.

Combinar varios proyectos de UModel

La fusión de proyectos a 3 bandas sirve para combinar proyectos de UModel de nivel superior, es decir, proyectos principales que pueden contener subproyectos. La fusión a 3 bandas no funciona con archivos independientes si tienen referencias sin resolver a otros archivos.

Cuando se combinan proyectos principales, sus subproyectos editables también se combinan automáticamente. Es decir, no hace falta combinar los subproyectos por separado.

Para ver un ejemplo consulte el apartado Ejemplo de fusión manual a 3 bandas.

- El proceso de combinación se puede deshacer entero paso a paso con el botón **Deshacer** de la barra de herramientas o con **Ctrl+Z**.
- Si hace clic en una entrada de la ventana Mensajes, el elemento de modelado correspondiente aparece resaltado en la Estructura del modelo.
- El proyecto principal combinado (el que se eligió como destino de la combinación) conserva el mismo **nombre de archivo**.

¿Qué efectos tiene la fusión de proyectos a 3 bandas?

Es importante tener en cuenta que cuando decimos "proyecto original", nos referimos al primer archivo de proyecto que se eligió como destino de la combinación al principio del proceso.

- **Elementos de modelado nuevos:**
  - si el segundo proyecto tiene elementos que no existen en el proyecto original, estos elementos se añaden al proyecto de destino (resultado de la fusión).
  - si el proyecto original tiene elementos que no existen en el segundo proyecto, estos elementos se conservan en el proyecto de destino (resultado de la fusión).
- **Elementos de modelado eliminados:**
  - si en el segundo proyecto se eliminaron elementos que todavía existen en el proyecto original, estos elementos se eliminan en el proyecto de destino (resultado de la fusión).
  - si en el proyecto original se eliminaron elementos que todavía existen en el segundo proyecto, estos elementos se conservan en el proyecto de destino (resultado de la fusión).
- **Diferencias entre los mismos elementos de modelado:**
  - si una propiedad (p. ej. el nivel de acceso de una clase) se modificó en el proyecto original o en el segundo proyecto, el modelo de destino incluirá el valor más reciente.
  - si una propiedad (p. ej. el nivel de acceso de una clase) se modificó tanto en el

archivo original como en el segundo archivo, el modelo de destino incluirá el valor del segundo archivo (y aparece una advertencia en la ventana Mensajes).

- **Elementos con una posición distinta:**
  - si un elemento cambió de posición en el proyecto original o en el segundo proyecto, en el modelo de destino también se cambia la posición del elemento.
  - si un elemento cambió de posición tanto en el proyecto original como en el segundo proyecto, aparece un aviso y el usuario puede seleccionar el elemento primario donde se debe insertar el elemento en el modelo de destino.

- **Diferencias entre diagramas**:
  UModel comprueba si hay diferencias entre los diagramas de los dos modelos.
  - si hay diferencias, el diagrama nuevo o diferente se añade al modelo de destino (con un sufijo numérico tipo `actividad1`, etc.) y el diagrama original se conserva.
  - si no hay diferencias, no se realizan cambios. Después puede eliminar los diagramas que quiera.

Sistemas de control de versiones para fusiones a 3 bandas

Cuando trabaje con un sistema de control de versiones y proteja/desproteja archivos de proyecto, UModel genera automáticamente archivos de instantánea denominados *archivo antecesor común,* que se utilizan para el proceso de fusión a 3 bandas y que permiten una combinación mucho más precisa que la fusión a 2 bandas.

La fusión a 3 bandas **automática** de UModel no funciona con todos los sistemas de control de versiones, pero en casos así puede usar la fusión a 3 bandas **manual**.

- Los sistemas de control de versiones que combinan archivos automáticamente sin la intervención del usuario no suelen ser compatibles con la fusión a 3 bandas automática de UModel.
- Los sistemas de control de versiones que piden que el usuario elija entre **Reemplazar** o **Combinar** cuando cambia un archivo de proyecto suelen ser compatibles con la fusión a 3 bandas automática de UModel. Después de que el control de versiones reemplace el archivo, seleccione el comando **Reemplazar** para activar el aviso de UModel que permite realizar una fusión a 3 bandas. Para el proceso de protección/desprotección de los archivos es necesario usar UModel.
- Puede poner bajo control de versiones tanto el proyecto principal como sus subproyectos. Si cambia los datos de un subproyecto, recibirá un aviso automático para desproteger el subproyecto.
- Cada acción de protección/desprotección crea un *archivo antecesor común* o de instantánea que se utiliza después durante el proceso de fusión a 3 bandas.

Nota: los archivos de instantánea solo se crean y emplean automáticamente con la versión independiente de UModel (es decir, esta función no está disponible en el complemento de UModel para Eclipse o Visual Studio).

Por ejemplo, imagine que `Usuario A` edita un archivo de proyecto de UModel y cambia el nombre de una clase en el diagrama `BankView Main`. Ahora imagine que `Usuario B` abre el mismo archivo de proyecto y cambia el nivel de acceso de esa misma clase.

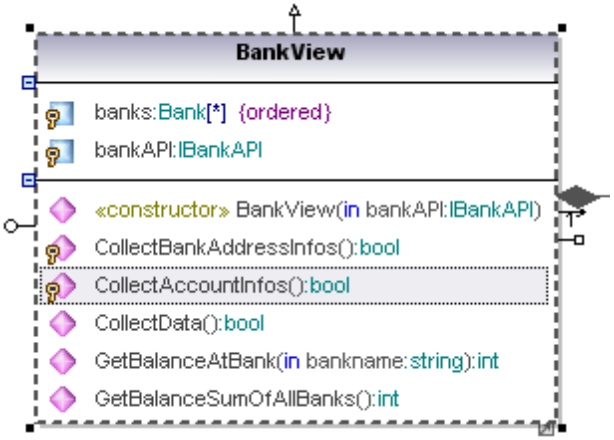
Como UModel genera un archivo de instantánea para cada usuario, el historial de edición de instantáneas permite combinar los diferentes cambios en el proyecto. Como resultado del

proceso de fusión a 3 bandas, el nuevo nombre de la clase y su nuevo nivel de acceso se pasan al archivo de proyecto de destino (resultado de la fusión).

6.12.3 Ejemplo de fusión manual a 3 bandas

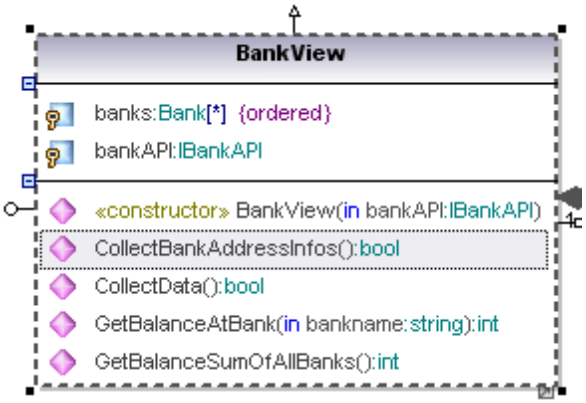
Para este sencillo ejemplo utilizamos el archivo de muestra `Bank_CSharp.ump` disponible en la carpeta `...\UModelExamples`. Las otras dos instancias del mismo proyecto se copian en dos carpetas secundarias (`\C#_1` y `\C#_2`) bajo la carpeta `UModelExamples`.

Para empezar, imagine que el `Usuario1` abre el archivo de proyecto `Bank_CSharp.ump` de la carpeta `C#_1` y realiza cambios en la clase **`BankView`**.



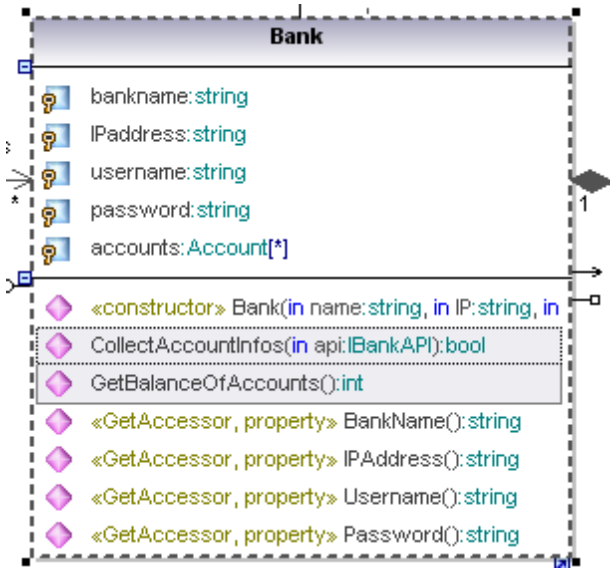
Cambios realizados por Usuario1 en la clase BankView:

1. `Usuario1` **elimina** la operación **`CollectAccountInfos():bool`** de la clase **`BankView`**.
2. Después cambia el **nivel de acceso** de la operación **`CollectBankAddressInfos():bool`** de `protected` a `public`.



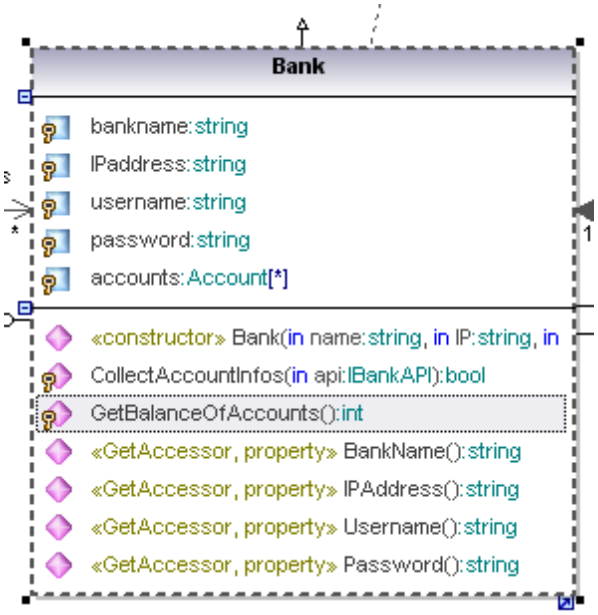
3. Y, por último, guarda el proyecto.

Ahora imagine que el `Usuario2` abre el archivo de proyecto `Bank_CSharp.ump` de la carpeta `C#_2` y realiza cambios en la clase **`Bank`**.



Cambios realizados por Usuario2 en la clase Bank:

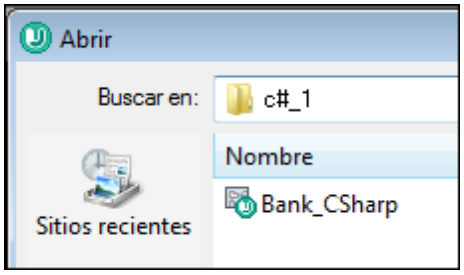
1. `Usuario2` cambia el nivel de acceso de las operaciones **`CollectAccountInfos`** y **`GetBalanceOfAccounts`** de `public` a `protected`.



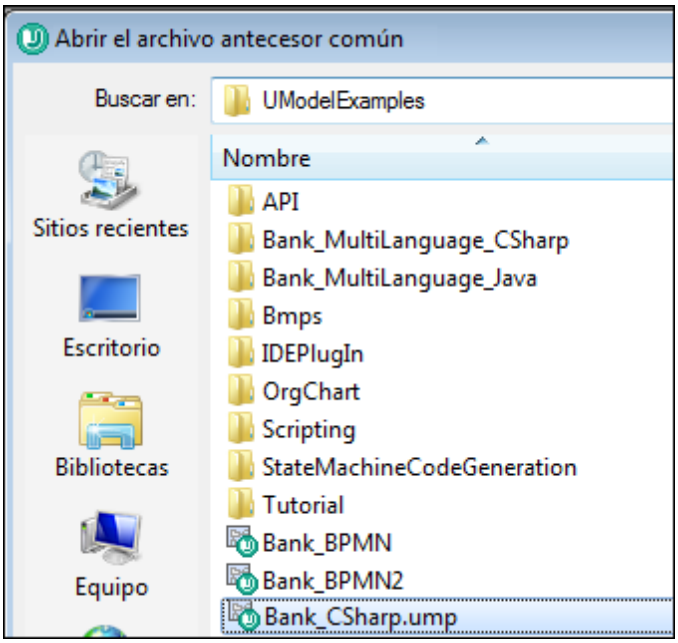
2. Y después guarda el proyecto.

Usuario2 inicia una fusión de proyecto a 3 bandas:

1. Primero selecciona el comando **Proyecto | Combinar el proyecto (fusión a 3 bandas)**. Aparece el cuadro de diálogo "Abrir".
2. Después selecciona el archivo de proyecto que modificó `Usuario1` en la carpeta `. . . \C#_1`.

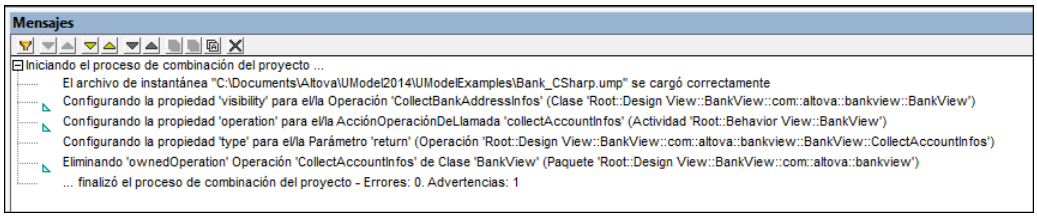


Ahora aparece un cuadro de diálogo "Abrir el archivo antecesor común", que es el archivo de proyecto original de la carpeta `...\UModelExamples`.



El proceso de fusión a 3 bandas se inicia y se retorna al archivo de proyecto desde el que se inició el proceso de fusión (es decir, desde el archivo de proyecto de la carpeta `C#_2`).

La ventana Mensajes muestra el desarrollo del proceso de combinación.



- Los cambios realizados en el proyecto de la carpeta `C#_1` se reproducen en el proyecto de la carpeta `C#_2`.
- Los cambios realizados en el proyecto de la carpeta `C#_2` se conservan en el archivo de proyecto.
- El archivo de proyecto de la carpeta `C#_2` debería utilizarse como archivo antecesor común para los procesos de fusión a 3 bandas que se realicen a partir de ahora con los archivos de proyecto de las carpetas `C#_1` y `C#_2`.

6.13 Compartir paquetes y diagramas

En UModel puede compartir paquetes y diagramas UML con otros proyectos. A su vez, los paquetes se pueden incluir en otros proyectos de UModel mediante referencia o como copia.

Además los subproyectos se pueden separar del proyecto principal en cualquier momento e incluirse otra vez en el proyecto principal como subproyectos editables o de solo lectura. Asimismo, los paquetes se comparten y se guardan como archivos de subproyecto.

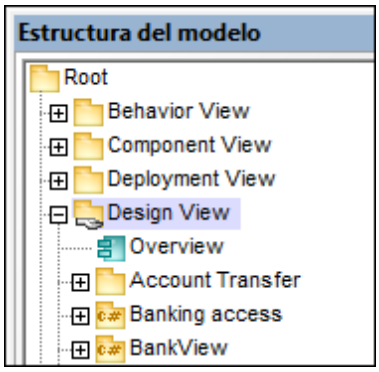
Nota: no olvide que los subproyectos también se pueden añadir al sistema de control de código fuente. Para más información consulte el apartado Trabajo en equipo con UModel.

Requisitos para compartir paquetes:
Es importante tener en cuenta que no se puede compartir un paquete que tenga vínculos a otros paquetes situados fuera del ámbito compartido.

Nota: cuando cree archivos de proyecto de UMOdel no use archivos de proyecto como plantilla/ copia de otro archivo de proyecto en el que quiere compartir un paquete. Esto daría lugar a conflictos porque cada elemento debe ser único de forma global (ver información sobre identificadores UUID) y en este caso los dos proyectos tendrían elementos con el mismo identificador uuid.

Para compartir un paquete con otros proyectos:

1. En la Estructura del modelo haga clic con el botón derecho en el paquete que desea compartir. Seleccione **Subproyecto | Compartir paquete**.

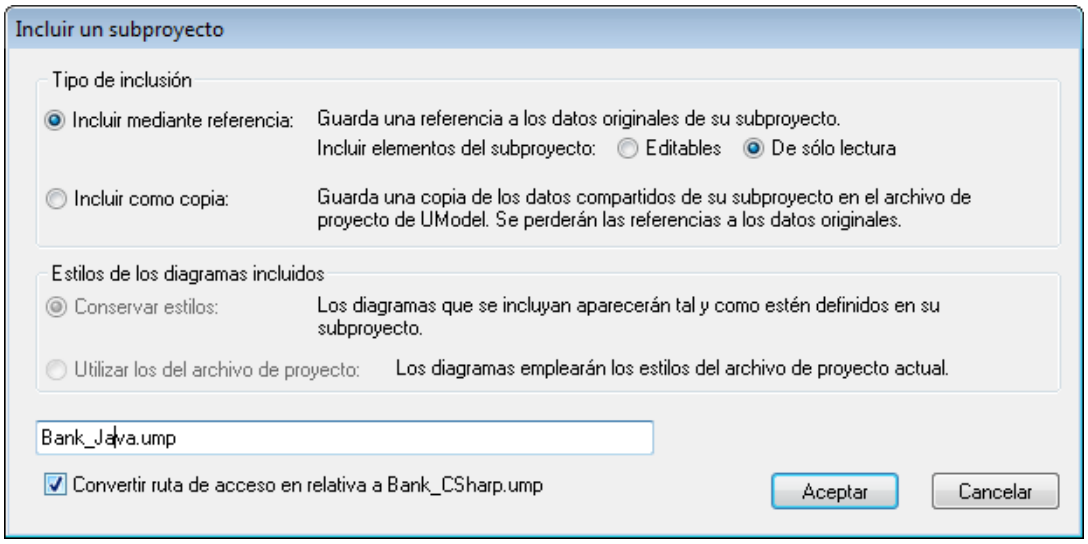


El icono del paquete ahora incluye una mano que indica que el paquete está compartido. A partir de ahora este paquete se podrá incluir en cualquier otro proyecto de UModel.

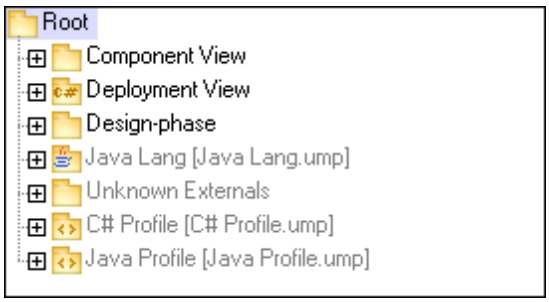
Para incluir/importar una carpeta compartida en un proyecto:

1. Abra el proyecto en el que desea incluir el paquete compartido (en este caso, un archivo vacío).

2. Seleccione la opción de menú **Proyecto | Incluir un subproyecto...**.
3. Aparece el cuadro de diálogo "Incluir un subproyecto" (*imagen siguiente*). Haga clic en el botón **Examinar**, seleccione el proyecto que incluye el paquete compartido y haga clic en **Abrir**.



4. En el cuadro de diálogo "Incluir un subproyecto" (*imagen anterior*) puede elegir si el paquete/proyecto se incluye mediante referencia o como copia.
Seleccione una de las dos opciones y haga clic en **Aceptar**.

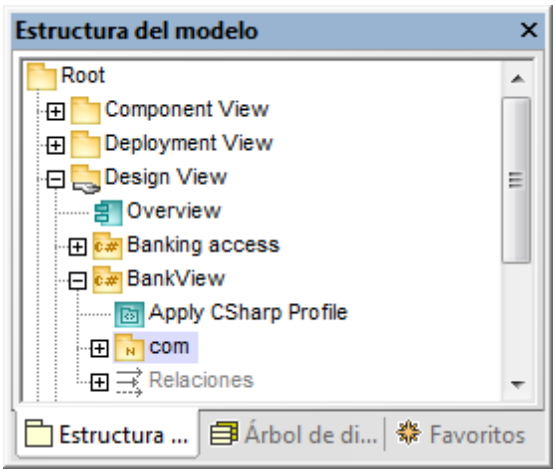


El paquete `Deployment View` aparece ahora en el paquete nuevo. El proyecto de origen del paquete aparece entre paréntesis (`BankView-start.ump`).

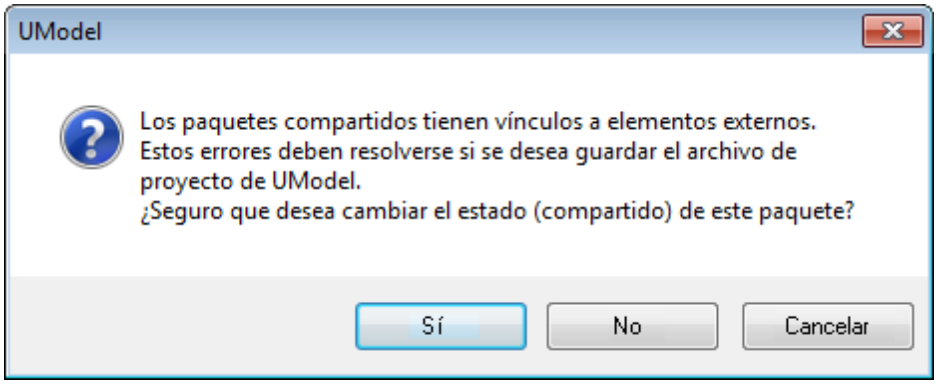
Las carpetas compartidas que se incluyeron mediante referencia se pueden cambiar a *Incluir como copia* en cualquier momento. Esto se consigue haciendo clic con el botón derecho en la carpeta y seleccione el comando **Subproyecto | Incluir como copia**.

Nota: también se incluyen todos los proyectos incluidos en el proyecto de origen: `Java Lang`, `Unknown Externals` y `Java Profile`.

Paquetes compartidos con vínculos a elementos externos:

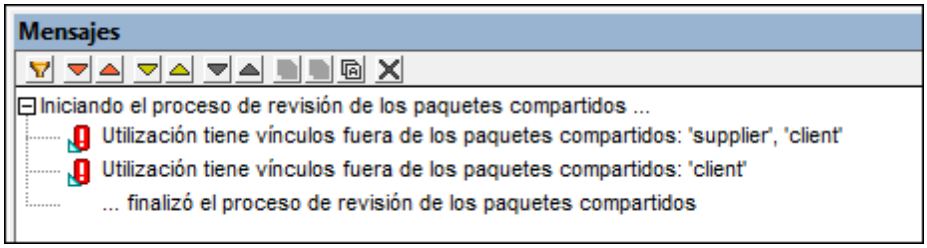


Si intenta compartir un paquete que tiene vínculos a elementos externos, aparece un aviso.

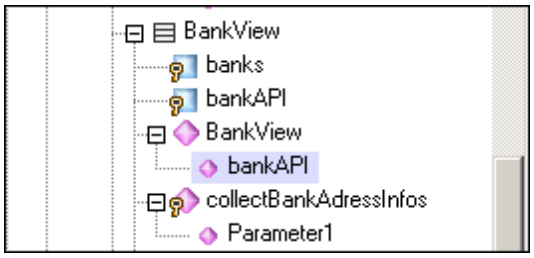


Si hace clic en **Sí**, UModel resuelve los vínculos externos antes de guardar el proyecto.

El panel Mensajes ofrece información sobre los vínculos externos.



Si hace clic en una entrada del panel Mensajes, el elemento correspondiente se resalta en el panel Estructura del modelo.



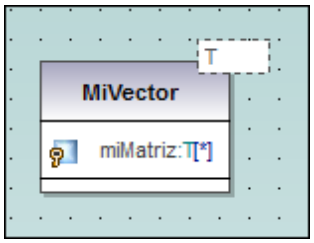
6.14 Plantillas UML

En UModel puede usar plantillas UML y crear asignaciones entre ellas y genéricos Java, C# y Visual Basic.

- Las plantillas son elementos de modelado potenciales con parámetros formales sin enlazar.
- Estos elementos de modelado parametrizados describen un grupo de elementos de modelado de un tipo concreto: clasificadores u operaciones.
- Las plantillas no se pueden usar como tipos directamente, sus parámetros deben estar enlazados.
- *Generar instancias* significa enlazar los parámetros de la plantilla con valores reales.
- Los valores reales de los parámetros son expresiones.
- El enlace que existe entre una plantilla y un elemento de modelado produce un elemento de modelado nuevo (elemento enlazado) basado en la plantilla.
- Si en C# existen varios clasificadores de restricción, los parámetros de la plantilla se pueden editar directamente en el panel Propiedades cuando se selecciona el parámetro de la plantilla.

Presentación de firmas de plantilla en UModel:

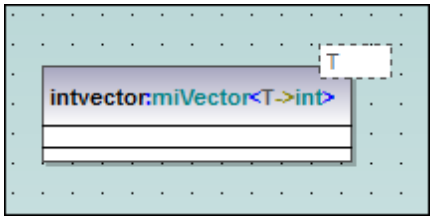
- En la imagen que aparece a continuación puede verse la plantilla de clase **`MiVector`**, cuyo parámetro de plantilla formal (`T`) aparece en la esquina superior derecha en un rectángulo discontinuo.
- Los parámetros formales sin información de tipo (`T`) son clasificadores implícitos: clase, tipo de datos, enumeración, tipo primitivo e interfaz. Los demás tipos de parámetro deben mostrarse explícitamente (p. ej. los enteros).
- La propiedad **`miMatriz`** tiene un número ilimitado de elementos de tipo `T`.



Cuando se hace clic con el botón derecho en la plantilla, aparece un menú contextual con el comando **Mostrar | Elementos enlazados**. Este comando muestra los elementos enlazados propiamente dichos.

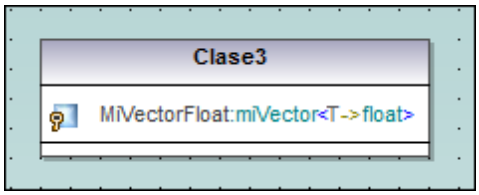
Presentación de enlaces de plantilla en UModel:

- En la imagen que aparece a continuación puede verse la plantilla con nombre enlazada **`intvector`**.
- Se trata de una plantilla de tipo **`miVector`**.
- El parámetro `T` se sustituye con `int`.
- Los caracteres `->` significan *sustituido con*.



Uso de plantillas en operaciones/propiedades:

- En la imagen anterior puede ver un enlace de plantilla anónimo.
- Tiene la propiedad **`MiVectorFloat`** de tipo **`MiVector<T->float>`**.



También puede definir plantillas cuando defina propiedades u operaciones. La función de finalización automática le ayudará a utilizar la sintaxis correcta.

- La operación `Operación1` devuelve un vector de tipo float.

6.14.1 Firmas de plantilla

Una *firma de plantilla* es una cadena de texto que especifica los parámetros de plantilla formales. Por su parte, una *plantilla* es un elemento parametrizado que se utiliza para generar elementos de modelado nuevos mediante la sustitución o el enlace de parámetros formales con parámetros reales (valores).

Parámetro de plantilla formal

`T`

Plantilla con un solo parámetro formal sin tipo
(almacena elementos de tipo T)

Varios parámetros de plantilla formales

`KeyType:DateType, ValueType`

Sustitución de parámetros

`T>unaClaseBase`

La sustitución de parámetros debe ser de tipo `unaClaseBase` o derivarse de ese tipo.

Valores predeterminados para parámetros de plantilla

```
T=unValorPredeterminado
```

Clasificadores de sustitución

```
T>{contract}unaClaseBase
allowsSubstitutable es true
```

El parámetro debe ser un clasificador que puede ser sustituido con el clasificador designado por el nombre de clasificador.

Parámetros de plantilla de restricción

```
T:Interface>unaInterfaz
```

Cuando la restricción limite a un elemento que no sea una clase (una interfaz, un tipo de datos), la restricción aparece después del carácter ":". Por ejemplo, `T` está restringido a una interfaz (`T:Interfaz`), que debe ser de tipo "`unaInterfaz`" (`>unaInterfaz`).

Usar comodines en firmas de plantilla

```
T>vector<T->?<unaClaseBase>
```

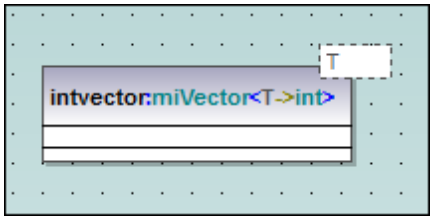
El parámetro de plantilla `T` debe ser de tipo "vector" que contiene objetos que son un supratipo de `unaClaseBase`.

Parámetros de plantilla de extensión

```
T>Comparable<T->T>
```

6.14.2 Enlace de plantilla

Un *enlace de plantilla* es el resultado de sustituir los parámetros formales con los valores reales (es decir, se crea una instancia de la plantilla). UModel genera automáticamente clases enlazadas anónimamente cuando se produce este enlace. Los enlaces se pueden definir en el campo del nombre de la clase, como puede ver en la imagen siguiente.



Parámetros formales de sustitución/enlace

```
vector <T->int>
```

Crear enlaces usando el nombre de la clase

```
a_float_vector:vector<T->float>
```

Enlazar varias plantillas simultáneamente

```
Clase5:vector<T->int, map<KeyType->int, ValueType<T->int>
```

Usar comodines ? como parámetros (Java 5.0)


```
vector<T->?>
```

Restringir comodines - límites superiores (extensión de UModel)

```
vector<T->?>unaClaseBase>
```

Restringir comodines - límites inferiores (extensión de UModel)

```
vector<T->?<unaClaseDerivada>
```

6.14.3 Usar plantillas en operaciones y propiedades

Operación que devuelve una plantilla enlazada

```
Clase1
Operación1():vector<T->int>
```

El parámetro `T` está enlazado con `int`. La `Operación1` devuelve un vector de tipos `int`.

Clase que contiene una operación de plantilla

```
Clase1
Operación1<T>(in T):T
```

Usar comodines

```
Clase1
Propiedad1:vector<T->?>
```

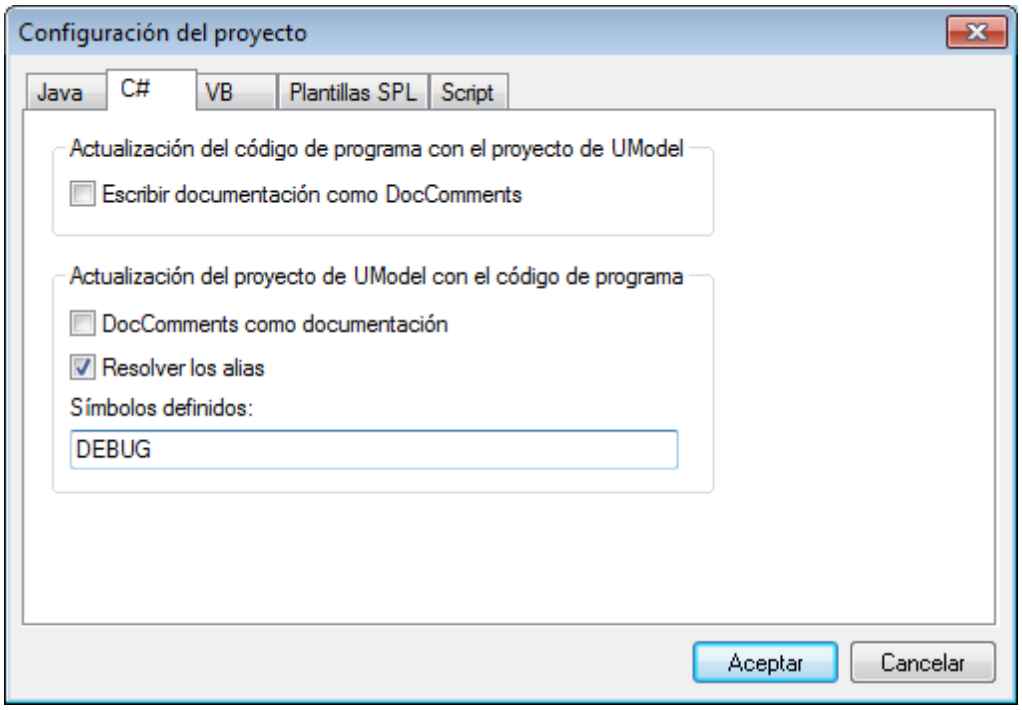
Esta clase contiene un vector genérico cuyo tipo no se ha especificado (? es el comodín).

Para ver las propiedades con tipo como asociaciones:

- haga clic con el botón derecho en una propiedad y seleccione **Mostrar | PropiedadX como asociación** o
- arrastre una propiedad hasta el diagrama.

6.15 Configuración del proyecto

El comando **Proyecto | Configuración del proyecto** sirve para definir las opciones globales de configuración del proyecto.



Para más información consulte el apartado **Herramientas | Opciones** de la *Referencia del usuario*.

6.16 Mejorar el rendimiento

Dado que algunos proyectos de modelado pueden alcanzar un tamaño considerable, hay varias maneras de mejorar el rendimiento del modelo:

- Para empezar, compruebe que utiliza el controlador más reciente de su tarjeta gráfica.
- Deshabilite la función de color de sintaxis (en la pestaña *Estilos* cambie el valor del estilo `usar color de sintaxis` de `true` a `falso`).
- Deshabilite el color de fondo `degradado` para los diagramas. En su lugar utilice un color sólido (p. ej. seleccione el valor `blanco` para el estilo `Diagrama - color de fondo` en la pestaña *Estilos*).
- La finalización automática se habilita por defecto, pero puede desactivarla en la pestaña *Edición de diagramas* del cuadro de diálogo "Opciones locales" (**Herramientas | Opciones**). Para ello desactive la casilla *Habilitar ayudante de entrada automática*.

Chapter 7

Crear relaciones entre los modelos

7 Crear relaciones entre los modelos

Hay varias maneras de crear e insertar relaciones entre los modelos en los diagramas:

- Puede usar los controladores de conexión (por ejemplo, consulte el ejemplo del apartado Casos de uso).
- O puede hacer clic en el icono correspondiente de la barra de herramientas y arrastrar el puntero del ratón para crear las conexiones entre los elementos:

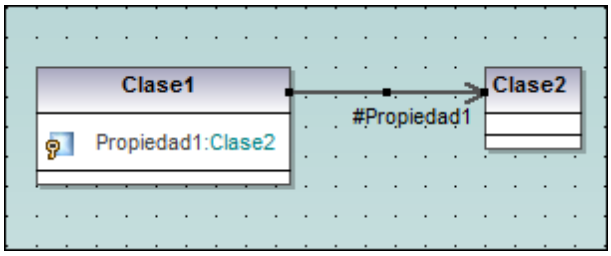
 icono de asociación

 agregación

 composición

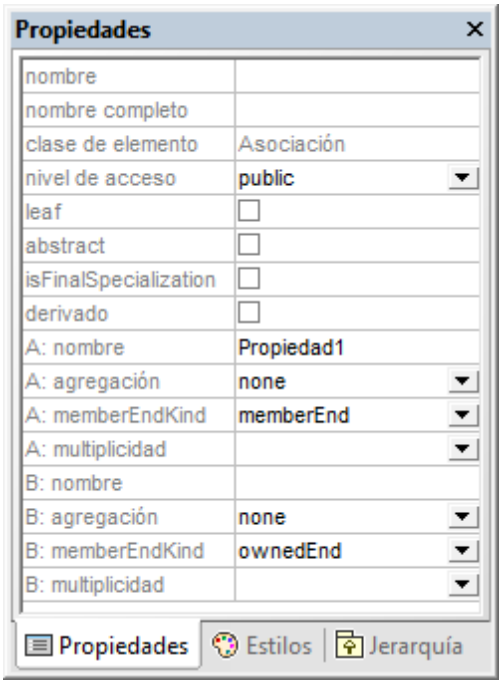
 contención

Cuando se crea una asociación, en la clase de origen (`A:nombre`) se inserta un atributo nuevo automáticamente (ver ejemplo de la siguiente imagen: `Propiedad1:Clase2`).



Una vez creada, la asociación está activa y la pestaña *Propiedades* muestra sus propiedades. El nombre predeterminado del `memberEnd` de la asociación aparece en una etiqueta de texto (es decir, **`Propiedad1`**). Recuerde que la opción **Etiqueta de texto...** del menú contextual permite ver/ocultar las etiquetas.

Al hacer clic en la línea de asociación, sus propiedades aparecen en la pestaña *Propiedades*. Las propiedades `A:Nombre` y `B:Nombre` indican qué papel desempeña cada clase en la otra.



Dependiendo del valor de la propiedad `memberEndKind` (de `A:nombre Propiedad1`), el atributo pertenece:

- a la clase (es decir, **A**:memberEndKind = **memberEnd**) (el atributo está visible en **`clase1`**), o
- a la asociación (es decir, **B**:memberEndKind = **ownedEnd**) (el atributo no está visible en **`clase2`**).

  Si ambos atributos pertenecen a la **asociación** (es decir si ambos extremos se definen como `ownedEnd`), entonces esta asociación es bilateral y desaparece la dirección de la flecha. Ambos extremos de la asociación son de tipo `ownedEnd`.

Si la propiedad `memberEndKind` de la asociación tiene el valor `navigableOwnedEnd`, entonces el atributo todavía forma parte de la asociación pero la flecha de dirección reaparece dependiendo del extremo en el que se defina (`A:nombre` o `B:nombre`).

Para definir el tipo de asociación (asociación, agregación o composición):

1. Haga clic en la flecha de asociación.
2. Desplácese hasta el campo `agregación` de la pestaña *Propiedades*.
3. Seleccione una opción: **none** (asociación estándar), **shared** (asociación por agregación) o **composite** (asociación por composición).

Nota: puede crear asociaciones usando la misma clase como origen y como destino. Esto recibe el nombre de *autovínculo* y hace referencia a la capacidad de un objeto para enviarse un mensaje a sí mismo, para llamadas recursivas. Haga clic en el icono de relación, después haga clic en el elemento y arrastre el puntero del ratón hasta otra parte del mismo elemento. Se crea un autovínculo.

Para ver automáticamente las asociaciones en los diagramas:

Si la opción *Crear asociaciones automáticamente* está activada en la pestaña *Edición* del cuadro de diálogo "Opciones locales " (**Herramientas | Opciones**), cuando inserte elementos en los

diagramas, las asociaciones ya existentes entre los elementos se pueden crear/ver automáticamente en el diagrama. Esto ocurre si está definido el tipo de atributo y el elemento de modelado `type` al que se hace referencia está en el diagrama.

Para eliminar relaciones/asociaciones:

1. Haga clic en la relación, bien en el área de trabajo, bien en la *Estructura del modelo*.
2. Pulse la tecla **Supr**.
   La dependencia se elimina en el diagrama y en el proyecto.

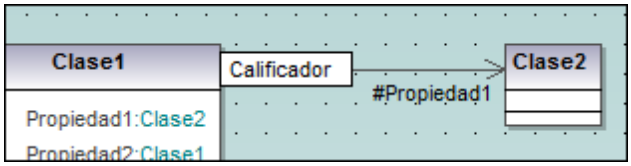
Para eliminar asociaciones de clases:

Cuando se elimina una asociación de **clases**, el **atributo**/la **propiedad** generado automáticamente no se elimina de la clase.

1. Haga clic con el botón derecho en el atributo / la propiedad de la clase.
2. Seleccione la opción **Eliminar propiedad X de la clase X**.

Para crear calificadores de asociación:

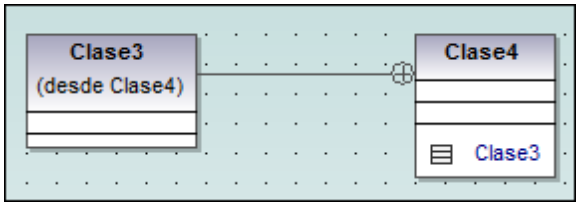
1. Defina la asociación entre las dos clases.
2. Haga clic con el botón derecho en la línea de asociación y seleccione **Nuevo/a | Calificador**.
   Recuerde que los calificadores son atributos de una asociación.



Para crear un asociación de contención:

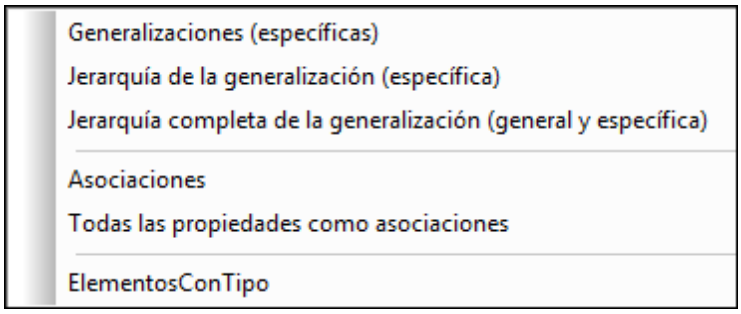
Las asociaciones de contención muestran relaciones entre primarios y secundarios.

1. Haga clic en el icono **contención**  de la barra de herramientas.
2. Haga clic en la clase que debe contenerse y arrastre el puntero hasta la clase que debe contenerla.
   Observe que la clase contenida (`Clase3` en este caso) ahora aparece dentro del compartimento de `Clase4`. Por lo general, esto coloca la clase contenida en el mismo espacio de nombres que la clase contenedora.



7.1 Ver las relaciones entre los modelos

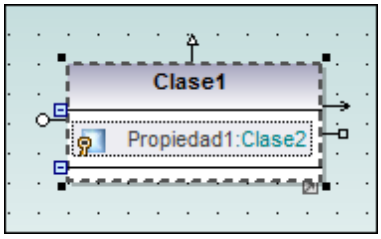
Para ver las relaciones que existen entre los elementos de modelado:
Haga clic con el botón derecho en el elemento que le interesa y seleccione **Mostrar**. Este submenú ofrece diferentes opciones dependiendo del elemento seleccionado.



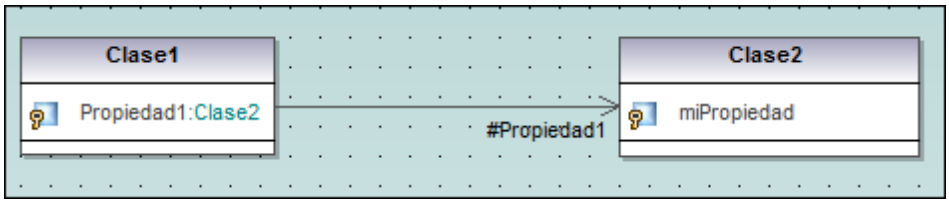
Para ver / ocultar las etiquetas de texto:
Haga clic con el botón derecho en una clase o en una flecha de asociación y seleccione **Etiquetas de texto | Mostrar (ocultar) todas las etiquetas de texto**.

Para ver un atributo / una propiedad de una clase en forma de asociación:

1. Haga clic con el botón derecho en la propiedad de la clase.



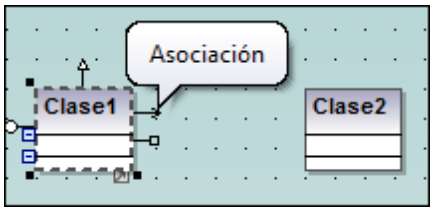
2. Seleccione la opción **Mostrar | <nombre de la propiedad> como asociación**. Esto inserta/abre la clase a la que se hace referencia y muestra la correspondiente asociación.



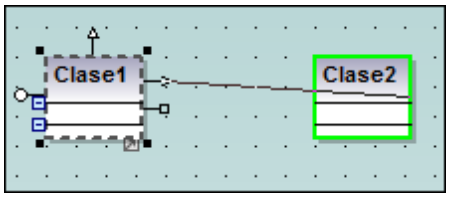
7.2 Asociaciones, realizaciones y dependencias

Para crear relaciones usando los controladores de conexión:

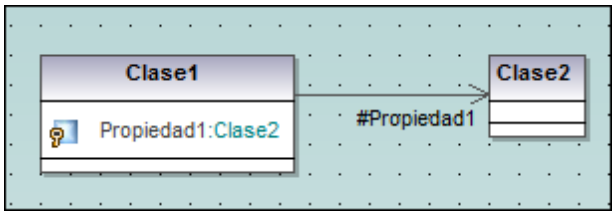
1. Imagine que tiene dos clases en un diagrama de clases.
2. Haga clic en la primera clase para seleccionarla.
   En los bordes de la clase aparecen controladores de conexión.
3. Pase el puntero del ratón por encima del controlador superior del lado derecho de la clase.



   Aparece información sobre el tipo de relación que crea este controlador. En este caso, se trata de una `asociación`.
4. Haga clic en la punta de flecha de la `asociación` y arrastre el puntero del ratón hasta la segunda clase para crear una línea de conexión.
   Si se puede crear este tipo de relación, la clase de destino se resalta. Suelte el botón del ratón.



   Se crea una asociación entre las dos clases.



   Tenga en cuenta que el controlador inferior del lado derecho de la clase crea una asociación de colección.

Los elementos de los diferentes diagramas tienen controladores de conexión distintos. Por ejemplo, las clases de los diagramas de clases incluyen estos controladores de relación (en el orden de las agujas del reloj):

- `RealizaciónDeInterfaz`
- `Generalización`
- `Asociación`

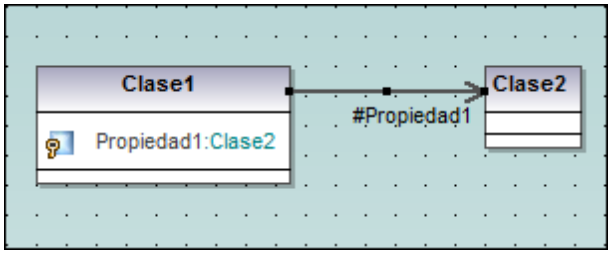
Y los artefactos de la vista **`Deployment View`** tienen estos controladores de relación:

- `Manifestación`
- `Asociación`
- `Implementación`

Para crear relaciones con los iconos de la barra de herramientas:

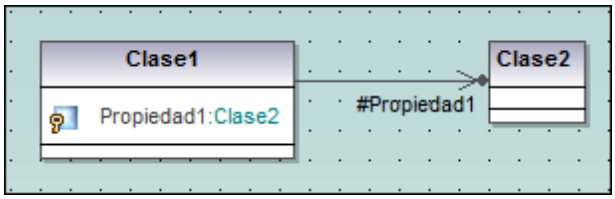
1. Imagine que tiene un diagrama de modelado con dos elementos.
2. Haga clic en el icono que representa la relación que desea crear (p. ej. `asociación`, `agregación`, `composición`, etc.).
3. Arrastre el puntero del ratón desde un objeto hasta el otro y deje de pulsar el botón del ratón cuando el segundo objeto esté resaltado.

   Cuando se crea una asociación nueva, en la clase de origen (`A:nombre`) se crea automáticamente un atributo nuevo (`Propiedad1:Clase2`) como en el ejemplo de esta imagen:



   UModel siempre muestra todos los atributos de una clase.

   Nota: las imágenes de este manual no muestran el **punto de propiedad** de la asociación.



   Para habilitarlo, asigne el valor `true` al estilo `Mostrar pto. de propiedad de la asoc.` de la pestaña *Propiedades*.

Para eliminar relaciones/asociaciones:

1. Haga clic en la relación, bien en el área de trabajo, bien en la *Estructura del modelo*.
2. Pulse la tecla **Supr**.
   La dependencia se elimina en el diagrama y en el proyecto.

Para eliminar asociaciones de clases:

Cuando se elimina una asociación de **clases**, el **atributo**/la **propiedad** generado automáticamente no se elimina en la clase.

1. Haga clic con el botón derecho en el atributo / en la propiedad de la clase.
2. Seleccione la opción **Eliminar propiedad X de la clase X**.

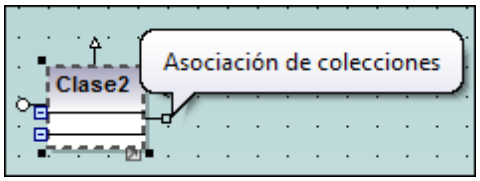
Asociaciones de colección

UModel permite ver las asociaciones de colección de forma especial.

Las asociaciones de colección son asociaciones especiales para las plantillas de colecciones. Las propiedades de una clase (p. ej. una interfaz) se suelen mostrar como asociaciones con el `type` de la propiedad. Pero en UModel este tipo de asociación se puede ver de otra manera

distinta.

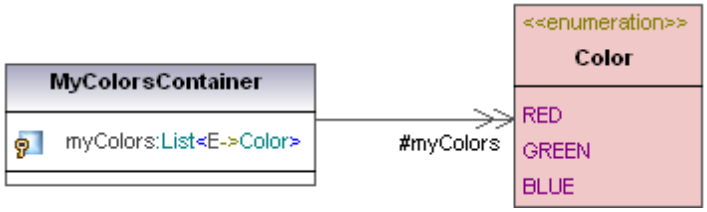
Haga clic en el controlador inferior del lado derecho de la clase y arrastre el conector hasta la clase de destino.



Al soltar el botón del ratón, aparece un menú emergente donde puede seleccionar el tipo de colección:

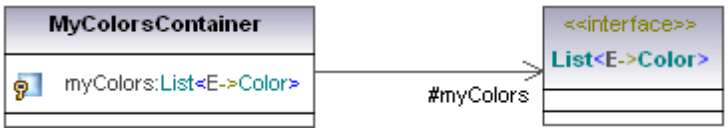
Ahora se crea un tipo distinto de flecha de asociación (con doble punta de flecha).

Ejemplo:
Imagine que durante la ingeniería inversa se crearon asociaciones automáticamente. Si en la pestaña *Edición de diagramas* del cuadro de diálogo "Opciones locales" seleccionó la opción *Resolver asociaciones a colecciones* y si están en el cuadro de diálogo "Plantillas de colecciones", entonces las asociaciones creadas durante la ingeniería inversa aparecerán como `Asociaciones de colección`.



La doble flecha indica que el tipo de **`myColors`** no sólo es **`Color`** sino una colección de varios **`Color`**.
La asociación no aparecerá como **`<E->Color>`**, sino unida directamente a la enumeración **`Color`**, ocultando el hecho de que **`Color`** se utiliza en un enlace de plantilla.

El tipo de colección específico de **`myColors`** todavía se puede ver en **`ColorsContainer`**, pero no en la asociación.



En la pestaña *Edición de diagramas* del cuadro de diálogo "Opciones locales" (**Herramientas | Opciones**) puede indicar en qué plantillas debe tener lugar este comportamiento y si quiere que las colecciones se resuelvan o no.

Chapter 8

Generar documentación UML

8 Generar documentación UML

Sitio web de Altova: Documentación de proyectos UML

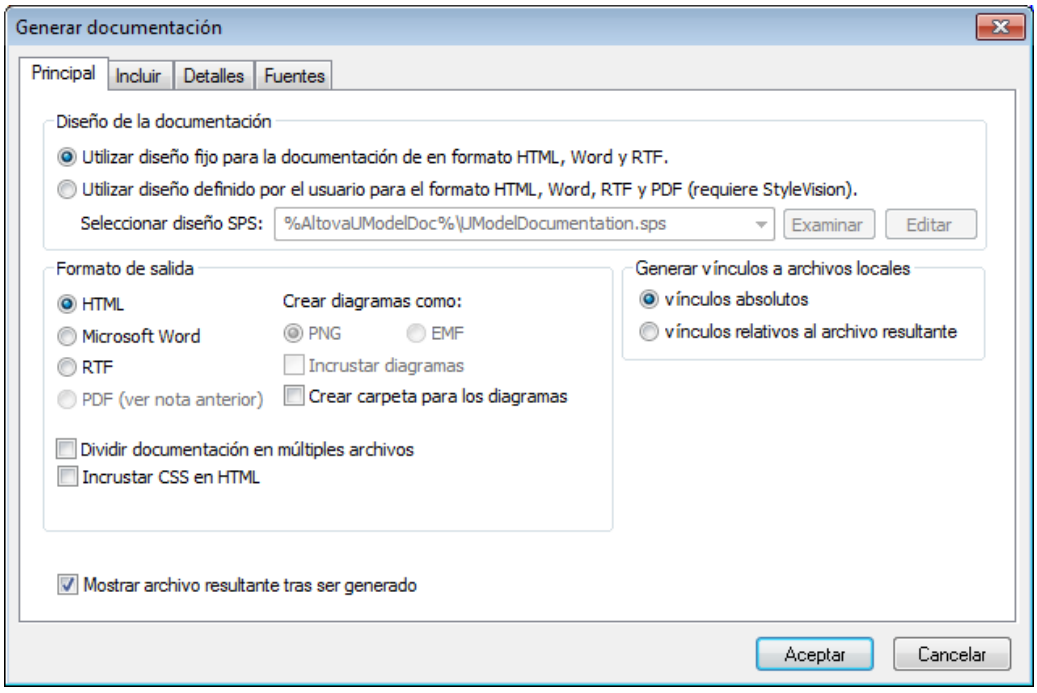
El comando **Generar documentación** genera documentación detallada sobre el proyecto UML en formato HTML, MS Word, RTF y PDF. Una vez generada, la documentación se puede modificar y utilizar sin permiso de Altova.

La documentación abarca los elementos de modelado seleccionados por el usuario en el cuadro de diálogo "Generar documentación". Además, puede generar la documentación con un diseño fijo o usar una hoja de estilos SPS de StyleVision para personalizarla. Consulte el apartado Hojas de estilos definidas por el usuario para más información.

Nota: si quiere usar una hoja de estilos SPS para generar la documentación, debe tener StyleVision instalado en el equipo. Los elementos relacionados se unen por medio de hipervínculos que permiten navegar de un componente a otro. Para generar documentación en formato Word, necesita tener instalado MS Word 2000 (o superior).

Para este ejemplo utilizaremos el proyecto de muestra `Bank_MultiLanguage.ump`:

1. Abra el proyecto en UModel y seleccione la opción de menú **Proyecto | Generar documentación**.
   Esto abre el cuadro de diálogo "Generar documentación" (*imagen siguiente*) en su configuración predeterminada.



Recuerde que también puede generar documentación de determinados elementos de modelado por separado. Esto se consigue haciendo clic con el botón derecho en uno o varios elementos en la Estructura del modelo y seleccionando **Generar documentación** en el menú contextual. El elemento puede ser una carpeta, una clase, una interfaz, etc. En cualquier caso, las opciones de

documentación son las mismas en ambos casos.

Los elementos relacionados se unen por medio de hipervínculos que permiten navegar por la documentación de un componente a otro. En la documentación también aparecen todos los hipervínculos creados a mano.

Nota: también se genera documentación para los subproyectos (perfiles) C#, Java y VB **incluidos**, si estos se habilitan en la pantalla *Incluir* del cuadro de diálogo "Generar documentación".

Pestaña *Principal*
A continuación describimos las opciones de esta pestaña.

Diseño de la documentación

- *Utilizar diseño fijo para....* seleccione esta opción si quiere usar la plantilla integrada para generar la documentación.
- *Utilizar diseño definido por el usuario...* seleccione esta opción si quiere usar una hoja de estilos de StyleVision predefinida. Los archivos SPS están disponibles en la carpeta `...\Mis Documentos\Altova\UModel2017\Documentation\UModel\`.
- **Examinar:** haga clic en este botón para buscar el archivo SPS predefinido.
- **Editar:** haga clic en este botón para abrir el SPS seleccionado en StyleVision.

Formato de salida:

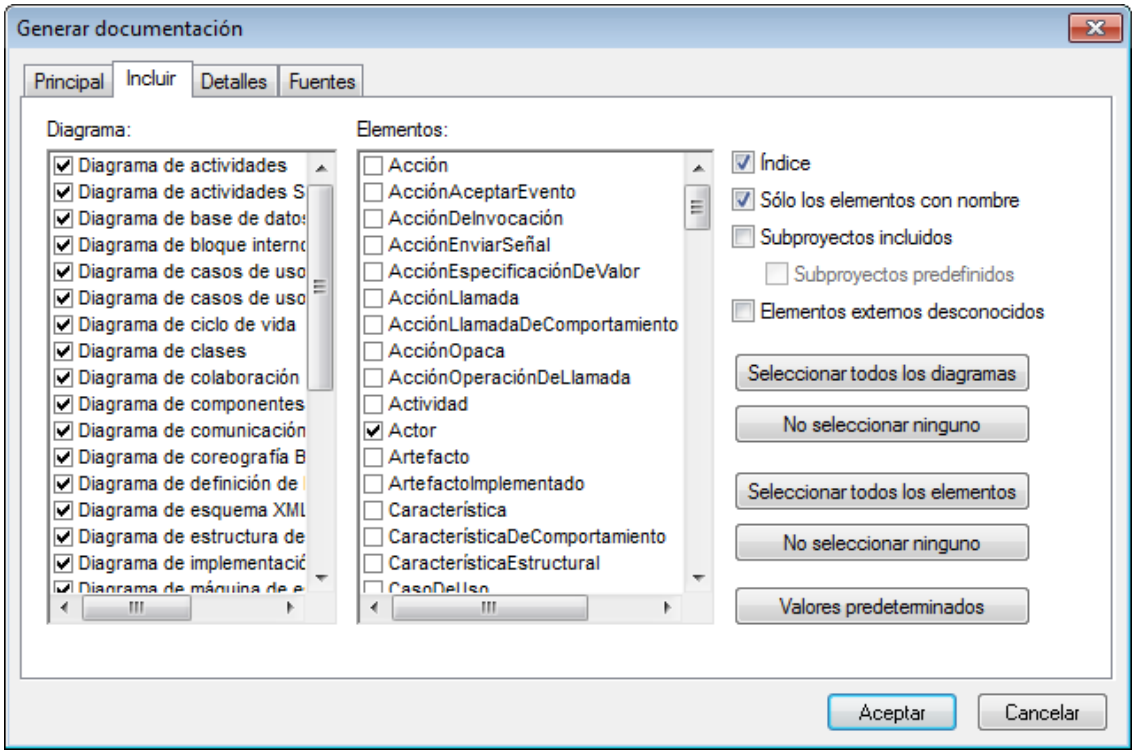
- Aquí se selecciona el formato de salida: *HTML*, *Word*, *RTF* o *PDF*.
  Los documentos de **Microsoft Word** se crean con la extensión de archivo **`.doc`** si utiliza un diseño fijo y con la extensión **`.docx`** si utiliza un archivo SPS.
  El formato **PDF** solamente está disponible si usa un archivo SPS para generar la documentación.
- *Dividir documentación en múltiples archivos:* marque esta casilla si quiere que UModel genere un archivo de salida por cada elemento de modelado de la tabla de contenido (por ejemplo, si hay una clase llamada `C1` que tiene una clase anidada llamada `CNest`, el archivo `C1.html` contiene toda la información sobre `C1` y `CNest`, así como sus atributos, propiedades, etc.).
- *Incrustar CSS en HTML:* marque esta casilla para incrustar un archivo CSS ya existente en la documentación HTML. Si quiere que la referencia al archivo y el archivo CSS propiamente dicho sean externos, desactive esta casilla.
- *Incrustar diagramas:* esta casilla se habilita si el formato de salida elegido es *Microsoft Word* o *RTF*. Si marca esta casilla, los diagramas se incrustan en el archivo generado. Los diagramas se crean como archivo PNG (para HTML) o como archivos PNG/EMF (para MS Word y RTF) y en el archivo de resultado se presentan a través de vínculos de objeto.
- *Crear carpeta para los diagramas:* seleccione esta opción para crear una subcarpeta dentro de la carpeta de salida seleccionada. En esta subcarpeta se guardan todos los diagramas.
- *Mostrar archivo resultante tras ser generado*: esta opción está disponible con todos los formatos de salida. Selecciónela para ver los archivos resultantes en la vista Explorador (para HTML), en MS Word (para formato Word) o en la aplicación predeterminada para archivos RTF y PDF (para RTF y PDF respectivamente).

Generar vínculos a archivos locales

En este grupo de opciones puede indicar si los vínculos generados deben ser absolutos o relativos al archivo de salida.

Pestaña *Incluir*

En esta pestaña (*imagen siguiente*) puede seleccionar qué diagramas y elementos de modelado debe abarcar la documentación.

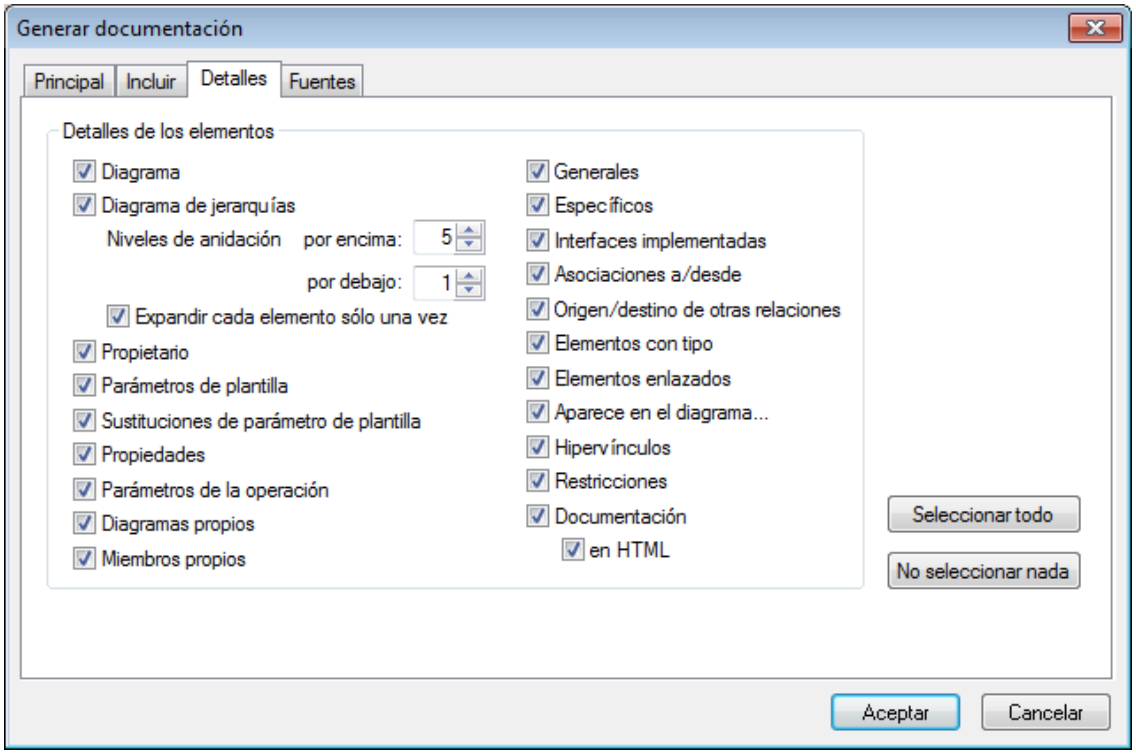


Recuerde que puede deshabilitar la documentación de subproyectos desactivando la casilla *Subproyectos incluidos*.

Pestaña *Detalles*

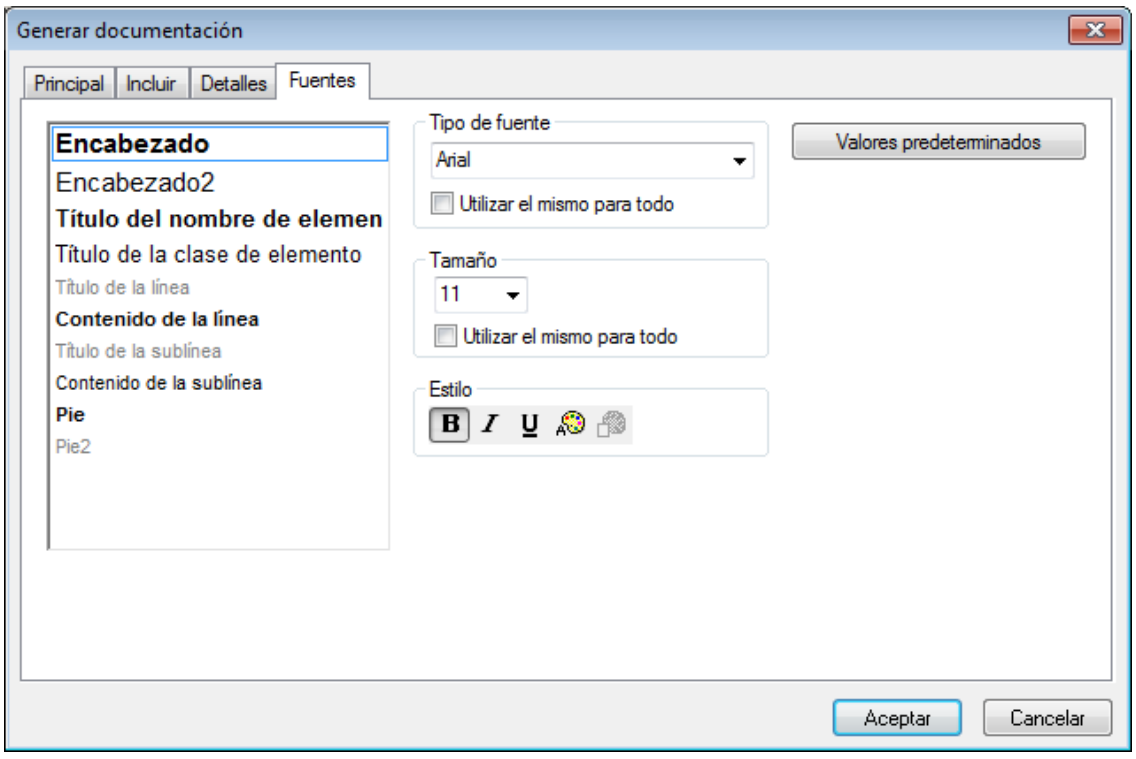
En esta pestaña (*imagen siguiente*) puede seleccionar los detalles de los elementos que se deben incluir en la documentación.

- Si tiene pensado importar el texto de **etiquetas XML** en la documentación, desactive la casilla *en HTML* situada debajo de la opción *Documentación*.
- En los cuadros combinados *por encima* y *por debajo* puede definir cuántos niveles se debe anidar la clase actual en el diagrama de jerarquías.
- Marque la casilla *Expandir cada elemento solo una vez* si quiere que solo un clasificador de cada tipo se pueda expandir en la misma imagen/en el mismo diagrama.



Pestaña *Fuentes*

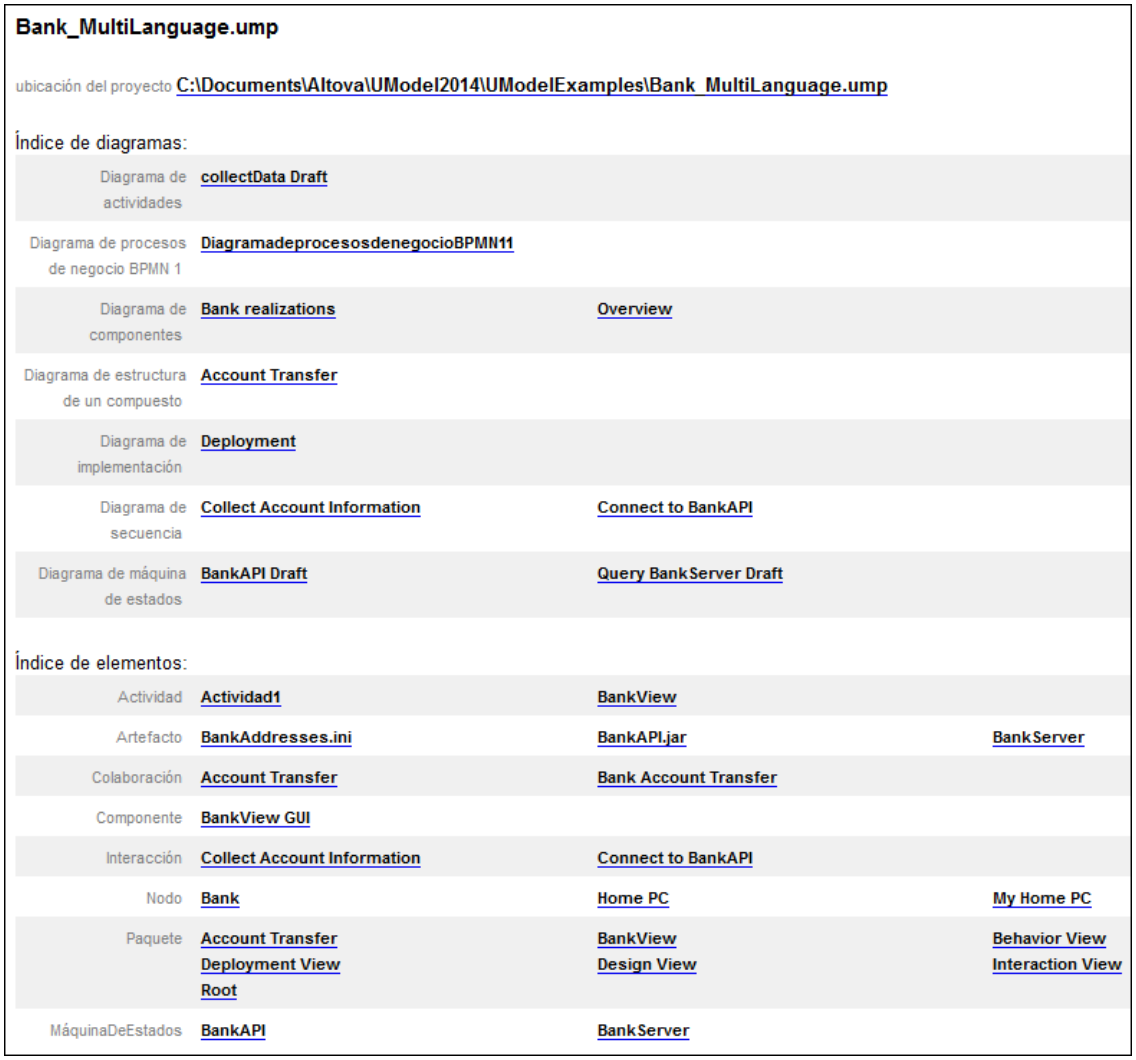
En esta pestaña (*imagen siguiente*) puede configurar la fuente utilizada para el contenido de texto y para los títulos y encabezados de la documentación.



Las imágenes que aparecen a continuación muestran la documentación generada para el archivo `Bank_MultiLanguage.ump` (directorio `...\UModelExamples`) utilizando el diseño fijo.

La primera imagen muestra la documentación generada con los índices de diagramas y de elementos al principio del archivo HTML.

La segunda imagen muestra los detalles de la clase **`Account`** y sus relaciones con otras clases. Observe que los atributos y propiedades de los diagramas de clases también están unidos a sus definiciones por medio de hipervínculos. Haga clic en una propiedad para ver su definición. Las clases de la jerarquía también incluyen hipervínculos (así como el texto subrayado).



Bank_MultiLanguage.ump

ubicación del proyecto **C:\Documents\Altova\UModel2014\UModelExamples\Bank_MultiLanguage.ump**

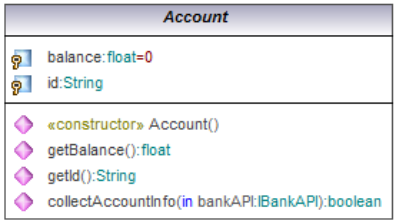
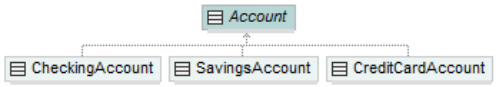
Índice de diagramas:

Diagrama de actividades	collectData Draft		
Diagrama de procesos de negocio BPMN 1	DiagramadeprocesosdenegocioBPMN11		
Diagrama de componentes	Bank realizations	Overview	
Diagrama de estructura de un compuesto	Account Transfer		
Diagrama de implementación	Deployment		
Diagrama de secuencia	Collect Account Information	Connect to BankAPI	
Diagrama de máquina de estados	BankAPI Draft	Query BankServer Draft	

Índice de elementos:

Actividad	Actividad1	BankView	
Artefacto	BankAddresses.ini	BankAPI.jar	BankServer
Colaboración	Account Transfer	Bank Account Transfer	
Componente	BankView GUI		
Interacción	Collect Account Information	Connect to BankAPI	
Nodo	Bank	Home PC	My Home PC
Paquete	Account Transfer Deployment View Root	BankView Design View	Behavior View Interaction View
MáquinaDeEstados	BankAPI	BankServer	

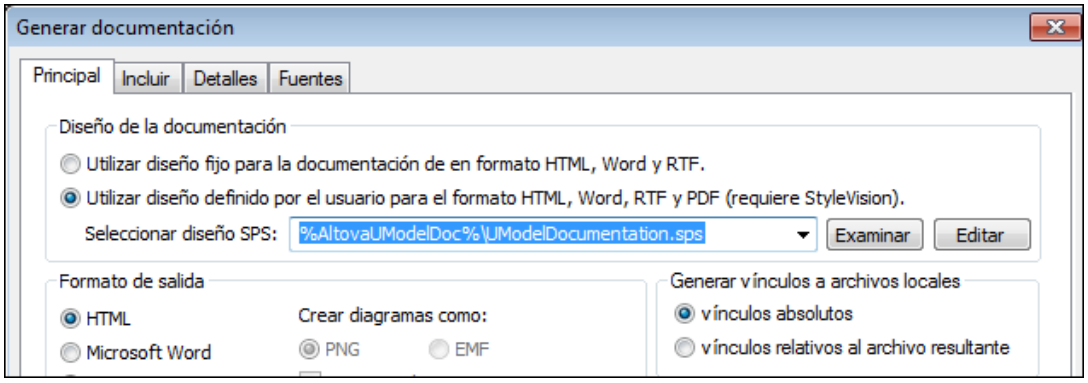
Clase **Account**

diagrama	
jerarquía	
propietario	**bankview**
propiedades	nombre completo **Design View::BankView::com::altova::bankview::Account** nivel de acceso **public** leaf **false** abstract **true** isFinalSpecialization **false** activo **false** nombre del archivo de código **Account.java** ruta de acceso del archivo de código **C:\Documents\Altova\UModel2014\UModelExamples\Bank_MultiLanguage_Java\Src\Account.java** «annotations» **false** «static» **false** «strictfp» **false**
miembroPropio	**Account balance collectAccountInfo getBalance getId id**
específico	**CheckingAccount CreditCardAccount SavingsAccount**
asociaciones desde	desde Nombre de la asociación **Bank (accounts)**
destino de la relación	RealizaciónDeComponente **BankView**
elementosConTipo	Clase **Bank** Propiedad **accounts** Interacción **Collect Account Information** Propiedad **b**
aparece en el diagrama	**BankView Main Hierarchy of Account**

8.1 Con una hoja de estilos SPS predeterminada

Para generar documentación con la hojas de estilo SPS predeterminada:

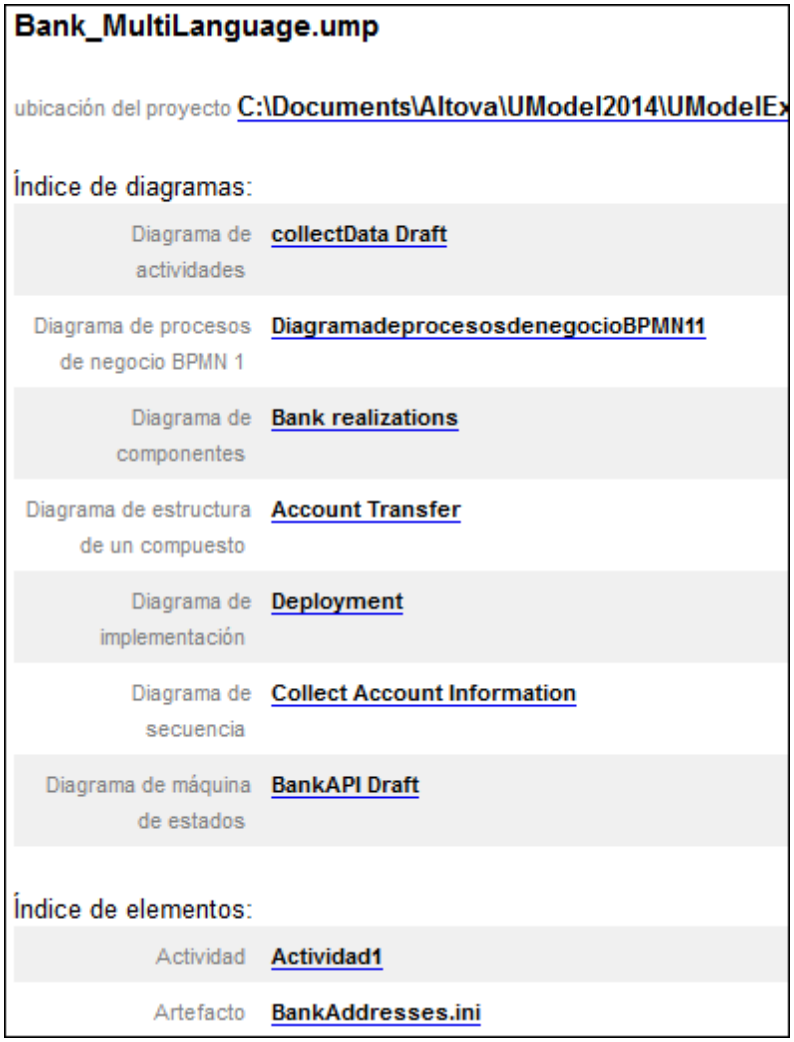
1. Seleccione el comando de menú **Proyecto | Generar documentación.**
2. En el cuadro de diálogo "Generar documentación" seleccione el botón de opción *Utilizar diseño definido por el usuario...* (en la pestaña *Principal*).
3. Seleccione la hoja de estilos `UModelDocumentation.sps` (si no está seleccionada todavía).



Aparece un aviso pidiendo que guarde el archivo.

4. Escriba el nombre del archivo y seleccione la ubicación donde desea guardarlo.

En la documentación generada (*imagen siguiente*) haga clic en un vínculo para navegar al elemento de modelado.



Bank_MultiLanguage.ump

ubicación del proyecto **C:\Documents\Altova\UModel2014\UModelEx**

Índice de diagramas:

Diagrama de actividades	**collectData Draft**
Diagrama de procesos de negocio BPMN 1	**DiagramadeprocesosdenegocioBPMN11**
Diagrama de componentes	**Bank realizations**
Diagrama de estructura de un compuesto	**Account Transfer**
Diagrama de implementación	**Deployment**
Diagrama de secuencia	**Collect Account Information**
Diagrama de máquina de estados	**BankAPI Draft**

Índice de elementos:

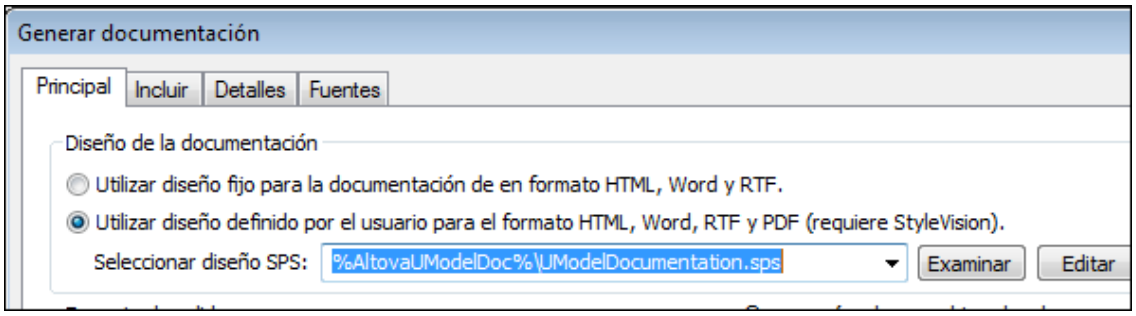
Actividad	**Actividad1**
Artefacto	**BankAddresses.ini**

8.2 Con hojas de estilos predefinidas por el usuario

Si lo prefiere, en lugar de usar el diseño fijo o la hoja de estilos SPS predeterminada, puede crear un diseño personal para su documentación UML. El diseño se crea en una hoja de estilos SPS de StyleVision. Recuerde que también puede editar la hoja de estilos SPS que viene con UModel.

Seleccionar el diseño SPS para generar la documentación

Debe especificar qué diseño SPS se utiliza para generar la documentación en el cuadro de diálogo "Generar documentación" (**Proyecto | Generar documentación**). Seleccione el botón de opción *Utilizar diseño definido por el usuario...* y después seleccione un archivo SPS en el cuadro combinado. Si el SPS no está en el cuadro combinado, haga clic en el botón **Examinar** para buscarlo. El SPS predeterminado disponible en este cuadro combinado es `UModelDocumentation.sps`, que está en la carpeta `...\Documentation\UModel`.



Nota: para poder generar la documentación usando un SPS necesita tener StyleVision instalado en el equipo.

¿Cómo se crea una hoja de estilos SPS?

Las hojas de estilos Power Stylesheet de StyleVision (o archivos SPS) se crean con la aplicación Altova StyleVision. El SPS utilizado para generar documentación UML de UModel debe estar basado en un esquema XML que especifique la estructura del documento XML que almacenará la documentación de UModel.

Se trata del esquema `UModelDocumentation.xsd` que viene con el paquete de instalación de UModel y está disponible en la carpeta `...\Mis Documentos\Altova\UModel2017\Documentation\UModel`.

Para crear el diseño SPS en StyleVision, coloque los nodos del esquema `UModelDocumentation.xsd` en el diseño y asígneles estilos y propiedades. Recuerde que el esquema `UModelDocumentation.xsd` **ya incluye** el archivo `Documentation.xsd` situado en la carpeta superior.

También puede añadir al diseño componentes adicionales, como vínculos e imágenes. Para más información consulte el *Manual del usuario de Altova StyleVision*.

La ventaja de usar diseños SPS para generar la documentación es que permite un mayor control sobre el diseño. Además recuerde que el formato de salida PDF solamente está disponible si utiliza un diseño SPS para generar la documentación. En otras palabras, la documentación no se puede generar en formato PDF si usa el diseño fijo.

Chapter 9

Diagramas UML

9 Diagramas UML

Hay dos grandes tipos de diagramas UML: (i) los diagramas de estructura, que muestran la vista estática del modelo, y (ii) los diagramas de comportamiento, que muestran su vista dinámica. UModel es compatible con los 14 tipos de diagramas de la especificación UML 2.4 y con los diagramas de esquema XML.

- Diagramas de comportamiento: actividades, máquina de estados, máquina de estados de protocolos, casos de uso, interacción, comunicación, interacción global, secuencia y ciclo de vida.
- Diagramas de estructura: clases, estructura de un compuesto, componentes, implementación, objetos y paquetes.
- Otros diagramas: diagramas de esquema XML.

Nota: en la mayoría de los diagramas de modelado puede pulsar **Ctrl+Entrar** para crear una línea nueva en el nombre de algunos elementos (por ejemplo, en las etiquetas de las líneas de vida de los diagramas de secuencia y de ciclo de vida o en las condiciones de protección, nombres de estado y nombres de actividades).

9.1 Diagramas de comportamiento

Este tipo de diagramas ilustran las características de la conducta de un sistema o proceso de negocio e incluye un subconjunto de diagramas que subrayan cómo interactúan los objetos.

Diagramas de comportamiento

- Diagrama de actividades
- Diagrama de máquina de estados
- Diagrama de máquina de estados de protocolos
- Diagrama de casos de uso

El subconjunto de diagramas de comportamiento que ilustran cómo interactúan los objetos está compuesto por estos diagramas:

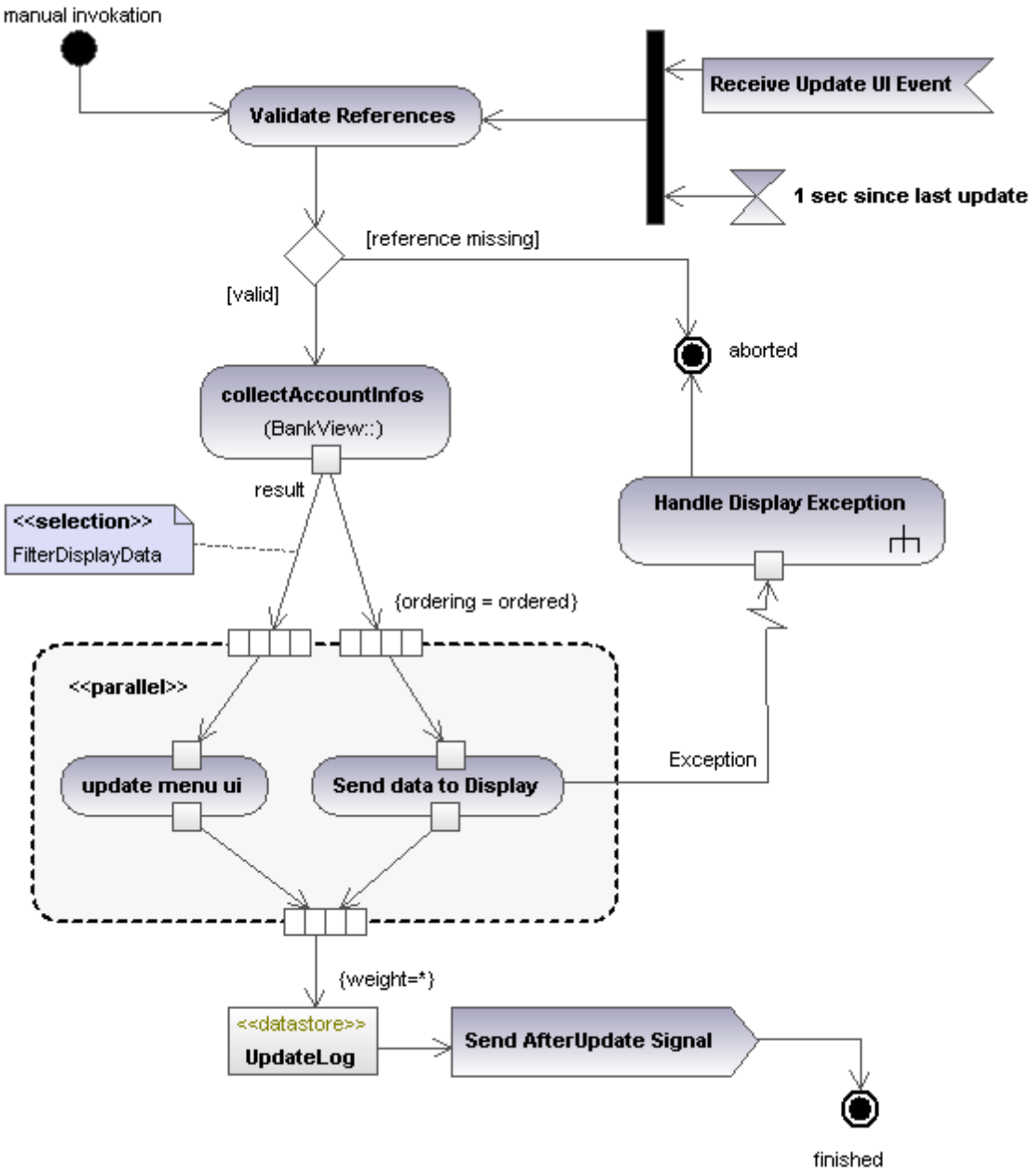
- Diagrama de comunicación
- Diagrama global de interacción
- Diagrama de secuencia
- Diagrama de ciclo de vida

9.1.1 Diagrama de actividades

Sitio web de Altova: Diagramas de actividades UML

Los diagramas de actividades sirven para modelar flujos de trabajo de procesos de negocios. Permiten ver qué acciones deben tener lugar y qué dependencias de comportamiento existen. Los diagramas de actividades describen el orden concreto de las actividades y permiten un procesamiento tanto condicional como en paralelo. Los diagramas de actividades son una especie de diagrama de estados, con actividades en lugar de estados.

Nota: el diagrama de actividades que aparece a continuación está disponible en el ejemplo `Bank_MultiLanguage.ump`, en la carpeta `...\UModelExamples`.



9.1.1.1 Insertar elementos

Iconos de las barras de herramientas:

1. Haga clic en el icono pertinente de la barra de herramientas Diagrama de actividades.
2. Ahora haga clic en el área de trabajo del diagrama para insertar el elemento. Si mantiene pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo podrá insertar varios elementos del tipo seleccionado.

Arrastrar elementos desde la Estructura del modelo hasta el diagrama:

1. En la Estructura del modelo busque el elemento que quiere insertar en el otro diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).
2. Arrastre el elemento hasta el diagrama de actividades.

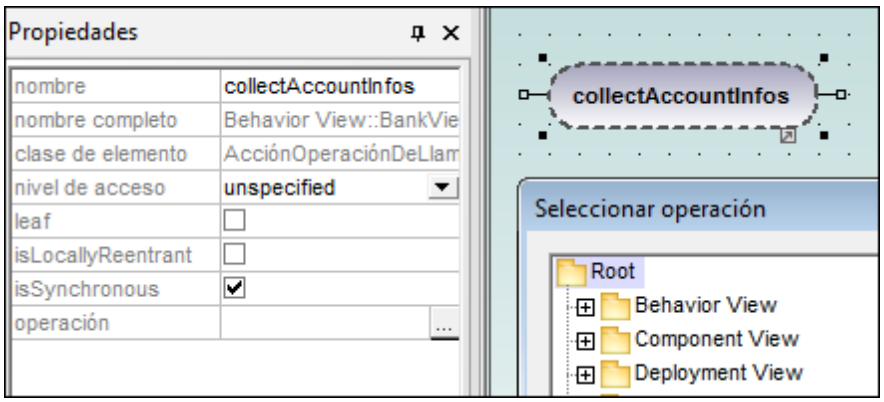
Insertar una acción (ComportamientoDeLlamada):

1. Haga clic en el icono **Acción (ComportamientoDeLlamada)** de la barra de herramientas y haga clic en el diagrama de actividades para insertar la acción.
2. Escriba el nombre de la Acción (p. ej. `Validar referencias`) y pulse **Entrar** para confirmar.

Nota: pulse **Ctrl+Entrar** para crear una línea nueva en el nombre de la acción.

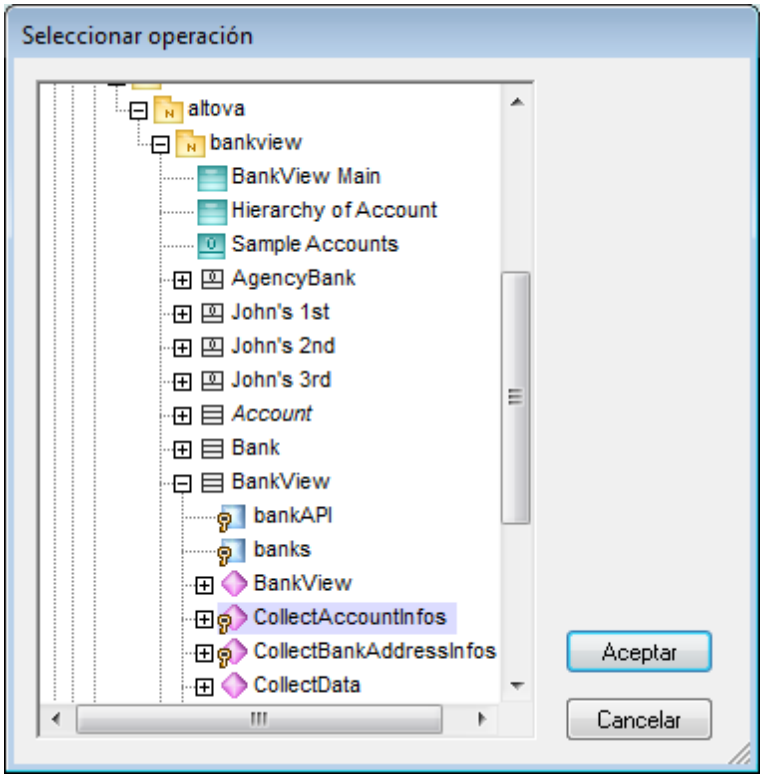
Insertar una acción (OperaciónDeLlamada) y seleccionar una operación determinada:

1. Haga clic en el icono **Acción (OperaciónDeLlamada)** de la barra de herramientas y haga clic en el diagrama de actividades para insertar la acción.
2. Escriba el nombre de la Acción (p. ej. `collectAccountInfos`) y pulse **Entrar** para confirmar.
3. En el panel Propiedades haga clic en el botón **Examinar** del campo `operation`.

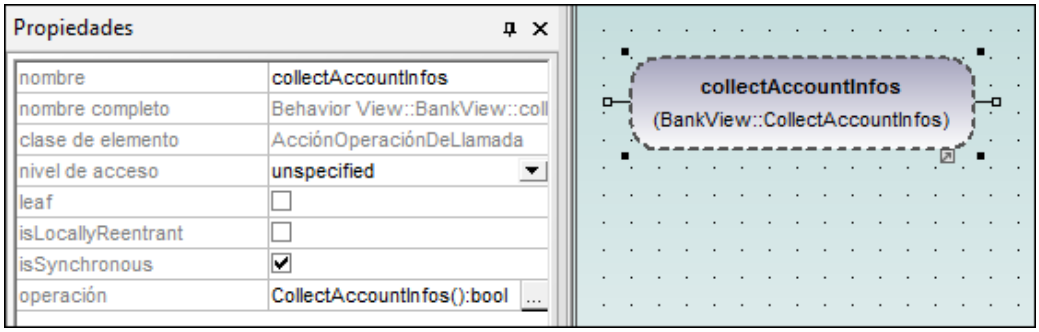


Esto abre el cuadro de diálogo "Seleccionar operación", donde puede seleccionar una operación determinada.

4. Navegue hasta la operación que desea insertar y haga clic en **Aceptar** para confirmar.



Para este ejemplo seleccionamos la operación `collectAccountInfos` de la clase **`BankView`**.



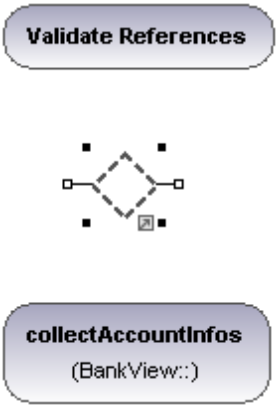
9.1.1.2 Crear bifurcaciones y convergencias

Crear una rama (flujo alterno)
Una rama tiene un flujo de entrada y varios flujos de salida protegidos por guardas. Solo se puede recorrer uno de esos flujos de salida, así que los guardas deben excluirse mutuamente.

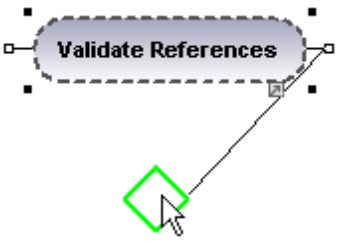
En el ejemplo que utilizamos a continuación vamos a validar las referencias de **`BankView`**:

- la rama1 tiene el guarda `reference missing`, que pasa a la actividad `abort`.
- la rama2 tiene el guarda `valid`, que pasa a la actividad `collectAccountInfos`.

1. Haga clic en el icono **NodoDeDecisión**  de la barra de herramientas y haga clic en el área de trabajo del diagrama de actividades.

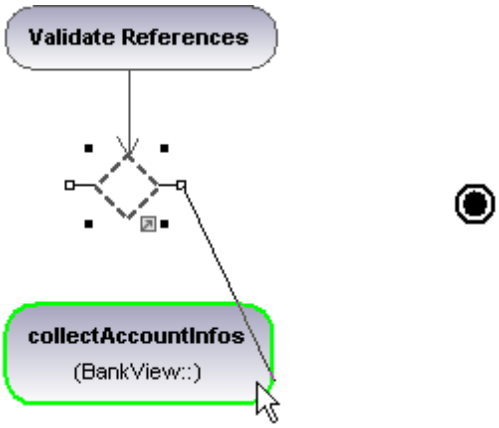


2. Haga clic en el icono **NodoFinalDeActividad**  , que representa la actividad `abort`, e insértelo en el diagrama de actividades.
3. Haga clic en la actividad `Validate References` y después haga clic en su conector derecho (el controlador `FlujoDeControl`). Ahora arrastre el conector hasta el elemento `NodoDeDecisión`.

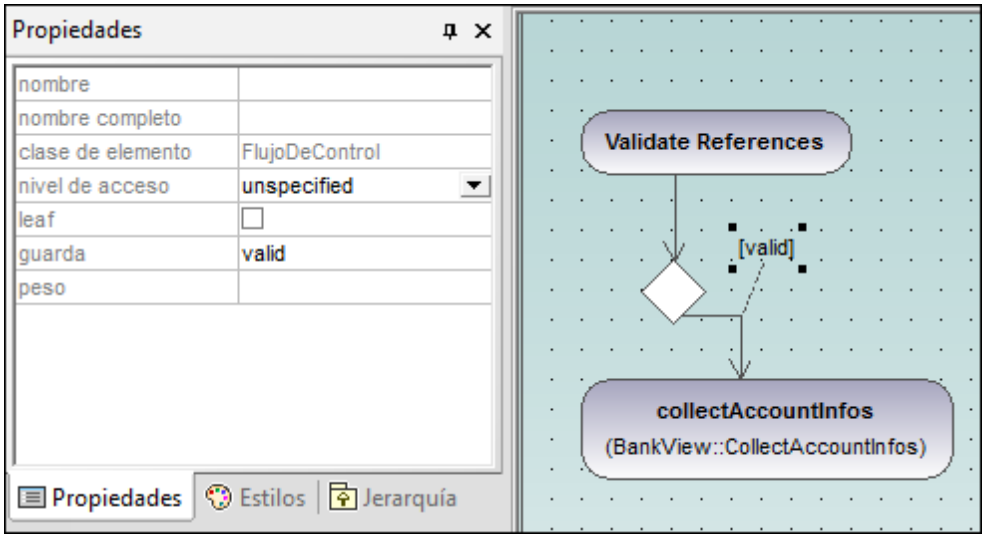


El elemento se resalta cuando sea posible colocar el conector.

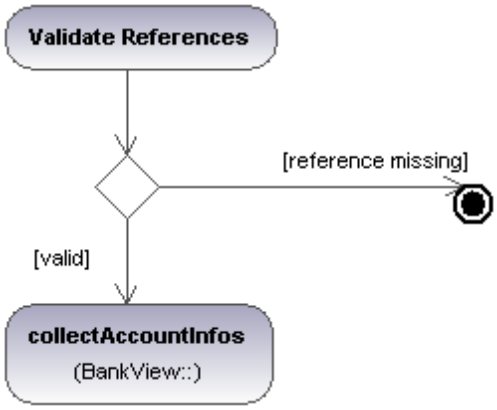
4. Haga clic en el elemento `NodoDeDecisión` y después en su conector derecho (el controlador `FlujoDeControl`). Arrástrelo hasta la acción `collectAccountInfos`. Consulte el apartado Insertar una acción (OperaciónDeLlamada) para obtener más información.



5. En el panel Propiedades seleccione el valor `valid` para la propiedad `guarda`.



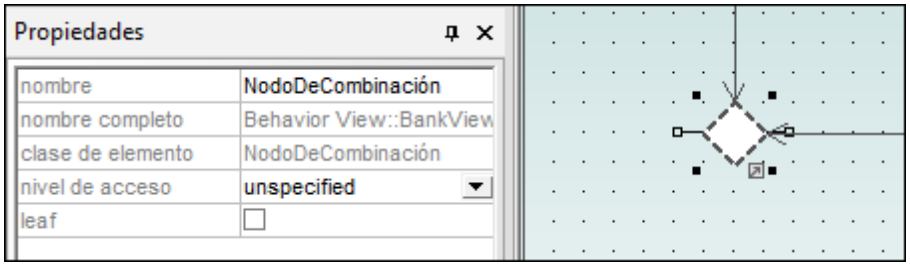
6. Haga clic en el elemento **`NodoDeDecisión`** y después en su conector derecho (el controlador `FlujoDeControl`). Arrástrelo hasta el elemento **`NodoFinalDeActividad`**. La condición de guarda de esta transición se define automáticamente como "else". Haga doble clic en la condición de guarda del diagrama para cambiarla por "reference missing".



Nota: recuerde que UModel no valida ni revisa el número flujos de control/objetos del diagrama.

Crear una combinación

1. Haga clic en el icono **NodoDeCombinación**  de la barra de herramientas y después haga clic en el diagrama para insertarlo.

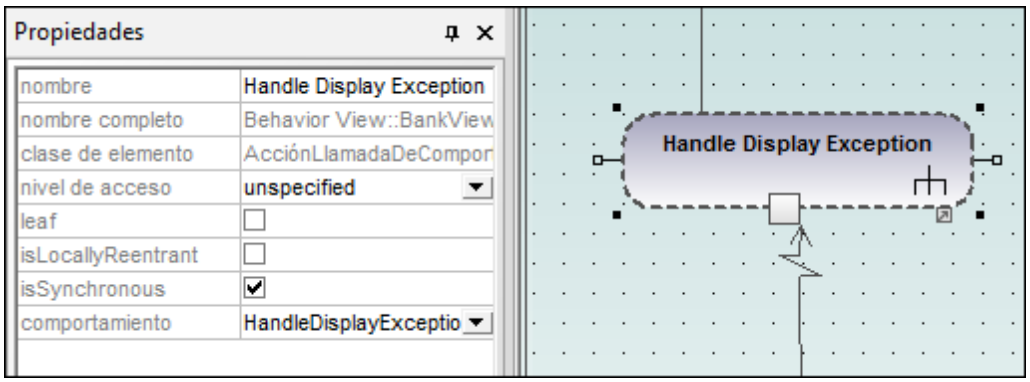


2. Haga clic en el conector `FlujoDeControl` (`FlujoDeObjetos`) de las acciones que desea combinar y arrástrelas hasta el símbolo del `NodoDeCombinación`.

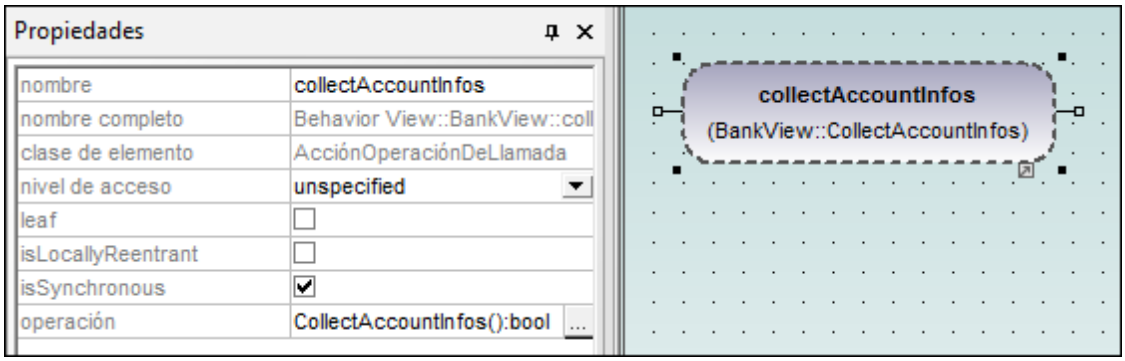
9.1.1.3 Elementos

Acción (ComportamientoDeLlamada)
Inserta el elemento **`AcciónComportamientoDeLlamada`**, que invoca un comportamiento concreto directamente.

Si selecciona un comportamiento ya existente desde el cuadro combinado `comportamiento` (del panel Propiedades), en el elemento aparece un icono en forma de rastrillo.



Acción (OperaciónDeLlamada)
Inserta el elemento **`AcciónOperaciónDeLlamada`**, que invoca un comportamiento concreto como método directamente. Para más información consulte el apartado Insertar una acción (OperaciónDeLlamada).



Acción (AcciónOpaca)
Un tipo de acción utilizada para especificar la información de implementación. Se puede usar como marcador de posición hasta que decida qué tipo de acción desea utilizar.

Acción (AcciónEspecificaciónDeValor)

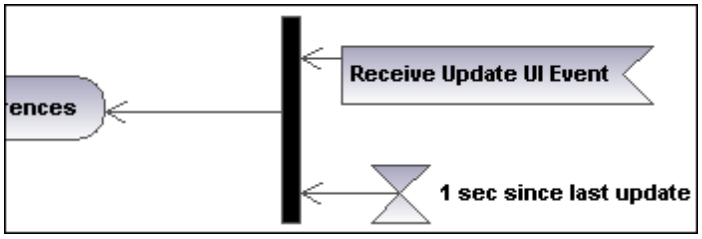
Un tipo de acción que evalúa (o genera) un valor determinado en el pin de salida. Viene definido por las propiedades (p. ej. `upperBound` para el límite superior.)

AcciónAceptarEvento

Inserta la acción **`AceptarEvento`**, que espera que tenga lugar un evento que cumpla determinados requisitos.

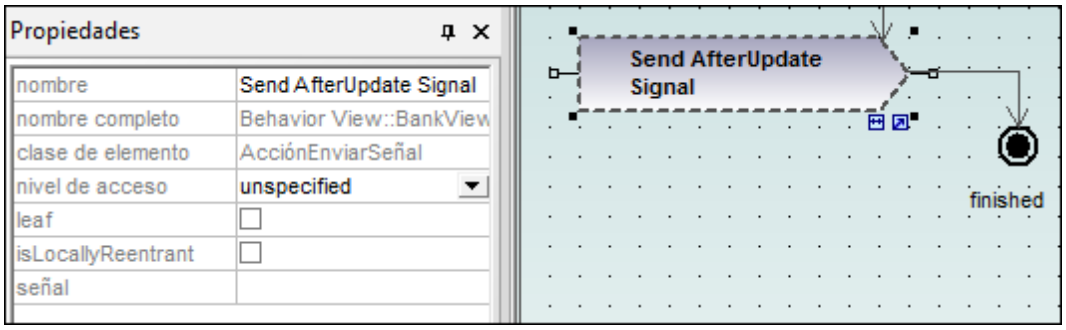
AcciónAceptarEvento (EventoDeTiempo)

Inserta una acción **`AceptarEvento`**, que está desencadenada por un evento de tiempo (un instante de tiempo que viene dado por una expresión).



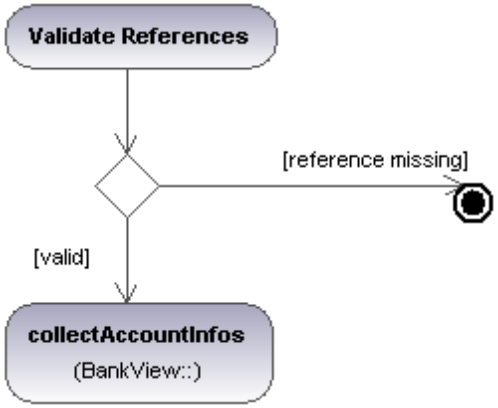
AcciónEnviarSeñal

Inserta la acción **`EnviarSeñal`**, que crea una señal desde sus entradas y transmite la señal al objeto de destino, donde puede provocar la ejecución de una actividad.



NodoDeDecisión

Inserta un **`NodoDeDecisión`**, que tiene una sola transición de entrada y varias transiciones de salida protegidas por guardas. Para más información consulte el apartado Crear ramas.



 **NodoDeCombinación**
Inserta un **`NodoDeCombinación`**, que combina varias transiciones alternas definidas por el **`NodoDeDecisión`**. El **`NodoDeCombinación`** no sincroniza procesos simultáneos, sino que selecciona uno de los procesos.

 **NodoInicial**
Se trata del comienzo del proceso de actividades. Una actividad puede tener varios nodos iniciales.

 **NodoFinalDeActividad**
Se trata del final del proceso de actividades. Una actividad puede tener varios nodos finales y todos los flujos de la actividad se detienen cuando se encuentra el primer nodo final.

 **NodoFinalDeFlujo**
Inserta un **`NodoFinalDeFlujo`**, que termina un flujo pero no los demás flujos de la actividad.

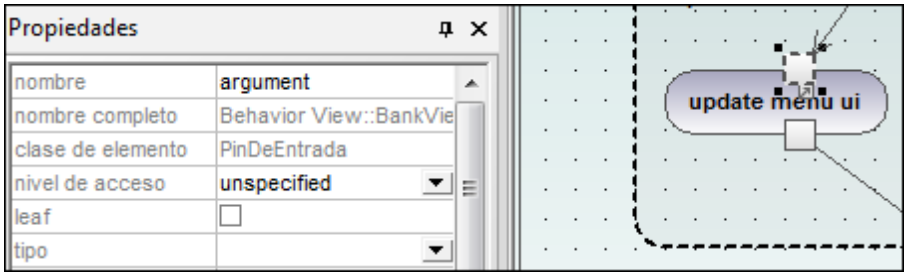
 **NodoDeBifurcación**
Inserta un nodo de bifurcación vertical. Sirve para dividir los flujos en varios flujos simultáneos.

 **NodoDeBifurcación (Horizontal)**
Inserta un nodo de bifurcación horizontal. Sirve para dividir flujos en varios flujos simultáneos.

 **NodoDeReunión**
Inserta un nodo de reunión vertical. Sirve para sincronizar varios flujos definidos por un nodo de bifurcación.

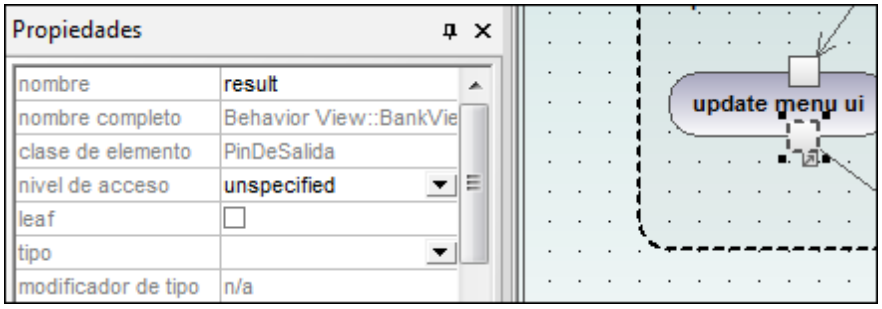
NodoDeReunión (horizontal)
Inserta un nodo de reunión horizontal. Sirve para sincronizar varios flujos definidos por un nodo de bifurcación.

PinDeEntrada
Inserta un pin de entrada en un **ComportamientoDeLlamada** o en una **OperaciónDeLlamada**. Los pins de entrada aportan los valores de entrada que utiliza la acción. A los pins de entrada se les asigna un nombre predeterminado (argumento) de forma automática.



Si al arrastrar el pin de entrada sobre un elemento el puntero se convierte en este símbolo , significa que puede colocar el pin de entrada en el elemento.

PinDeSalida
Inserta un pin de salida. Los pins de salida contienen los valores de salida que produce una acción. Al pin de salida se le asigna automáticamente un nombre equivalente a la propiedad UML de la acción (p. ej. `result`).



Si al arrastrar el pin de salida sobre un elemento el puntero se convierte en este símbolo , significa que puede colocar el pin de salida en el elemento.

Pin de Excepción
Puede convertir un **PinDeSalida** en un pin de **Excepción** haciendo clic en el pin y seleccionando `esPinDeExcepción` en el panel Propiedades.

PinDeValor
Inserta un pin de valor que es un pin de entrada que aporta un valor a una acción que no viene de

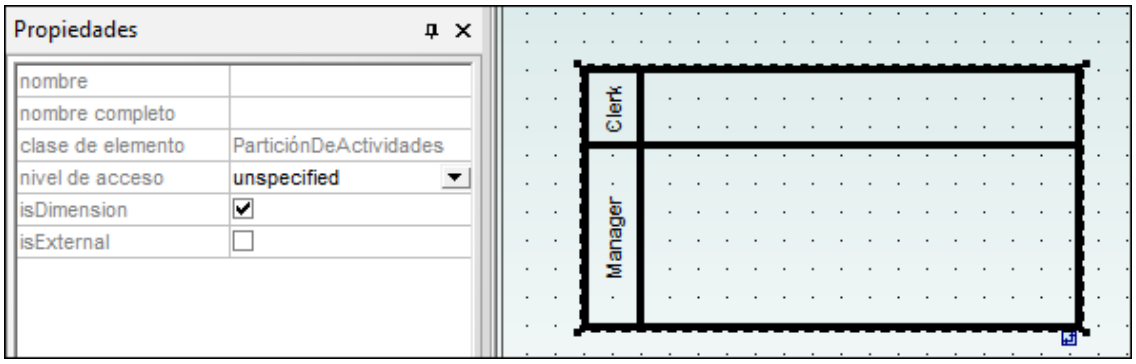
un flujo de objeto de entrada. Se representa con el símbolo de un pin de entrada y tiene las mismas propiedades que un pin de entrada.

NodoDeObjeto
Inserta un nodo de objeto que es un nodo de actividad abstracto que define el flujo de objeto de una actividad. Los nodos de objeto solo contiene valores en tiempo de ejecución que se ajustan al tipo del nodo de objeto.

NodoDeBúferCentral
Inserta un nodo de búfer central que funciona de búfer para varios flujos de entrada y salida de otros nodos de objeto.

NodoAlmacénDeDatos
Inserta un nodo de almacén de datos, un nodo de búfer central especial que almacena datos persistentes (es decir, no transitorios).

ParticiónDeActividades (horizontal)
Inserta una partición de actividades horizontal, un tipo de grupo de actividades que sirve para identificar acciones que tienen características en común. Suelen equivaler a las unidades de organización de los modelos de negocio.

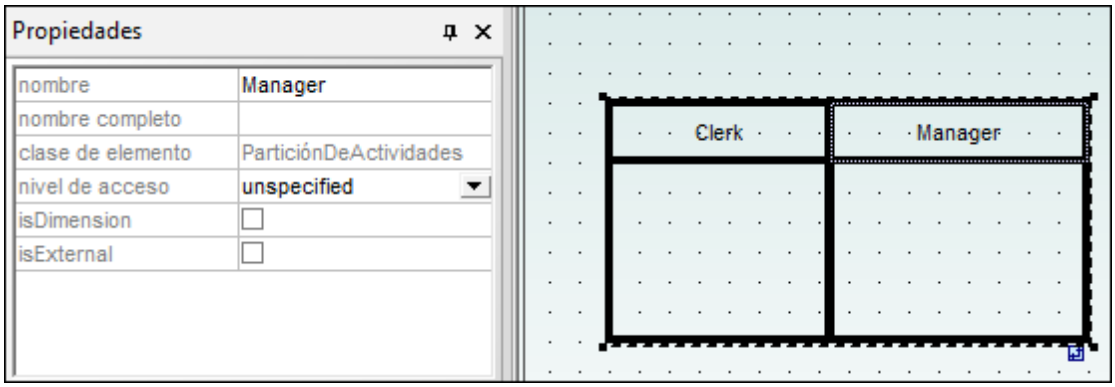


Para editar una etiqueta, haga doble clic en ella. Para orientar el texto correctamente, pulse **Entrar**.

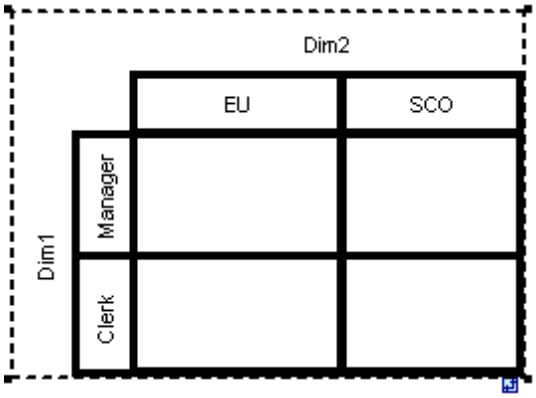
Recuerde que las particiones de actividades de UML 2.0 son el equivalente de los *compartimentos* (`swimlanes`) de las versiones antiguas de UML.

- Los elementos colocados dentro de una **`ParticiónDeActividades`** pasan a formar parte de ella cuando su contorno esté resaltado.
- Los objetos colocados dentro de una **`ParticiónDeActividades`** se pueden seleccionar uno por uno con **Ctrl+clic** o con el recuadro de selección.
- Para mover la **`ParticiónDeActividades`** a otra posición, haga clic en su contorno o en su título y arrástrela.

ParticiónDeActividades (vertical)
Inserta una partición de actividades vertical, un tipo de grupo de actividades que sirve para identificar acciones que tienen características en común. Suelen equivaler a las unidades de organización de los modelos de negocio.

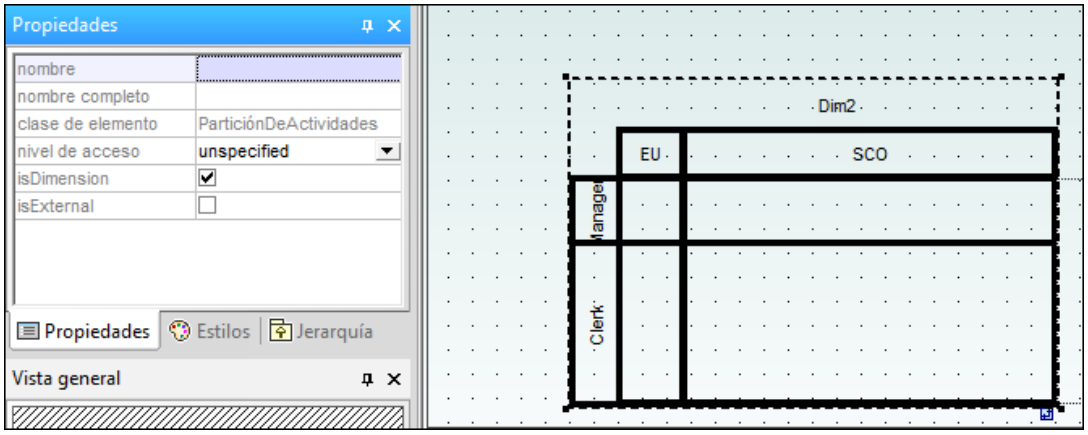


ParticiónDeActividades (2D)
Inserta una partición de actividades bidimensional, un tipo de grupo de actividades que sirve para identificar acciones que tienen características en común. Las etiquetas de los dos ejes son editables.



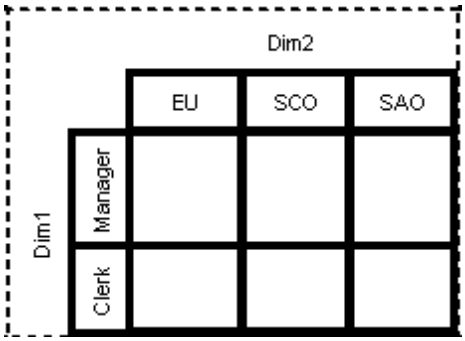
Para quitar las etiquetas de las dimensiones Dim1 y Dim2:

1. Haga clic en la etiqueta que desea eliminar (p. ej. `Dim1`).
2. En el panel Propiedades haga doble clic en la entrada `Dim1`, elimínela y pulse **Entrar** para confirmar.



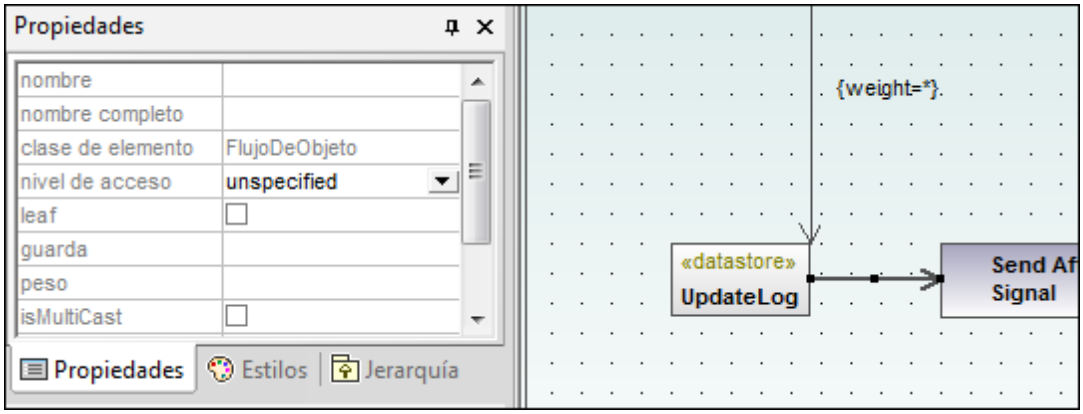
Para anidar particiones de actividades:

1. Haga clic con el botón derecho en la etiqueta de la dimensión donde desea insertar un partición nueva.
2. Seleccione **Nuevo/a | ParticiónDeActividades**.



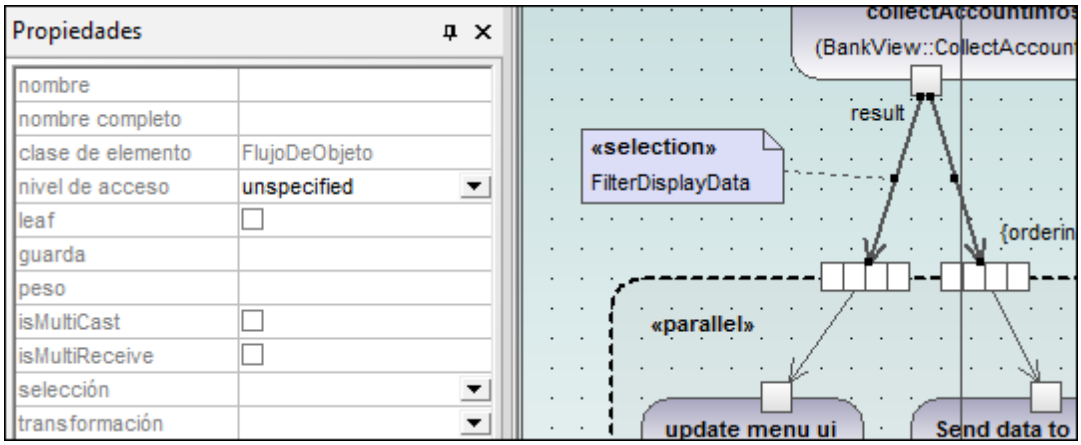
FlujoDeControl

Un flujo de control es una línea con una flecha que conecta dos actividades/comportamientos e inicia una actividad una vez finaliza la actividad anterior.



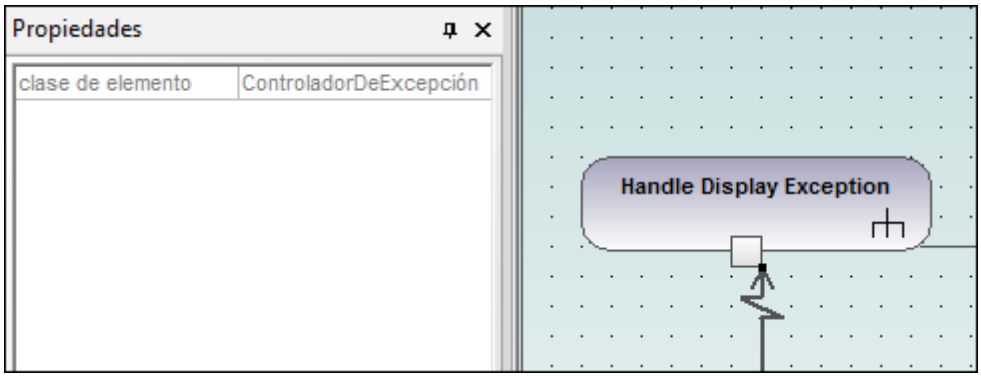
FlujoDeObjeto

Un flujo de objeto es una línea con una flecha que conecta dos acciones/nodos de objeto e inicia una actividad una vez finaliza la actividad anterior. Los objetos y datos se pueden pasar a través del flujo de objeto.



ControladorDeExcepción

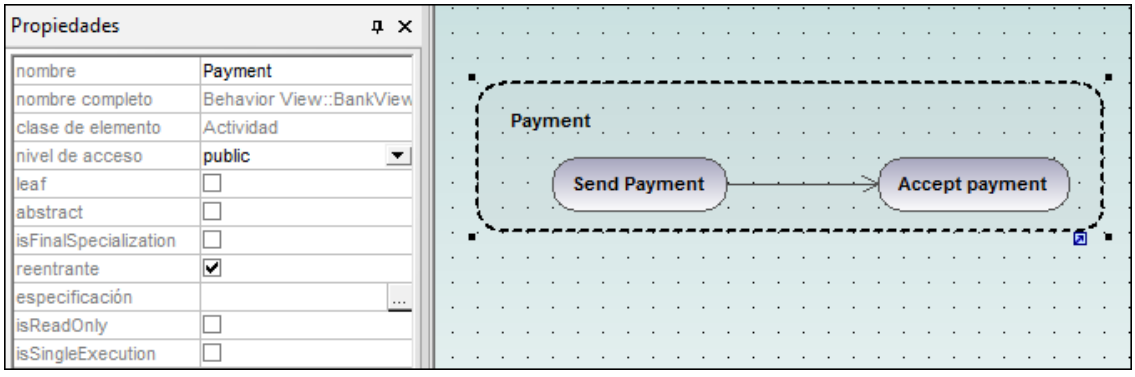
Un controlador de excepción es un elemento que especifica qué acción debe ejecutarse si se genera determinada excepción durante la ejecución del nodo protegido.



Los controladores de excepción solo se pueden colocar en el pin de entrada de una acción.

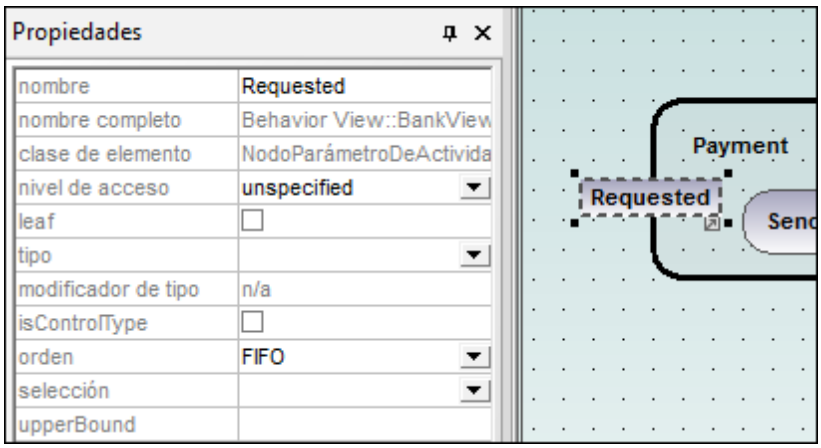
Actividad

Inserta una actividad en el diagrama de actividades.



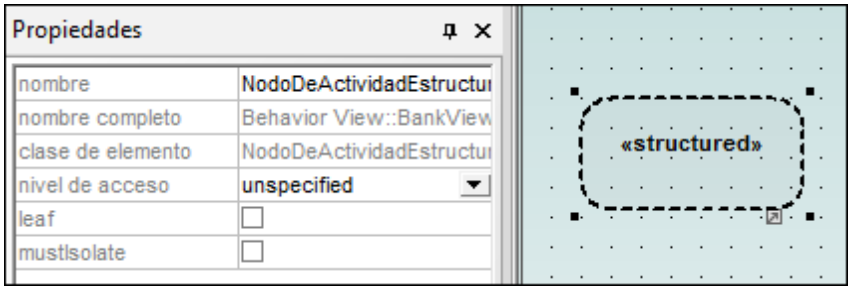
NodoParámetroDeActividad

Inserta un nodo parámetro de actividad en una actividad. Al hacer clic en la actividad se inserta el nodo parámetro en el contorno de la actividad.



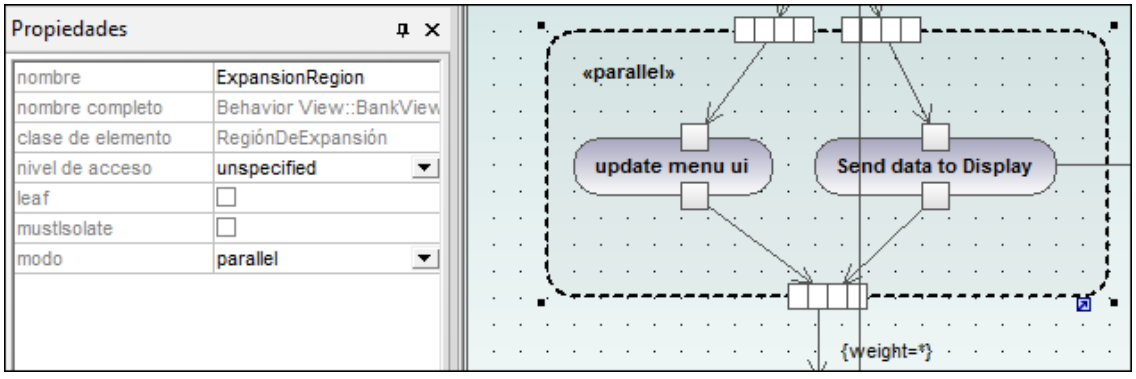
NodoDeActividadEstructurada

Inserta un nodo de actividad estructurada, que es una parte estructurada de la actividad que no se comparte con ningún otro nodo estructurado.



RegiónDeExpansión

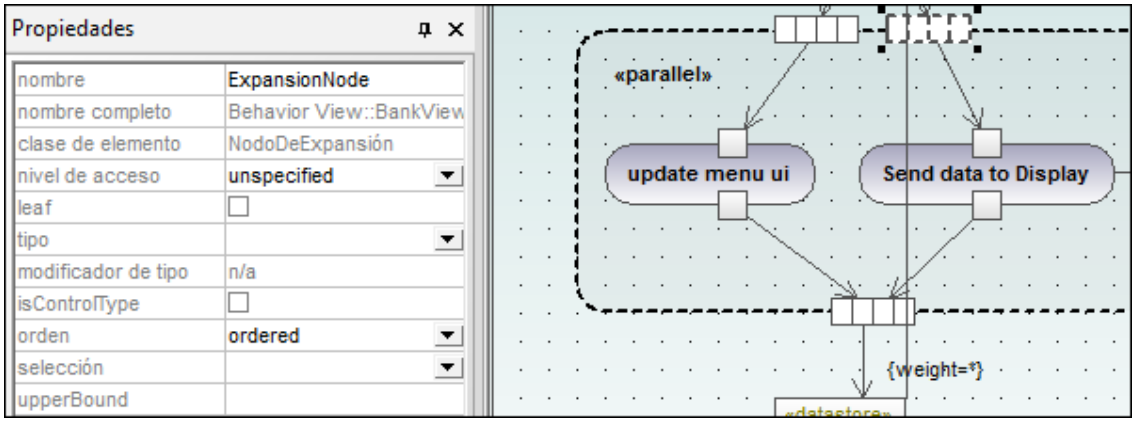
Una región de expansión es una región de una actividad que tiene entradas y salidas explícitas (usando **`NodosDeExpansión`**). Cada entrada es una colección de valores.



El modo región de expansión aparece como palabra clave y para cambiarlo basta con hacer clic en el cuadro combinado `modo` del panel Propiedades. Las opciones disponibles son: `parallel`, `iterative` o `stream`.

NodoDeExpansión

Inserta un nodo de expansión en una región de expansión. Los nodos de expansión son nodos de entrada y salida para la región de expansión, donde cada entrada/salida es una colección de valores. Las flechas que entran y salen de la región de expansión determinan el tipo concreto de nodo de expansión.

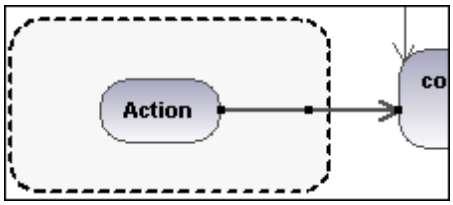


RegiónDeActividadInterrumpible

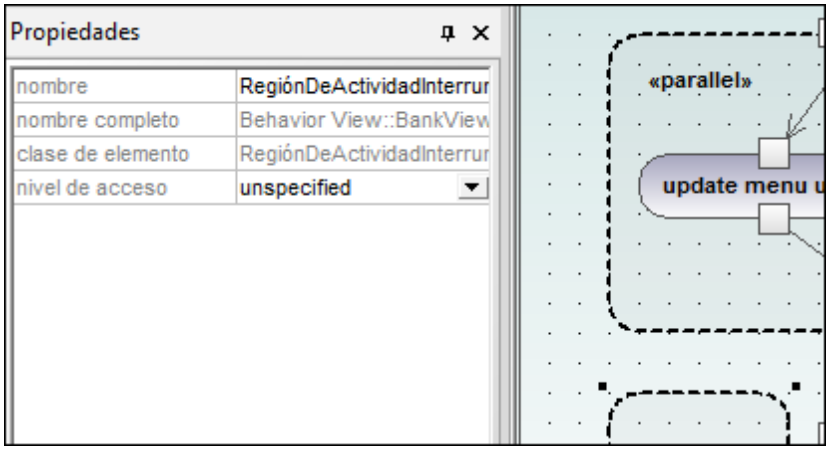
Una región interrumpible contiene nodos de actividad. Cuando un flujo de control abandona una región interrumpible, todos los flujos y comportamientos de la región finalizan.

Para añadir un encadenamiento interruptor:

Primero debe comprobar que hay un elemento **`Acción`** en la **`RegiónDeActividadInterrumpible`**, así como un flujo de control de salida hacia otra acción:



1. Haga clic con el botón derecho en la flecha del **`FlujoDeControl`** y seleccione **Nuevo/a | EncadenamientoInterruptor**.



Nota: hay otra manera de añadir un **`EncadenamientoInterruptor`**: haga clic en la **`RegiónDeActividadInterrumpible`**, después haga clic con el botón derecho en el panel Propiedades y seleccione **Agregar encadenamientoInterruptor** en el menú emergente.

9.1.2 Diagrama de máquina de estados

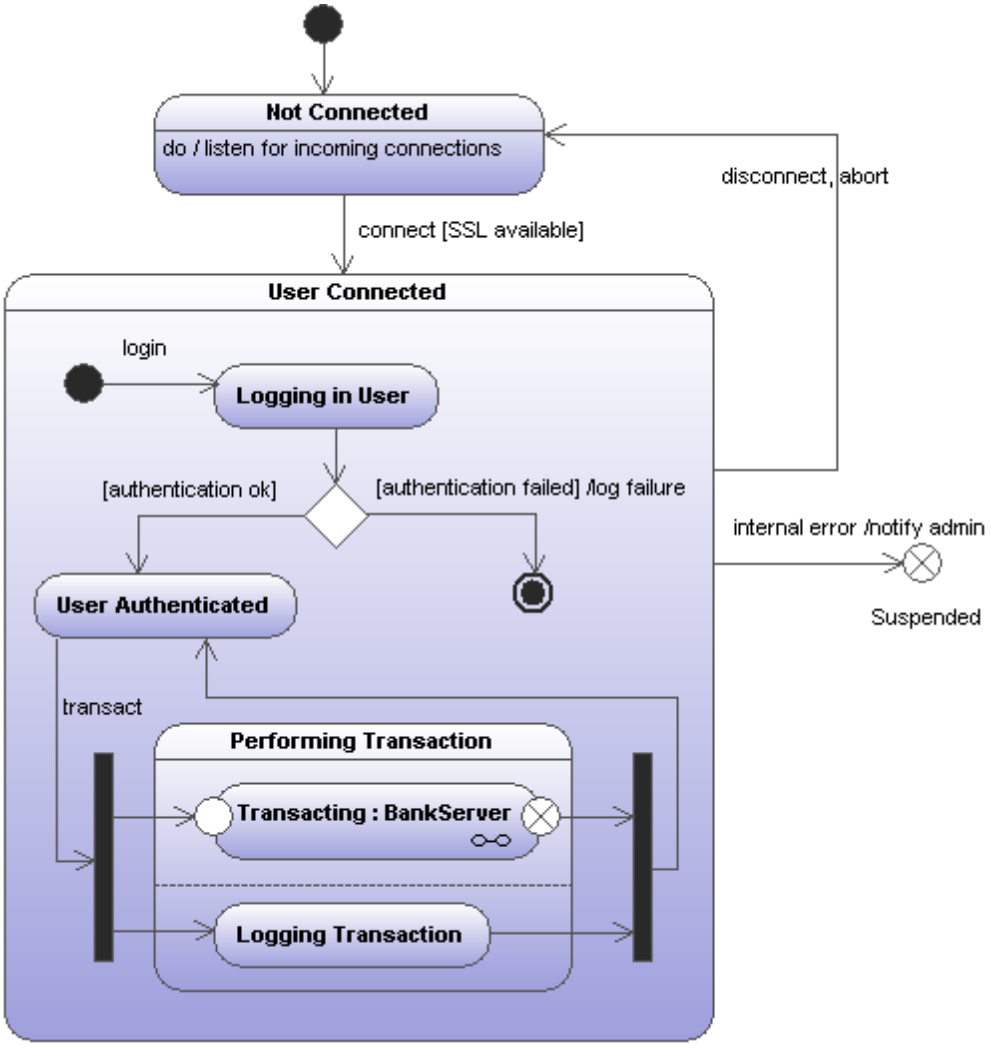
Sitio web de Altova: diagramas de máquina de estados

Los diagramas de máquina de estados modelan el comportamiento de un sistema, describiendo los diferentes estados por los que puede pasar un objeto y las transiciones de unos estados a otros. Se suelen utilizar para describir el comportamiento de un objeto que pasa por varios casos de uso. Una máquina de estados puede tener un número ilimitado de diagramas de máquina de estados (o de diagramas de estado).

Esto se puede conseguir con dos tipos de procesos:

- **Acciones:** están asociadas a las **transiciones** y son procesos a corto plazo que no se pueden interrumpir. Por ejemplo: una transición inicial **`error interno / notificar admin`**.
- **Actividades de estado (comportamientos):** están asociadas a los estados y son procesos a largo plazo que pueden ser interrumpidos por otros eventos. Por ejemplo: **`escuchar si hay conexiones entrantes`**.

Nota: el diagrama de máquina de estados que aparece a continuación está disponible en el ejemplo `Bank_MultiLanguage.ump`, en la carpeta `...\UModelExamples`.



9.1.2.1 Insertar elementos

Usar los iconos de la barra de herramientas:

1. Haga clic en un icono de la barra de herramientas Diagrama de máquina de estados.
2. Haga clic en el área de trabajo del diagrama en el que desea insertar el elemento. Para insertar varios elementos del tipo seleccionado, mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo.

Arrastrar elementos desde la Estructura del modelo hasta el diagrama:

1. En la Estructura del modelo busque el elemento que quiere insertar en el otro diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).
2. Arrastre el elemento hasta el diagrama de máquina de estados.

9.1.2.2 *Crear estados, actividades y transiciones*

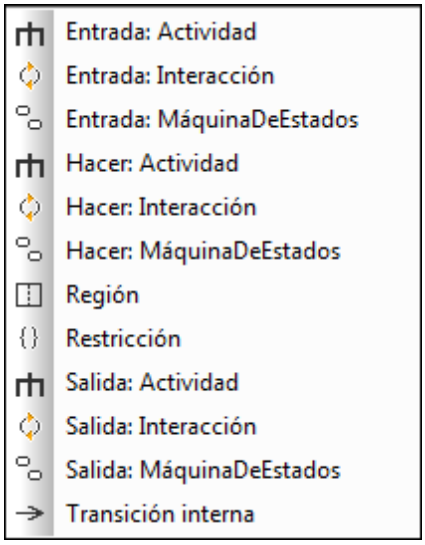
Para agregar un estado simple:

1. Haga clic en el icono **Estado** de la barra de herramientas () y después haga clic dentro del diagrama.
2. Escriba el nombre del estado y pulse **Entrar** para confirmar.

Para añadir una actividad al estado:

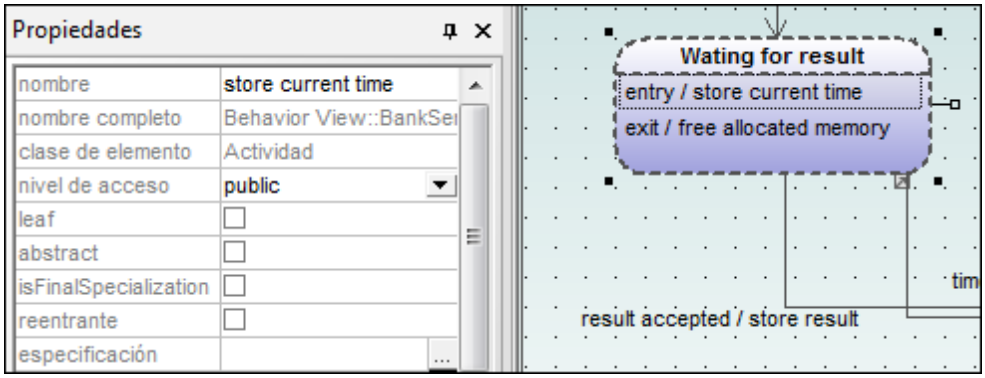
1. Haga clic con el botón derecho en el estado. En el menú contextual seleccione **Nuevo/a** y después una de las opciones del submenú.

   Las actividades Entrada, Salida y Hacer están asociadas con uno de estos comportamientos posibles: "Actividad", "Interacción" y "MáquinaDeEstados". Por tanto, las opciones de este menú contextual son:



   Estas opciones proceden de la especificación UML. Es decir, todas estas acciones internas son comportamientos y, en la especificación UML, se derivan tres clases de la clase "Comportamiento": Actividad, MáquinaDeEstados e Interacción. En el código generado no importa qué comportamiento se seleccione.

   Hay tres tipos de acciones: **Hacer** (`Do`), **Entrada** (`Entry`) y **Salida** (`Exit`). Las actividades se colocan dentro de su propio compartimento en el estado (no en una región distinta). El tipo de actividad seleccionada se utiliza como prefijo para la actividad (p. ej. `entry / store current time`).

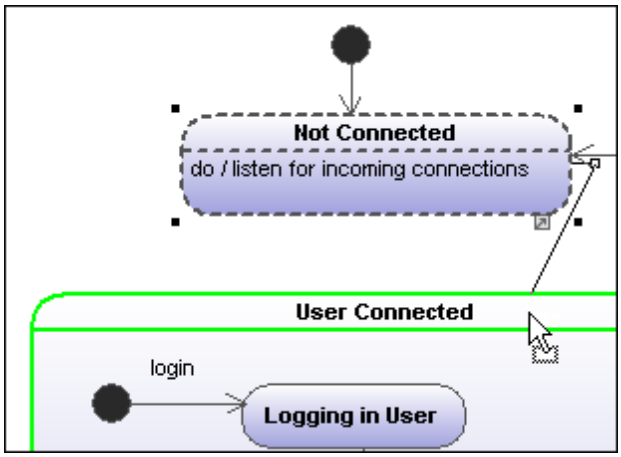


Para eliminar una actividad:

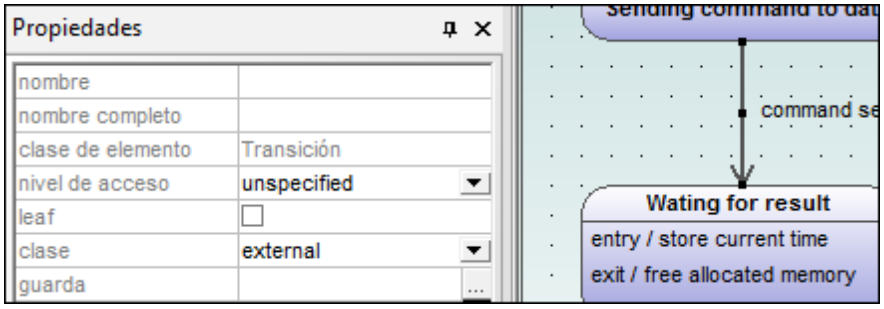
1. Haga clic en la actividad del estado y pulse la tecla **Supr**.

Para crear una transición entre dos estados:

1. Haga clic en el controlador `Transición` del estado de origen (situado a la derecha del elemento).
2. Arrastre la flecha de la transición hasta el estado de destino.



Las propiedades de la transición se pueden ver en la ventana Propiedades. En el cuadro combinado del campo `clase` puede definir el tipo de transición: externa, interna o local.



Las transiciones pueden tener un disparador de eventos, una condición de protección y una acción con el formato **disparadorEvento [condición de protección] /actividad**.

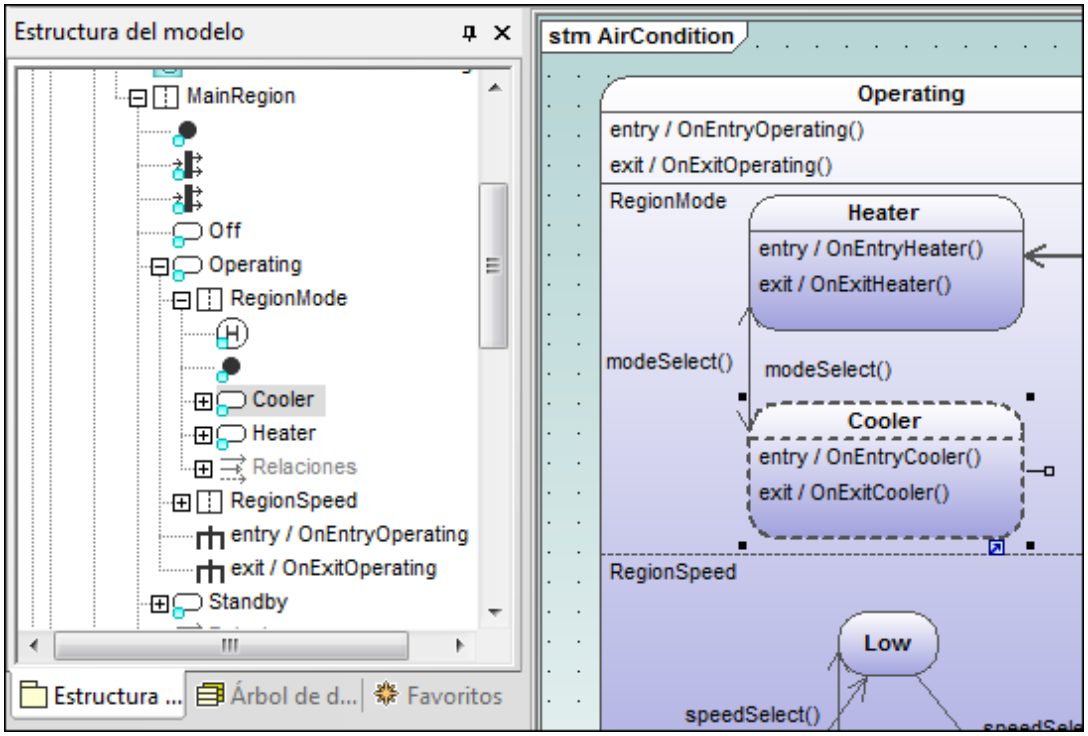
Para crear operaciones desde transiciones automáticamente

Si activa el icono **Activar/desactivar la creación automática de operaciones en el destino al escribir el nombre de la operación**  , operación correspondiente se crea automáticamente en la clase referenciada cuando se crea una transición y se escribe un nombre (p. ej. `miOperación()`).

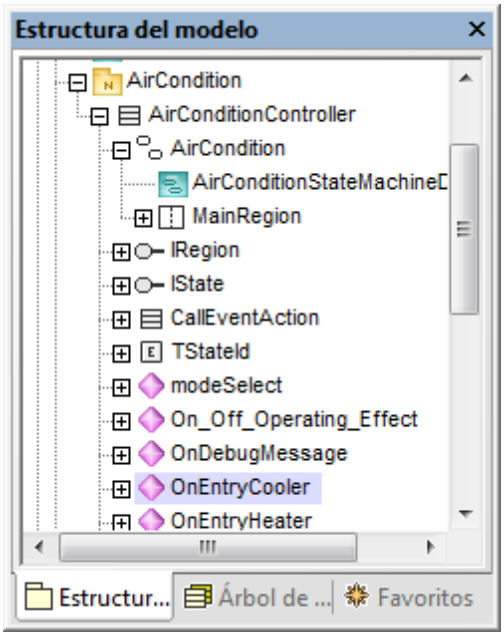
Nota: solamente se pueden crear operaciones automáticamente cuando la máquina de estados está dentro de una clase o de una interfaz.

Para crear operaciones automáticamente desde actividades:

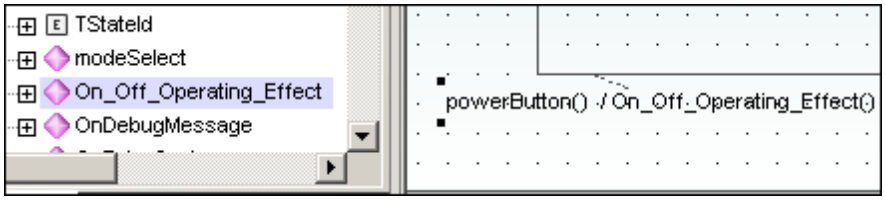
1. Haga clic con el botón derecho en el estado y seleccione la actividad/acción que desea insertar (**Nuevo/a | Entrada:Actividad**).
2. Escriba el nombre de la actividad, asegurándose de que termina con ().



El elemento nuevo también está disponible en la Estructura del modelo. Desplácese hacia abajo en la Estructura del modelo y observe que la operación `OnEntryCooler` se añadió a la clase primaria `AirConditionController`.

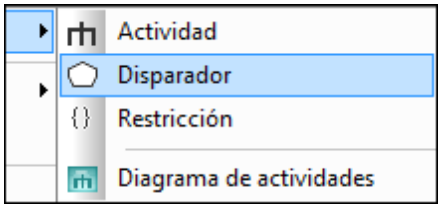


Nota: se añaden operaciones automáticamente para: **Hacer:Actividad, Entrada:Actividad, Salida:Actividad**.



Para crear un disparador de transición:

1. Haga clic con el botón derecho en una transición (en la flecha).
2. Selección **Nuevo/a | Disparador**.



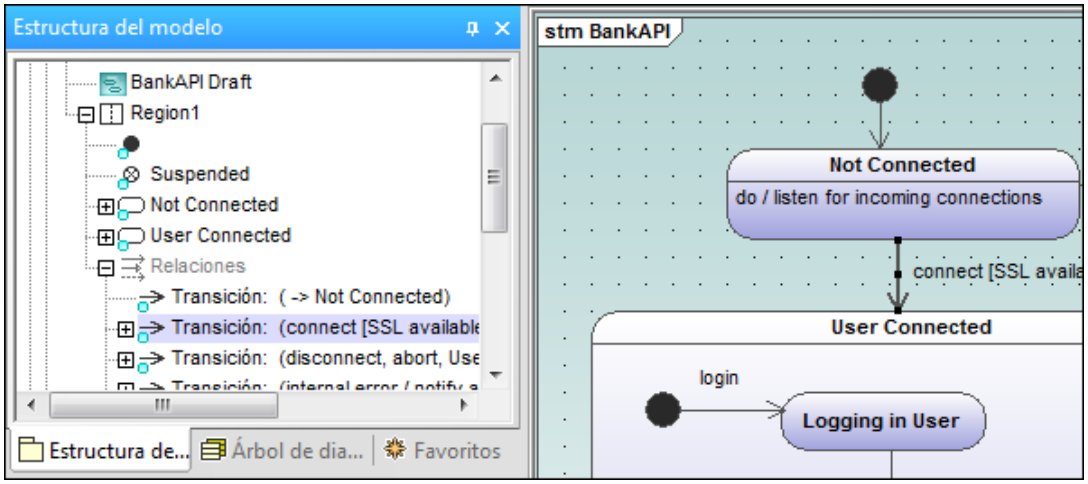
Si se trata del primer disparador del diagrama, en la etiqueta de la transición, situada encima de la flecha, aparece el carácter "a". Los disparadores tienen asignados valores predeterminados en forma de letra, estado de origen -> estado de destino.

3. Haga doble clic en el nuevo carácter e inserte las propiedades de la transición en el formato **disparadorEvento [condición de protección] / actividad**.

   Sintaxis de las propiedades de la transición: el texto insertado antes de los corchetes es el disparador. Entre los corchetes va la condición de protección y después de la barra diagonal va la actividad. Manipule esta cadena para crear o eliminar automáticamente los elementos correspondientes en la Estructura del modelo.

   Nota: para ver las propiedades de una transición, haga clic con el botón derecho en la transición y seleccione **Seleccionar en la Estructura del modelo**. El evento, la

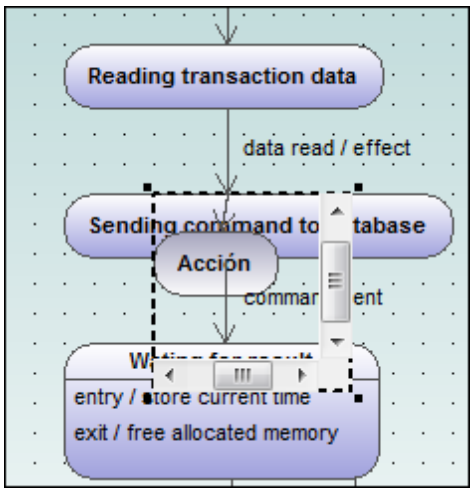
actividad y los elementos de restricción aparece debajo de la transición seleccionada.



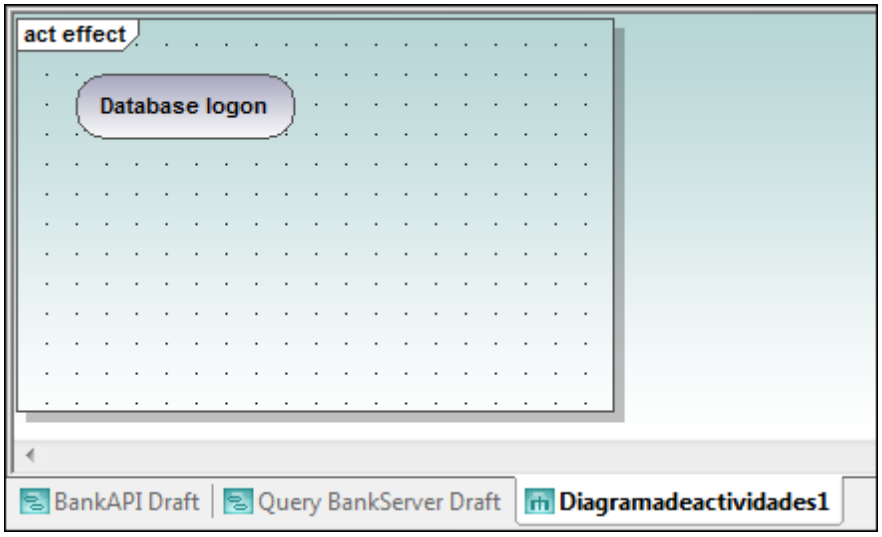
Para agregar un diagrama de actividades a una transición:

UModel ofrece una función única para añadir diagramas de actividades a las transiciones a fin de describir la transición en detalle.

1. Haga clic con el botón derecho en la transición y seleccione **Nuevo/a | Diagrama de actividades**.
   Esto inserta una ventana con un diagrama de actividades en la posición de la flecha de transición.
2. Haga clic en la ventana recién insertada y utilice las barras de desplazamiento para desplazarse por la ventana.

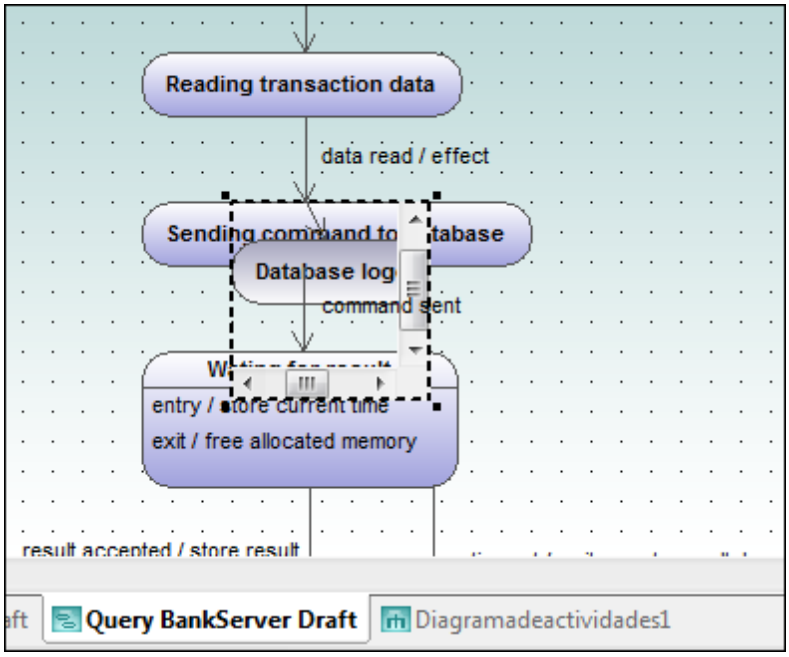


3. Haga doble clic en la ventana para abrir el diagrama de actividades en otra pestaña y seguir definiendo la transición (p. ej. cambiando el nombre de la acción a `Database logon`).

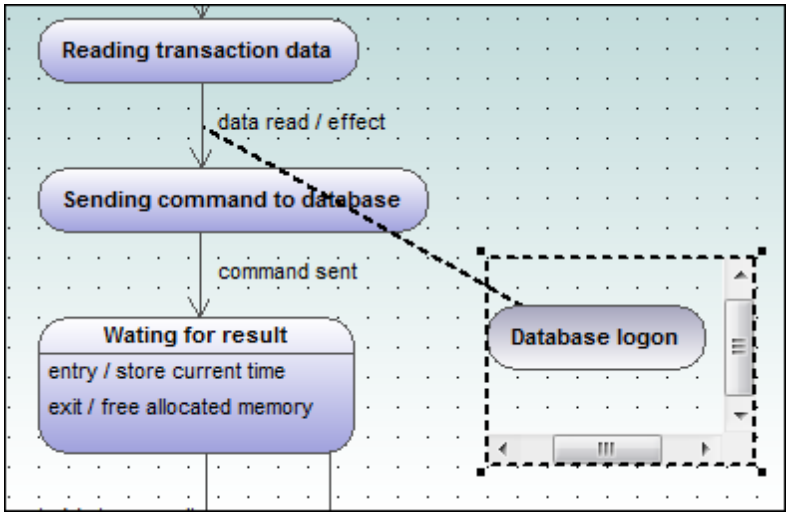


Al proyecto se añade un diagrama de actividades nuevo. Para aprender a añadir elementos de modelado de actividades nuevos al diagrama, consulte el apartado Diagrama de actividades.

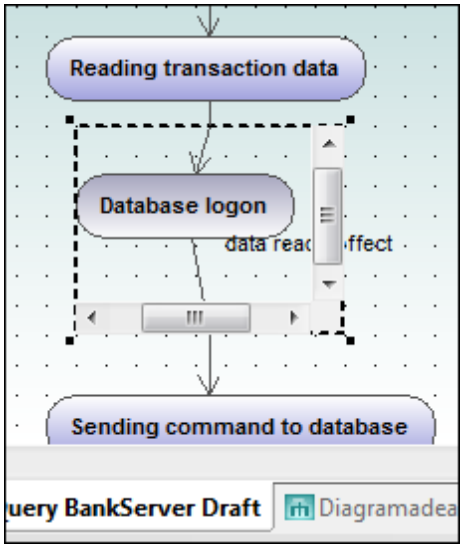
4. Haga clic en la pestaña del diagrama de máquina de estados para ver la transición actualizada.



5. Arrastre la ventana de la actividad a una posición nueva donde no moleste y ajuste el tamaño de la ventana si es necesario.



Si arrastra la ventana de la actividad y la pone entre los dos estados, la ventana ilustra la transición hacia y desde la actividad.



9.1.2.3 *Estados compuestos*

Estado compuesto
Este tipo de estado contiene un compartimento más, formado por una región. Dentro de esta región puede colocar un número ilimitado de estados.

Para añadir una región a un estado compuesto:

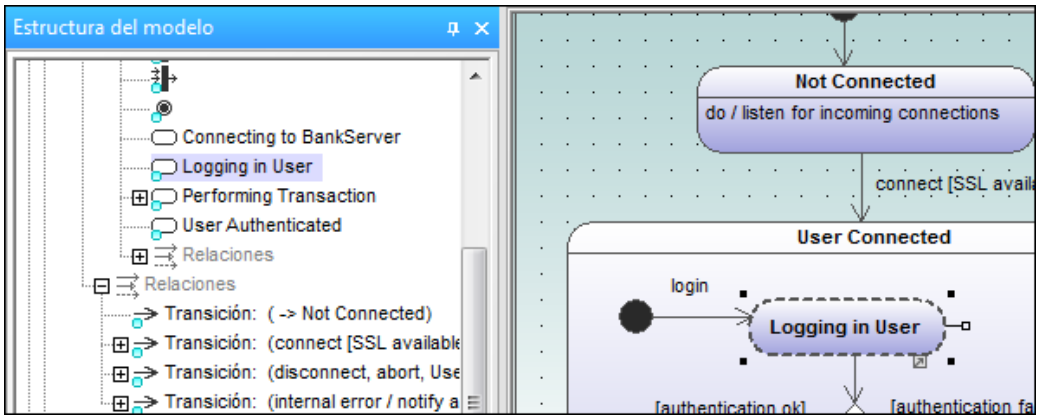
1. Haga clic con el botón derecho en el estado compuesto y seleccione **Nuevo/a | Región** del menú contextual.
   Al estado se le añade una región nueva. Las regiones se dividen con líneas discontinuas.

Para eliminar una región:

1. Haga clic en la región que desea eliminar y pulse **Supr**.
   Cuando se elimina una región de un estado ortogonal, el estado vuelve a ser un estado compuesto. Cuando se elimina la última región de un estado compuesto, el estado pasa a ser un estado simple.

Para poner un estado dentro de un estado compuesto:

1. Haga clic en el estado que desea insertar (p. ej. **Logging in User**) y arrástrelo hasta el compartimento de la región del estado compuesto.
   El compartimento de la región se resalta al soltar el elemento. El elemento insertado ahora forma parte de la región y aparece como elemento secundario de la región en el panel Estructura del modelo.

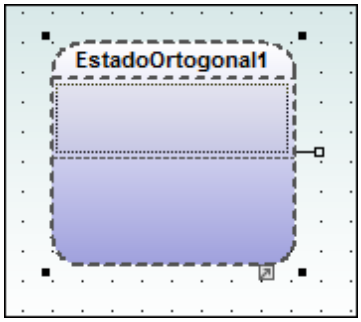


Cuando se mueve el estado compuesto, también se mueven los estados que están dentro de él.

Estado ortogonal

Este tipo de estado contiene un compartimento más, formado por dos o más regiones, que indican simultaneidad.

Haga clic con el botón derecho en un estado y seleccione **Nuevo/a | Región** para añadir regiones nuevas.



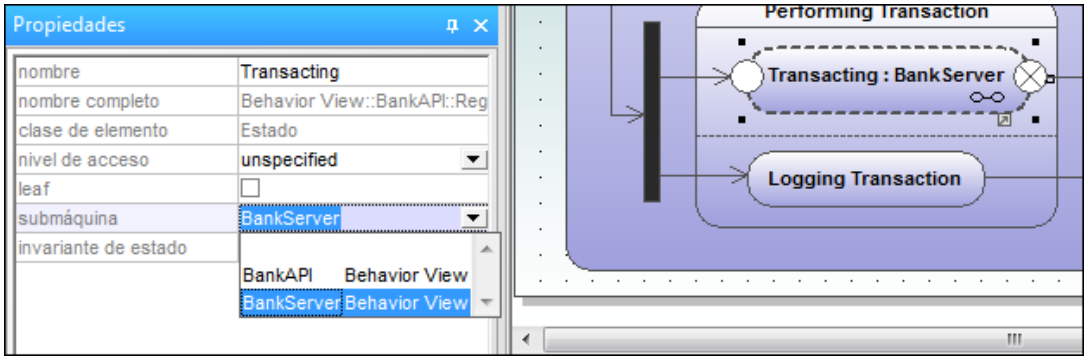
Para mostrar/ocultar el nombre de las regiones:

Haga clic en el panel Estilos, desplácese hasta el estilo `Mostrar los nombres de región en los estados` y seleccione el valor `verdadero/falso`.

Estado de submáquina
Este estado sirve para ocultar los detalles de una máquina de estados. Este estado no tiene regiones, sino que está asociado a una máquina de estados distinta.

Para definir un estado de submáquina:

1. Tras seleccionar un estado, haga clic en el cuadro combinado `submáquina` del panel Propiedades.
   Aparece una lista con todas las máquinas de estados que están definidas.
2. Seleccione la máquina de estados a la que debe hacer referencia esta submáquina.

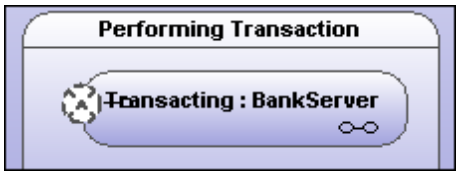


Observe que en la submáquina aparece automáticamente un icono de hipervínculo. Al hacer clic en este icono se abre la máquina de estados a la que se hace referencia (**`BankServer`**, por ejemplo).

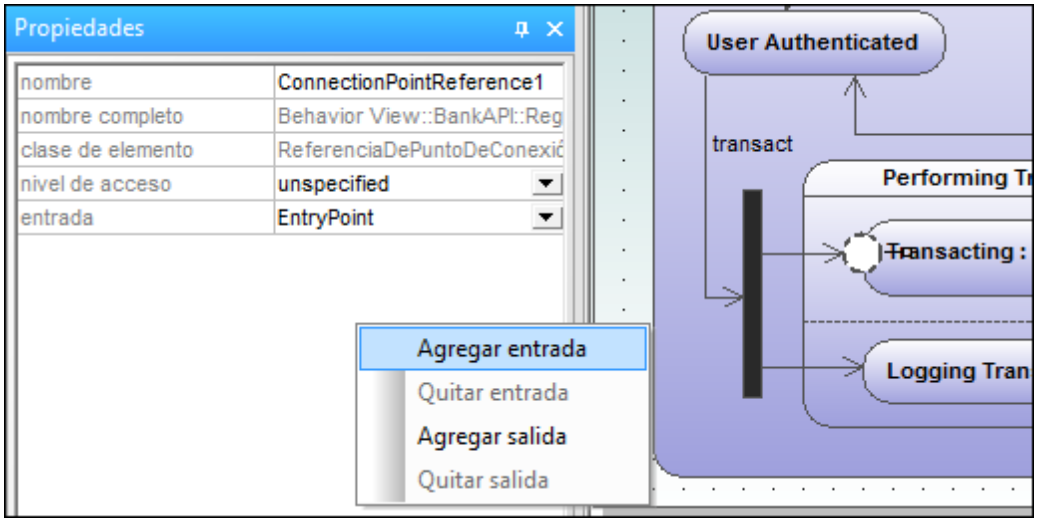
Para añadir puntos de entrada/ salida a un estado de submáquina:

- El estado de submáquina al que está conectado el punto de entrada/salida debe hacer referencia a una máquina de estados (visible en el panel Propiedades).
- Esta submáquina debe contener un punto de entrada y de salida como mínimo.

1. Haga clic en el icono **ReferenciaDePuntoDeConexión** de la barra de herramientas y después haga clic en el estado de submáquina en el que quiere insertar el punto de entrada/salida.



2. Haga clic con el botón derecho en el panel Propiedades y seleccione **Agregar entrada**. Recuerde que este menú emergente solamente aparece si en el diagrama ya existe un punto de entrada o salida.



El comando **Agregar entrada** añade un punto de entrada (`EntryPoint`) nuevo en el panel Propiedades y cambia el aspecto de la referencia de punto de conexión `ConnectionPointReference`.

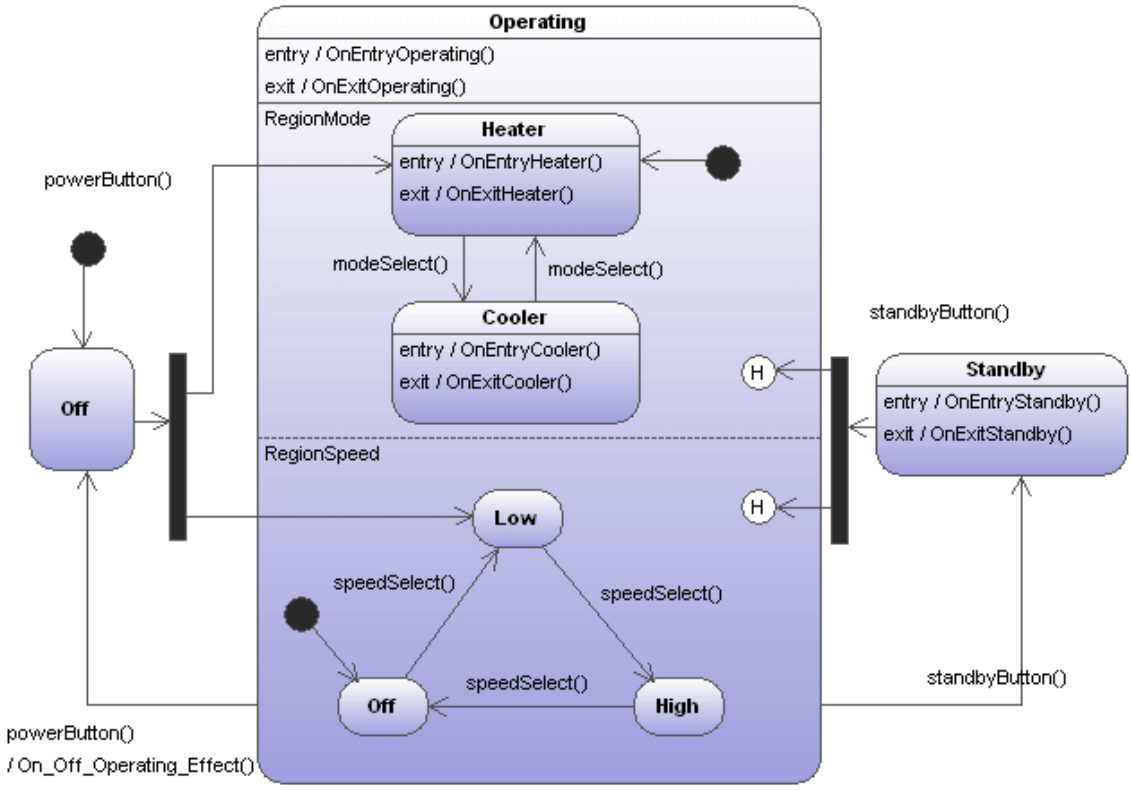
3. Use el mismo método para insertar un punto de salida (`ExitPoint`) con la opción **Agregar salida** del menú emergente.

9.1.2.4 Generar código a partir de diagramas de máquina de estados

Con UModel puede generar código ejecutable a partir de diagramas de máquina de estados (Java, VB.NET o C#). Esta función de generación de código es compatible con casi todos los elementos y características de los diagramas de máquina de estados:

- Estado
- EstadoCompuesto, con cualquier nivel jerárquico
- EstadoOrtogonal, con cualquier número de regiones
- Región
- EstadoInicial
- EstadoFinal
- Transición
- Guarda
- Disparador
- Evento de llamada
- Bifurcación
- Reunión
- Elección
- Unión
- HistorialDetallado
- HistorialSuperficial
- Acciones de entrada/salida/hacer
- Efectos

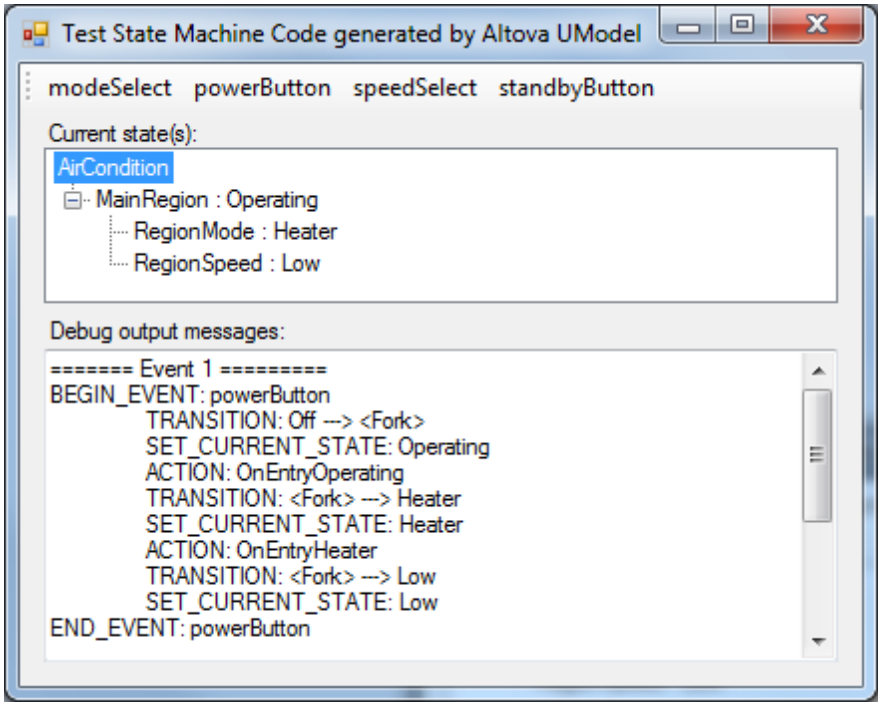
La generación de código de máquina de estados se integra en el proceso "normal" de ingeniería de ida y vuelta. Esto significa que el código de máquina de estados se puede actualizar automáticamente durante el proceso de ingeniería directa.



La imagen anterior muestra el diagrama de máquina de estados **`AirCondition`** de la carpeta `..\StateMachineCodeGeneration` del directorio `...\UModelExamples`. Por cada lenguaje de programación hay una carpeta (C#, Java y VB).

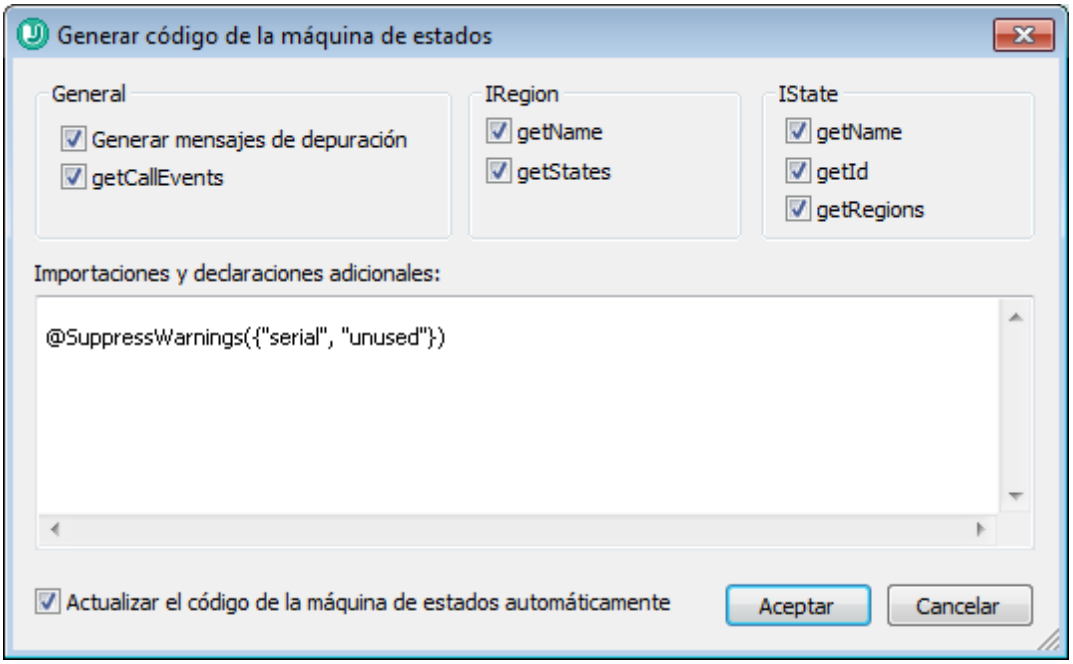
Cada directorio contiene dos carpetas: `AirCondition` y `Complex`. Cada una contiene el proyecto de UModel correspondiente, los archivos de proyecto del lenguaje de programación y los archivos de código generados. El archivo de proyecto `Complex.ump` contiene casi todos los elementos y funciones de modelado compatibles con la función de generación de código de UModel para diagramas de máquina de estados.

Además, cada carpeta contiene una aplicación de prueba (p. ej. `TestSTMAirCondition.sln` para C#) para que pueda trabajar inmediatamente con los archivos de código generados.



Para generar código a partir de un diagrama de máquina de estados:

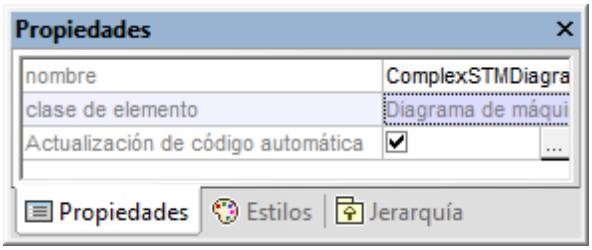
1. Haga clic con el **botón derecho** en el diagrama de máquina de estados y seleccione el comando **Generar código de la máquina de estados** o
2. Haga clic en **Proyecto | Generar código de la máquina de estados**.



Aparece un cuadro de diálogo (*imagen siguiente*). Si es necesario, ajuste las opciones de configuración predeterminadas y haga clic en **Aceptar** para generar el código.

El código de máquina de estados se actualiza automáticamente cuando se inicia el proceso de ingeniería directa. Sin embargo, esta configuración se puede cambiar. Para ello haga clic en el fondo del diagrama de máquina de estados y marque la casilla Actualización de código automática del panel Propiedades.

No es recomendable realizar cambios a mano en el código generado porque estos cambios no se traspasarán al diagrama de máquina de estados durante el proceso de ingeniería inversa.



En el panel Propiedades haga clic en el icono **Examinar** ... del campo Actualización de código automática para abrir el cuadro de diálogo "Generar código de la máquina de estados" y cambiar las opciones de configuración.

Nota: puede revisar la sintaxis del diagrama de máquina de estados haciendo clic con el botón derecho en el diagrama y seleccionando **Revisar la sintaxis de la máquina de estados**.

9.1.2.5 Trabajar con código de máquina de estados

La clase primaria de la máquina de estados (es decir, la clase controladora `controller` o la clase de contexto) es la única interfaz que existe entre el usuario de la máquina de estados y su implementación.

La clase controladora `controller` aporta los métodos que se pueden usar desde "fuera" para cambiar los estados (p. ej. después de que tengan lugar eventos externos).

No obstante, la implementación de la máquina de estados llama a los métodos de la clase `controller` (devoluciones de llamada) para informar al usuario de la máquina de estados sobre cambios de estado (`OnEntry`, `OnExit`, ...), efectos de las transiciones y la posibilidad de invalidar e implementar métodos para condiciones (guardas).

UModel puede crear operaciones simples (sin parámetros) automáticamente para comportamientos entrar/salir/hacer, efectos de transición, etc. cuando se activa la opción correspondiente (consulte el apartado Crear estados, actividades y transiciones.) Estos métodos se pueden cambiar (añadiéndoles parámetros, configurándolos como métodos abstractos, etc.).

Puede generar instancias de una máquina de estados (es decir, de su clase controladora `controller`) y todas las instancias funcionan independientemente.

- La ejecución de la máquina de estados UML está diseñada para el *modelo de ejecución hasta el final*.
- Las máquinas de estados UML suponen que el procesamiento de cada evento finaliza antes de que empiece a procesarse el siguiente evento.
- Esto también significa que las acciones entrar/salir/hacer y los efectos de las transiciones no pueden disparar transiciones/cambios de estado nuevos directamente.

Inicialización:

- Cada región de una máquina de estados debe tener un estado inicial.
- El código generado con UModel inicializa automáticamente todas las regiones de la máquina de estados (o cuando se llama al método `Initialize()` de la clase controladora).
- Si no necesita eventos `OnEntry` durante la inicialización, puede llamar a mano al método `Initialize()` e ignorar los eventos `OnEntry` durante el inicio.

Obtener el estado actual:

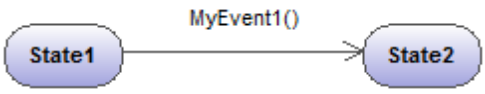
UModel admite estados compuestos y estados ortogonales, así que no hay un solo estado actual: cada región (de cualquier nivel jerárquico) puede tener un estado actual.

En el proyecto de ejemplo `AirCondition.ump` puede ver cómo se pueden recorrer las regiones hasta llegar a los estados actuales:

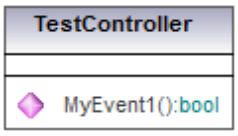
```
TreeNode rootNode = m_CurrentStateTree.Nodes.Add(m_STM.getRootState().getName());
UpdateCurrentStateTree(m_STM.getRootState(), rootNode);

private void UpdateCurrentStateTree(AirCondition.AirConditionController.IState
state, TreeNode node)
{
    foreach (AirCondition.AirConditionController.IRegion r in state.getRegions())
    {
        TreeNode childNode = node.Nodes.Add(r.getName() + " : " +
        r.getCurrentState().getName());
        UpdateCurrentStateTree(r.getCurrentState(), childNode);
    }
}
```

Ejemplo nº1: una transición simple



La operación correspondiente se genera automáticamente en UModel.



Método generado en el código:

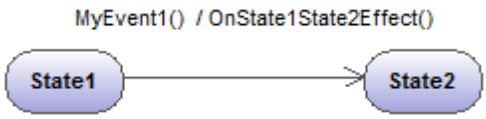
```
private class CTestStateMachine : IState
{
        …
        public bool MyEvent1()
        {
                …
        }
```



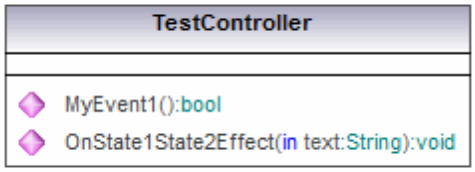
```
}
```

- El usuario de la máquina de estados debería llamar al método generado "MyEvent1" cuando tenga lugar el evento correspondiente (fuera de la máquina de estados).
- El parámetro de devolución de estos métodos-evento aporta información si el evento provocó un cambio de estado (es decir, si tuvo un efecto o no en la máquina de estados). Por ejemplo, si "State1" está activo y ocurre el evento "MyEvent1()", entonces el estado actual cambia a "State2" y "MyEvent1()" devuelve true. Si "State2" está activo y ocurre el evento "MyEvent1()", nada cambia en la máquina de estados y MyEvent1() devuelve false.

Ejemplo nº2: una transición simple con un efecto



La operación correspondiente se genera automáticamente en UModel



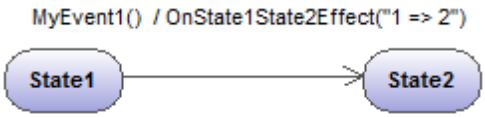
Método generado en el código:

```
private class CTestStateMachine : IState
{
        …
        // Override to handle entry/exit/do actions, transition effects,...:
        public virtual void OnState1State2Effect() {}
}
```

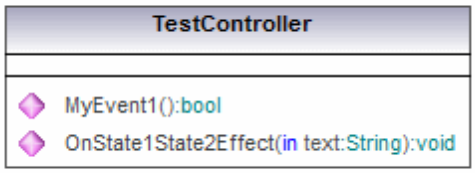
- La implementación de la máquina de estados llamará a "OnState1State2Effect()" cuando se dispare la transición del estado "State1" al estado "State2".
- Para reaccionar a este efecto "OnState1State2Effect()" debería sobrescribirse en una clase derivada de "CTestStateMachine".
- "CTestStateMachine:: OnState1State2Effect()" también puede configurarse como `abstract` y obtendrá errores de compilación hasta que se sobrescriba el método.
- Cuando "OnState1State2Effect()" no es abstracto y está activa la opción Generar mensajes de depuración, UModel genera este resultado:

```
// Overwrite to handle entry/exit/do actions, transition effects,...:
public virtual void OnState1State2Effect() {OnDebugMessage("ACTION:
OnState1State2Effect");}
```

Ejemplo nº3: una transición simple con un efecto y un parámetro



La operación correspondiente se genera automáticamente en UModel.

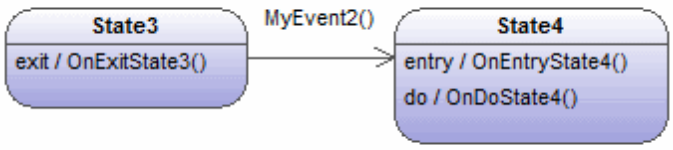


Método generado en el código:

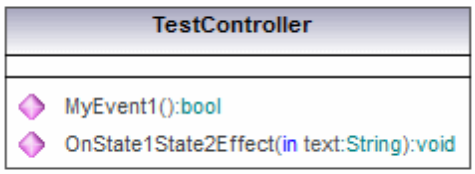
```
private class CTestStateMachine : IState
{
        …
        // Additional defined operations of the controller class:
        public virtual void OnState1State2Effect(String text)
        {
        }
}
```

- Para llevar a cabo las operaciones (creadas automáticamente por UModel), puede añadir parámetros manualmente (UModel no puede conocer el tipo necesario).
- En este ejemplo el parámetro "text:String" se añadió al método Effect de TestController. Es necesario especificar un argumento adecuado cuando se llame a este método (en este caso: "1 => 2").
- Otra posibilidad es llamar a los métodos estáticos ("MyStatic.OnState1State2Effect("1 => 2")") o a los métodos de singleton ("getSingleton().OnState1State2Effect("1 => 2")").

Ejemplo nº4: acciones entrar/salir/hacer



Las operaciones correspondientes se generan automáticamente en UModel.



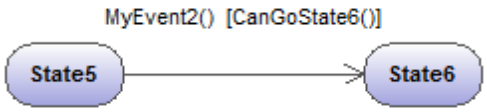
Método generado en el código:

```
private class CTestStateMachine : IState
{
        …
        // Override to handle entry/exit/do actions, transition effects,...:
        public virtual void OnExitState3() {}
        public virtual void OnEntryState4() {}
        public virtual void OnDoState4() {}
}
```

- Los estados pueden tener comportamientos entrar/salir/hacer. UModel crea automáticamente las operaciones necesarias para ellos.
- Cuando tiene lugar "MyEvent2()", la implementación de la máquina de estados llama a "OnExitState3()", si "MyEvent2" tuviera un efecto, se le llamaría después y posteriormente se llamaría a "OnEntryState4" y "OnDoState4".
- Por lo general estos métodos deberían sobrescribirse. Cuando no son abstractos y está activa la opción *Generar mensajes de depuración*, UModel genera el resultado que se describe en el ejemplo nº2.
- Estos métodos también pueden tener los parámetros que aparecen en el ejemplo nº3.

Ejemplo nº5: guardas

Las transiciones pueden tener guardas, que determinan si la transición se dispara verdaderamente.

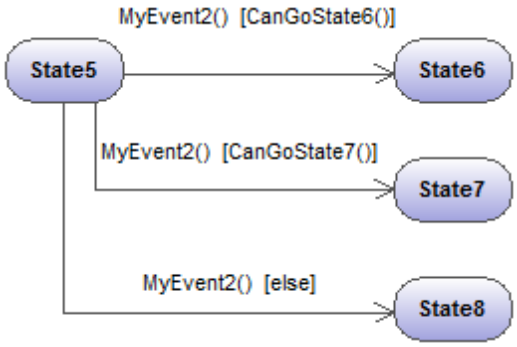


La operación correspondiente se genera automáticamente en UModel.

Método generado en el código:

```
private class CTestStateMachine : IState
{
        …
        // Additional defined operations of the controller class:
        public virtual bool CanGoState6()
        {
            return true; // Override!
        }
}
```

- Si "State5" es el estado activo y tiene lugar "MyEvent2", la implementación de la máquina de estados llamará a "CanGoState6" y, dependiendo de su resultado, la transición se disparará o no.
- Por lo general estos métodos deberían sobrescribirse. Cuando no son abstractos y está activa la opción *Generar mensajes de depuración*, UModel genera el resultado que se describe en el ejemplo nº2.
- Estos métodos también pueden tener los parámetros que aparecen en el ejemplo nº3.
- Varias transiciones pueden tener el mismo evento, pero guardas diferentes. No hay un orden definido para sondear los guardas. Si una transición no tiene guarda o si su guarda es "else", se trata como la última transición (es decir, esta transición solo se disparará si los guardas de las demás transiciones devuelven false. Por ejemplo, no está definido si primero se dispara CanGoState6 o CanGoState7, pero lo que está claro es que la tercera transición solo se disparará si CanGoState6 y CanGoState7 devuelven `false`.



Para ver más funciones y construcciones consulte los ejemplos de los archivos `AirCondition.ump` y `Complex.ump`.

9.1.2.6 Elementos

EstadoInicial (pseudoestado)
El inicio del proceso.

EstadoFinal
El final de la secuencia de los procesos.

 **PuntoDeEntrada (pseudoestado)**
El punto de entrada de una máquina de estados o de un estado compuesto.

 **PuntoDeSalida (pseudoestado)**
El punto de salida de una máquina de estados o de un estado compuesto.

 **Elección**
Representa una rama condicional dinámica donde se evalúan disparadores de guardas que se excluyen mutuamente (operación OR).

 **Unión (pseudoestado)**
Representa el final de la operación OR definida por el elemento **`Elección`**.

 **Terminar (pseudoestado)**
La detención de la ejecución de la máquina de estados.

 **Bifurcación (pseudoestado)**
Inserta una barra de bifurcación vertical. Sirve para dividir secuencias en subsecuencias simultáneas.

 **Bifurcación horizontal (pseudoestado)**
Inserta una barra de bifurcación horizontal. Sirve para dividir secuencias en subsecuencias simultáneas.

 **Reunión (pseudoestado)**
Reúne/combina subsecuencias definidas previamente. Para poder continuar todas las actividades deben completarse.

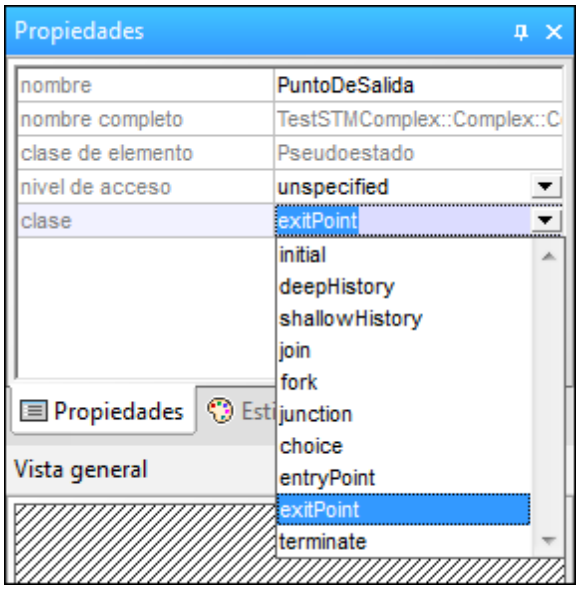
 **Reunión horizontal (pseudoestado)**
Reúne/combina subsecuencias definidas previamente. Para poder continuar todas las actividades deben completarse.

 **HistorialDetallado**

Pseudoestado que restaura el estado activo previo del estado dentro de un estado compuesto.

HistorialSuperficial
Pseudoestado que restaura el estado inicial de un estado compuesto.

Para cambiar el tipo de pseudoestado, cambie el valor del cuadro combinado `clase` en el panel Propiedades.



ReferenciaDePuntoDeConexión
Una referencia de punto de conexión representa un uso (como parte de un estado de submáquina) de un punto de entrada/salida definido en la referencia de máquina de estados al que hace referencia el estado de submáquina.

Para agregar puntos de entrada o salida a una referencia de punto de conexión:

- El estado al que está conectado el punto debe hacer referencia a una máquina de estados de submáquina (visible en el panel Propiedades).
- Esta submáquina debe contener un punto de entrada y salida como mínimo.

Transición
La relación directa que existe entre dos estados. Un objeto del primer estado realiza una acción o más y después hace referencia al segundo estado, dependiendo de un evento y de que se cumplan las condiciones de protección.

Las transiciones tienen un disparador de eventos, condiciones de protección, una acción (comportamiento) y un estado de destino.

Subelementos de evento compatibles:

- **EventoRecibirSeñal, EventoSeñal, EventoEnviarSeñal, EventoRecibirOperación, EventoEnviarOperación y EventoDeCambio**.

 **Activar/desactivar la creación automática de operaciones en el destino al escribir el nombre de la operación**
Active este icono para crear automáticamente la operación correspondiente en la clase a la que se hace referencia cuando se cree una transición y se inserte el nombre de la operación.

Nota: solamente se pueden crear operaciones automáticamente cuando la máquina de estado está dentro de una clase o de una interfaz.

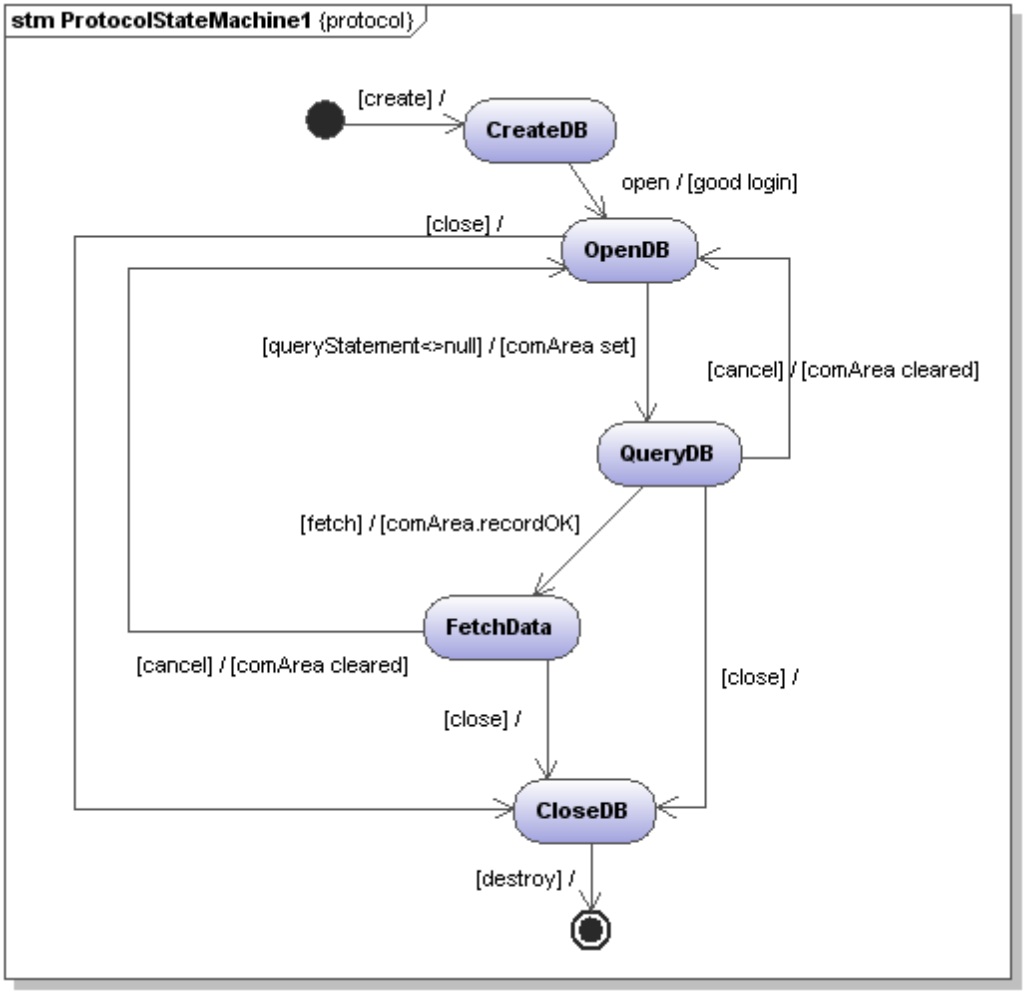
9.1.3 Diagrama de máquina de estados de protocolos

Sitio web de Altova: Diagramas de máquina de estados de protocolos UML

Las máquinas de estados de protocolos ilustran una **secuencia** de eventos a los que responde un objeto, sin necesidad de ilustrar su comportamiento propiamente dicho. La secuencia necesaria de eventos y los cambios resultantes en el estado del objeto se modelan en este tipo de diagramas.

Las máquinas de estados de protocolos se usan sobre todo para describir protocolos complejos. Por ejemplo, el acceso a bases de datos a través de una interfaz determinada o protocolos de comunicación como TCP/IP.

Las máquinas de estados de protocolos se crean igual que los diagramas de máquina de estados, pero tienen menos elementos de modelado. Las transiciones de protocolo entre los estados pueden tener condiciones previas o posteriores que definen qué debe ocurrir para que tenga lugar la transición a otro estado o cuál debe ser el estado resultante una vez tiene lugar la transición.



9.1.3.1 Insertar elementos

Usar los iconos de la barra de herramientas:

1. Haga clic en un icono de la barra de herramientas Diagrama de máquina de estados de protocolos.
2. Haga clic en el área de trabajo del diagrama en el que desea insertar el elemento. Para insertar varios elementos del tipo seleccionado, mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo.

Arrastrar elementos desde la Estructura del modelo hasta el diagrama:

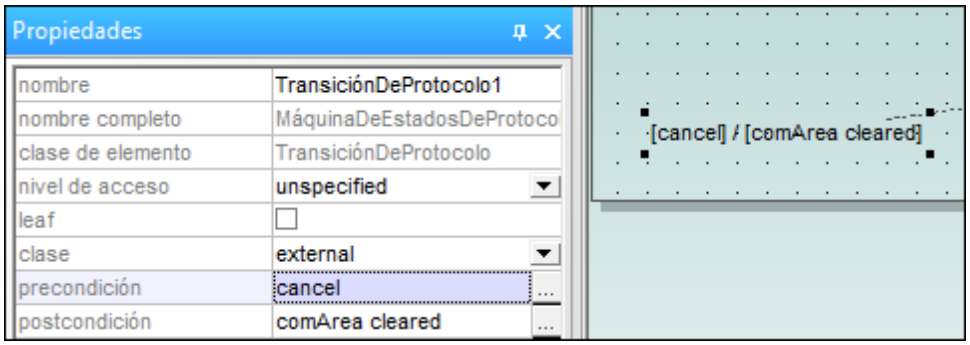
1. En la Estructura del modelo busque el elemento que quiere insertar en el otro diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).
2. Arrastre el elemento hasta el diagrama de máquina de estados de protocolos.

Para insertar un estado simple:

1. Haga clic en el icono **Estado** de la barra de herramientas y haga clic en el diagrama para insertarlo.
2. Escriba el nombre del estado y pulse **Entrar** para confirmar.
   Los estados simples no tienen regiones ni subestructuras.

Para crear una transición de protocolo entre dos estados:

1. Haga clic en el controlador `Transición` del estado de origen (situado a la derecha del elemento) o en el icono **TransiciónDeProtocIo** de la barra de herramientas.
2. Arrastre la flecha de la transición hasta el estado de destino.
   El cursor de texto se habilita automáticamente para que pueda insertar la condición previa o posterior. Recuerde que es obligatorio utilizar corchetes y la barra diagonal en las condiciones.
   Si inserta la condición (previa o posterior) en el panel Propiedades, los corchetes y la barra diagonal se escriben automáticamente en el diagrama.



Para crear e insertar estados compuestos y estados de submáquina:

- Consulte el apartado Estados compuestos

9.1.3.2 *Elementos*

Estado
Estado simple con un compartimento.

Estado compuesto
Este tipo de estado contiene un compartimento más que tiene una sola región. Dentro de esta región puede colocar un número ilimitado de estados.

Estado ortogonal
Este tipo de estado contiene un compartimento más, formado por dos o más regiones, que indican simultaneidad.

Haga clic con el botón derecho en un estado y seleccione **Nuevo/a | Región** para añadir una región nueva.

 **Estado de submáquina**
Este estado sirve para ocultar detalles de una máquina de estados. Este estado no tiene regiones pero está asociado a una máquina de estados distinta.

 **EstadoInicial (pseudoestado)**
El principio del proceso

 **EstadoFinal**
El fin de la secuencia de los procesos

 **PuntoDeEntrada (pseudoestado)**
El punto de entrada de una máquina de estados o de un estado compuesto.

 **PuntoDeSalida (pseudoestado)**
El punto de salida de una máquina de estados o de un estado compuesto.

 **Elección**
Representa una rama condicional dinámica en la que se evalúan disparadores de guardas que se excluyen mutuamente (operación OR).

 **Unión (pseudoestado)**
Representa el final de la operación OR definida por el elemento Elección.

 **Terminar (pseudoestado)**
La detención de la ejecución de la máquina de estados.

 **Bifurcación (pseudoestado)**
Inserta una barra de bifurcación vertical. Sirve para dividir secuencias en subsecuencias simultáneas.

 **Bifurcación horizontal (pseudoestado)**
Inserta una barra de bifurcación horizontal. Sirve para dividir secuencias en subsecuencias simultáneas.

 **Reunión (pseudoestado)**
Reúne/combina subsecuencias definidas previamente. Para poder continuar todas las actividades deben completarse.

 **Reunión horizontal (pseudoestado)**
Reúne/combina subsecuencias definidas previamente. Para poder continuar todas las actividades deben completarse.

 **ReferenciaDePuntoDeConexión**
Representa un uso (como parte de un estado de submáquina) de un punto de entrada/salida definido en la referencia de máquina de estados por el estado de submáquina.

Para añadir puntos de entrada/salida en una referencia de punto de conexión:

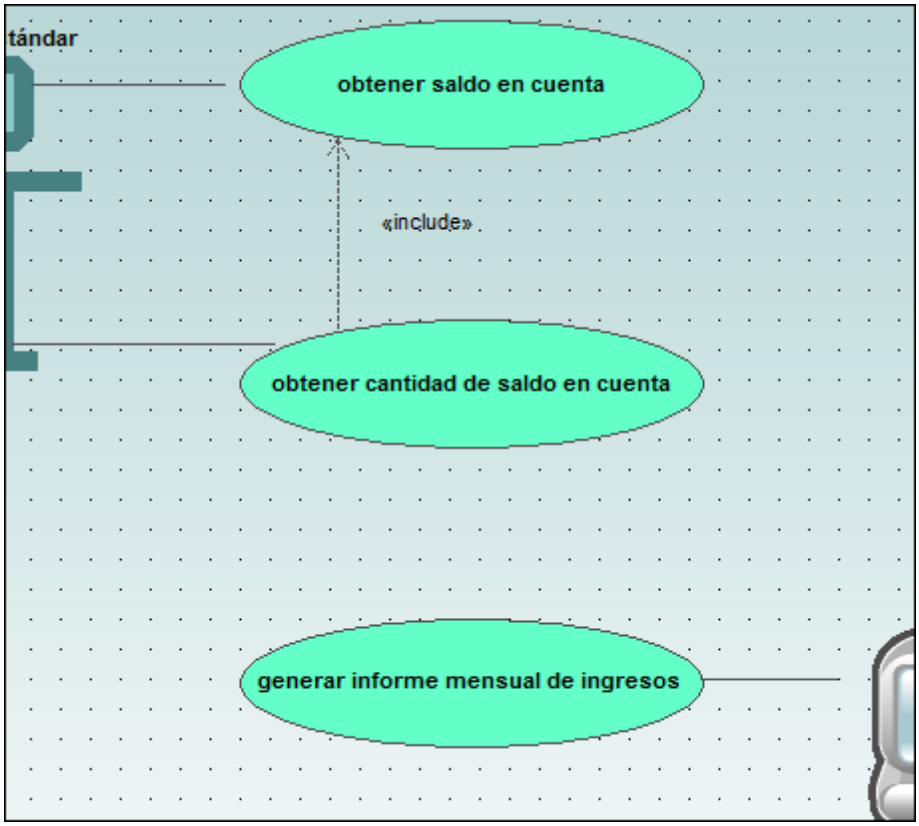
- El estado al que está conectado el punto debe hacer referencia a una máquina de estado de submáquina (visible en el panel Propiedades).
- Esta submáquina debe contener un punto de entrada y otro de salida como mínimo.

 **TransiciónDeProtocolo**
Relación directa entre dos estados. Un objeto del primer estado realiza una operación o más y después hace referencia al segundo estado, dependiendo de un evento y de que se cumplan las condiciones previas o posteriores.

Para más información consulte el apartado Insertar elementos.

9.1.4 Diagrama de casos de uso

Consulte la sección Casos de uso del tutorial para obtener más información sobre cómo usar los diagramas de casos de uso.



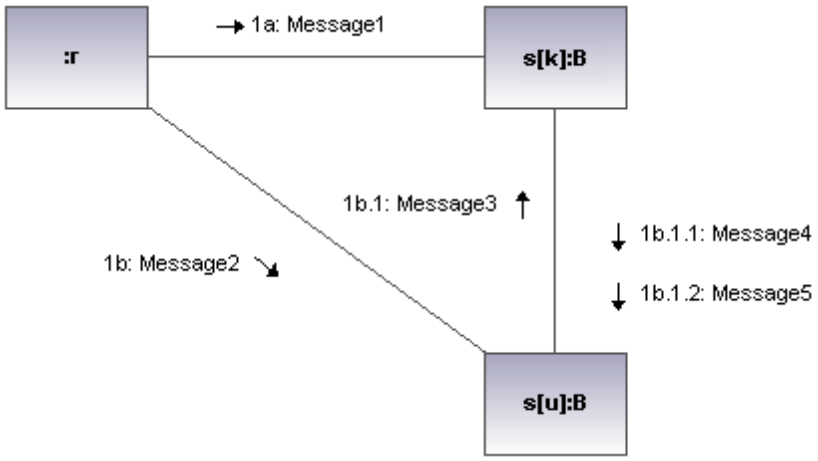
9.1.5 Diagrama de comunicación

Sitio web de Altova: Diagramas de comunicación UML

Los diagramas de comunicación muestran cómo interactúan los objetos en tiempo de ejecución (p. ej. los flujos de mensaje) e ilustran las relaciones que existen entre los objetos. Básicamente modelan el comportamiento dinámico de los casos de uso.

Los diagramas de comunicación se diseñan igual que los diagramas de secuencia, excepto que la notación tiene otro formato. Los mensajes se numeran para ilustrar su secuencia y su anidamiento.

Con UModel puede generar diagramas de comunicación a partir de diagramas de secuencia y viceversa. Para más información consulte el apartado Generar diagramas de secuencia.



9.1.5.1 Insertar elementos

Usar iconos de la barra de herramientas:

1. Haga clic en un icono de la barra de herramientas Diagrama de comunicación.
2. Haga clic en el área de trabajo del diagrama en el que desea insertar el elemento. Para insertar varios elementos del tipo seleccionado, mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo.

Arrastrar elementos desde la Estructura del modelo hasta el diagrama:

1. En la Estructura del modelo busque el elemento que quiere insertar en el otro diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).
2. Arrastre el elemento hasta el diagrama de comunicación.

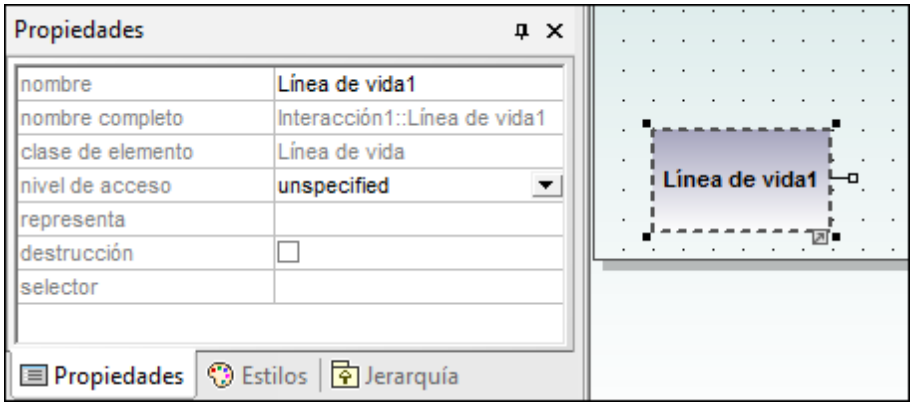
Línea de vida

El elemento línea de vida es un participante de la interacción. En UModel puede insertar otros elementos (clases, por ejemplo) en el diagrama de secuencia. Cada elemento aparece como una línea de vida nuevas. El color y el degradado de las líneas de vida se pueden redefinir en el cuadro combinado `Título - color de degradado` del panel Estilos.

Para crear un nombre de línea de vida **multilínea** pulse **Ctrl+Entrar**.

Para insertar una línea de vida de comunicación:

1. Haga clic en el icono **Línea de vida**  de la barra de herramientas y después haga clic en el área de trabajo del diagrama para insertarla.



2. Escriba el nombre de la línea de vida o conserve el nombre predeterminado `Línea de vida 1`.

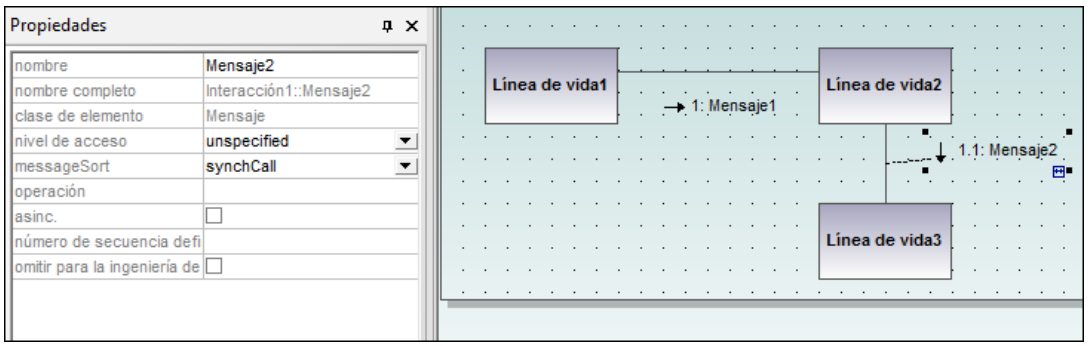
Mensajes

Un mensaje es un elemento de modelado que define un tipo concreto de comunicación en una interacción. Una comunicación puede lanzar una señal, invocar una operación, crear o destruir una instancia, etc. El mensaje especifica el tipo de comunicación, así como el remitente y el destinatario.

Mensaje (Llamada) **Mensaje (Respuesta)** **Mensaje (Creación)** **Mensaje (Destrucción)**

Para insertar un mensaje:

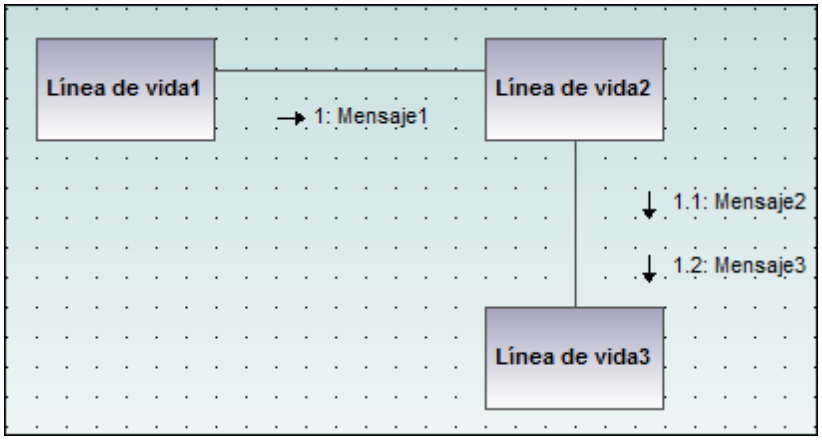
1. En la barra de herramientas haga clic en el icono del mensaje que desea insertar.
2. Ahora haga clic en el remitente y arrastre el puntero hasta el destinatario (un destinatario válido es el que aparece resaltado al pasar el puntero por encima).



Nota: mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo para insertar varios mensajes.

Para insertar más mensajes:

1. Haga clic con el botón derecho en una línea de comunicación del diagrama y seleccione **Nuevo/a | Mensaje**.



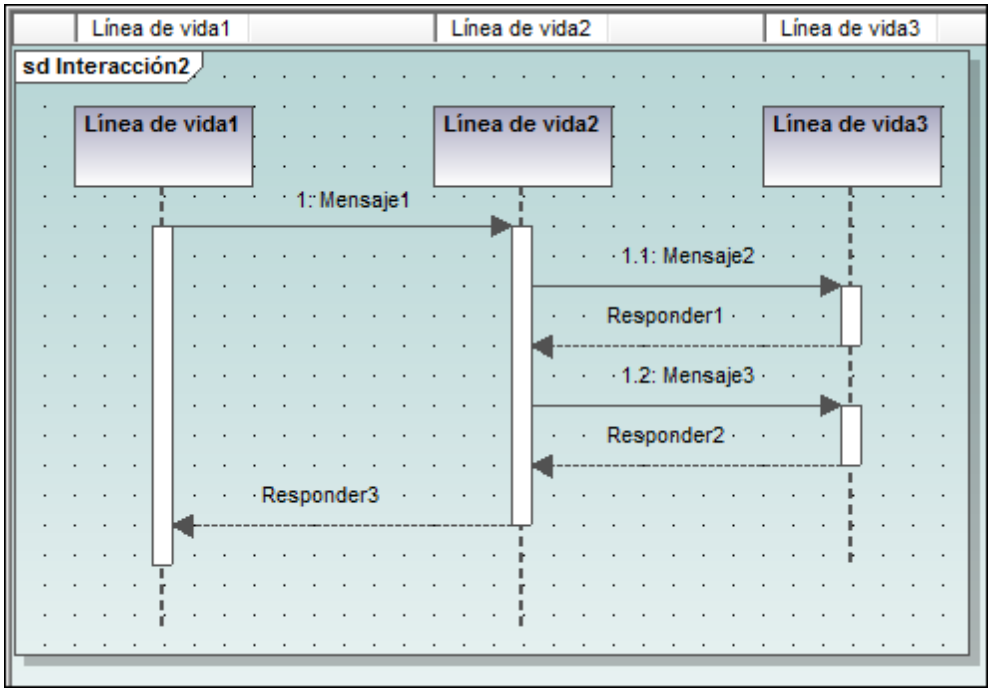
Nota: la dirección en la que se arrastra la flecha define la dirección del mensaje. Los mensajes de respuesta pueden apuntar en ambas direcciones.

Numeración de los mensajes
Los diagramas de comunicación utilizan la notación decimal para numerar los mensajes, lo cualquier facilita comprender la estructura jerárquica de los mensajes del diagrama. La secuencia es una lista separada por puntos de números en secuencia seguidos por dos puntos y el nombre del mensaje.

Generar diagramas de secuencia a partir de diagramas de comunicación:

UModel puede generar diagramas de comunicación a partir de diagramas de secuencia y viceversa:

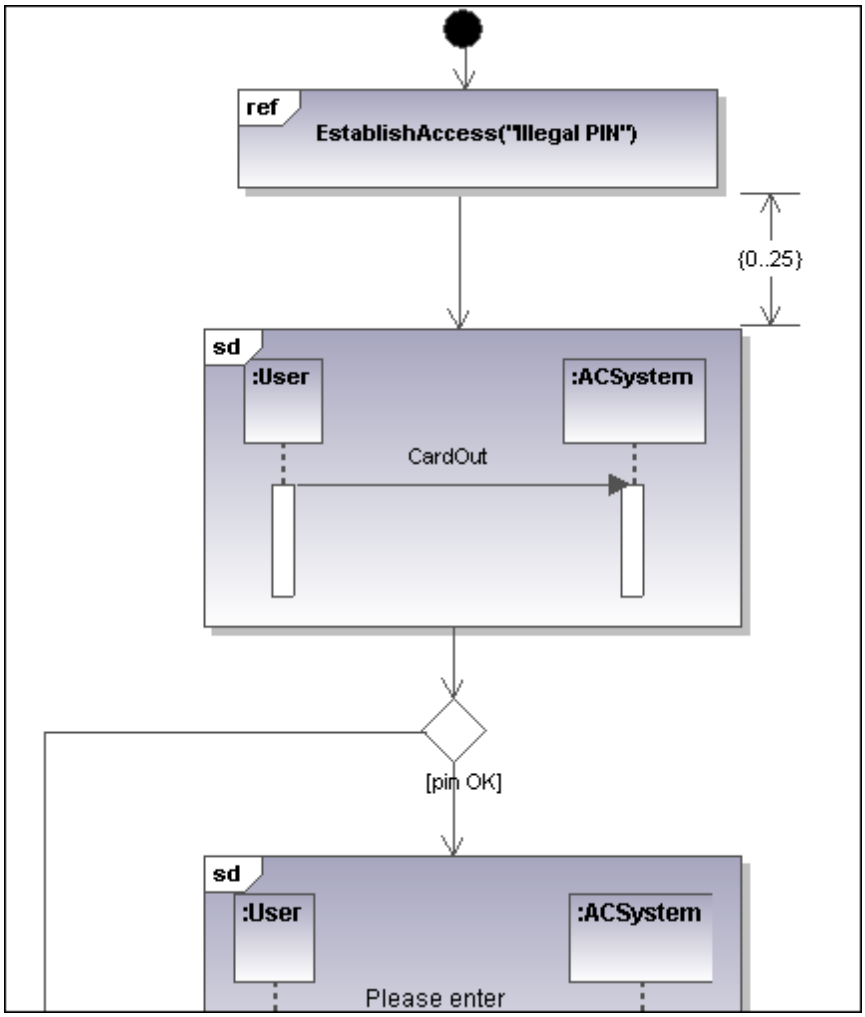
- Haga clic con el botón derecho en el área de trabajo del diagrama de comunicación y seleccione **Generar diagrama de secuencia** del menú contextual.



9.1.6 Diagrama global de interacción

Sitio web de Altova: Diagramas globales de interacción

Los diagramas globales de interacción son un tipo de diagrama de actividades que ofrecen un resumen de la interacción entre otros diagramas de interacción como diagramas de secuencia, de actividades, de comunicación o de ciclo de vida. El método para construir este tipo de diagramas es similar al utilizado para los diagramas de actividades y usa los mismos elementos de modelado: bifurcaciones, reuniones, nodo inicial, nodo final, etc.

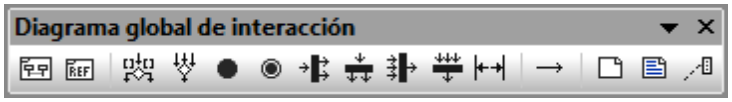


En lugar de actividades, este diagrama utiliza dos tipos distintos de interacciones: **Interacción** y **UsoDeInteracción**.

Los elementos **Interacción** se presentan como iconos de un diagrama de secuencia, comunicación, ciclo de vida o diagrama global de interacción, en un marco que tiene la abreviatura `SD` en la esquina superior izquierda.

Las instancias de los elementos **Interacción** son referencias a diagramas de interacción ya disponibles. Éstas tienen la abreviatura `Ref` y el nombre de la instancia en el marco de título.

9.1.6.1 *Insertar elementos*



Usar iconos de la barra de herramientas:

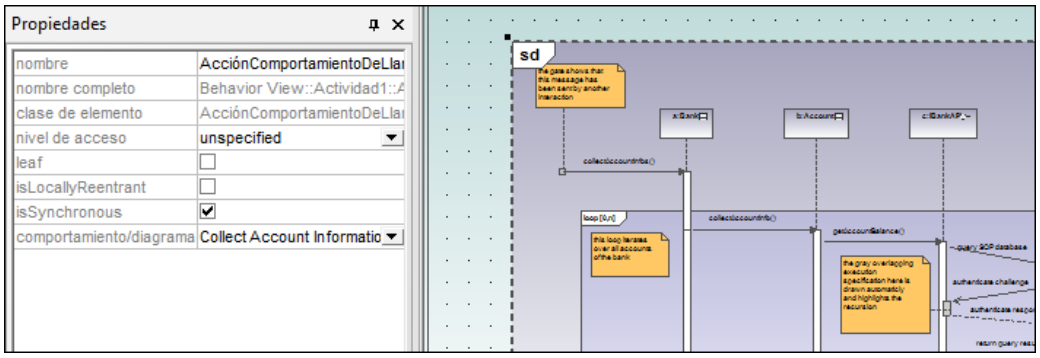
1. Haga clic en un icono de la barra de herramientas Diagrama global de interacción.
2. Haga clic en el área de trabajo del diagrama en el que desea insertar el elemento. Para insertar varios elementos del tipo seleccionado, mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo.

Arrastrar elementos desde la Estructura del modelo hasta el diagrama:

1. En la Estructura del modelo busque el elemento que quiere insertar en el otro diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).
2. Arrastre el elemento hasta el diagrama global de interacción.

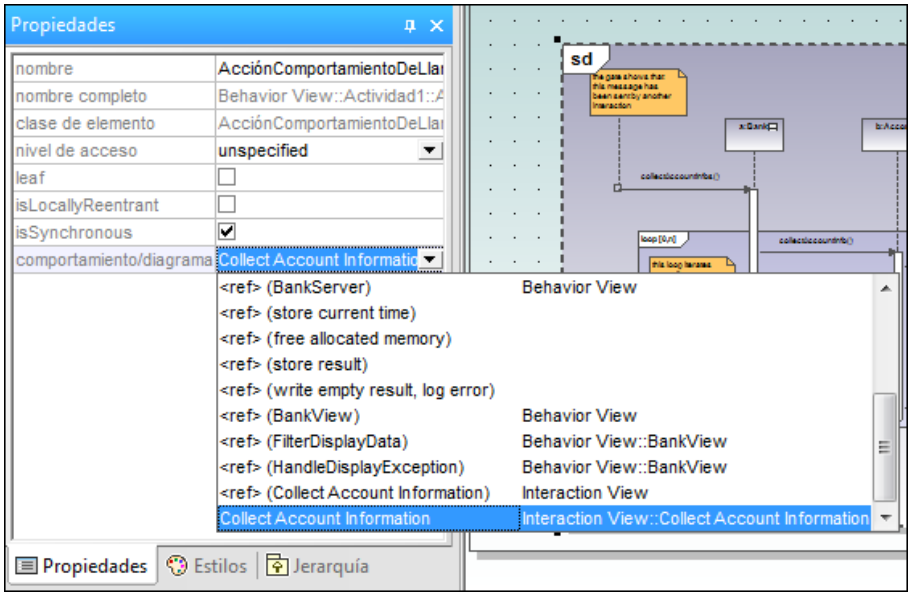
Insertar un elemento Interacción:

1. Haga clic en el icono **AcciónComportamientoDeLlamada (Interacción)**  de la barra de herramientas y haga clic en el área de trabajo del diagrama para insertar la interacción.

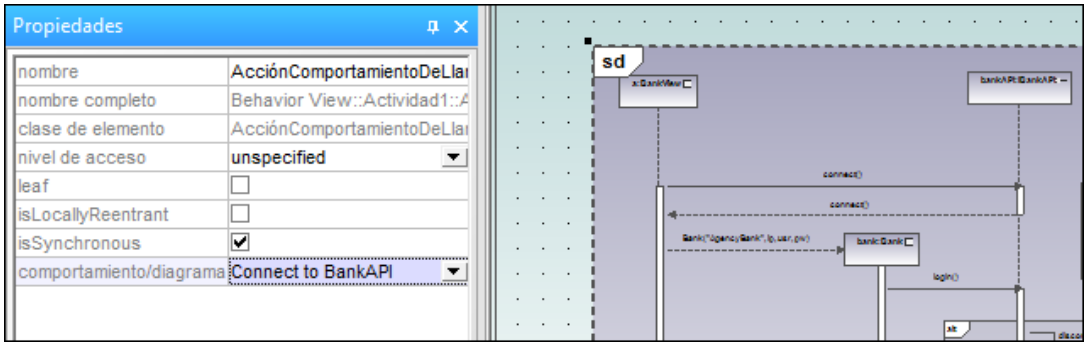


Si usa el archivo de ejemplo `Bank_MultiLanguage.ump` de la carpeta `...\UModelExamples`, el diagrama de secuencia **`Collect Account Information`** se inserta automáticamente. El primer diagrama de secuencia de la Estructura del modelo se selecciona por defecto.

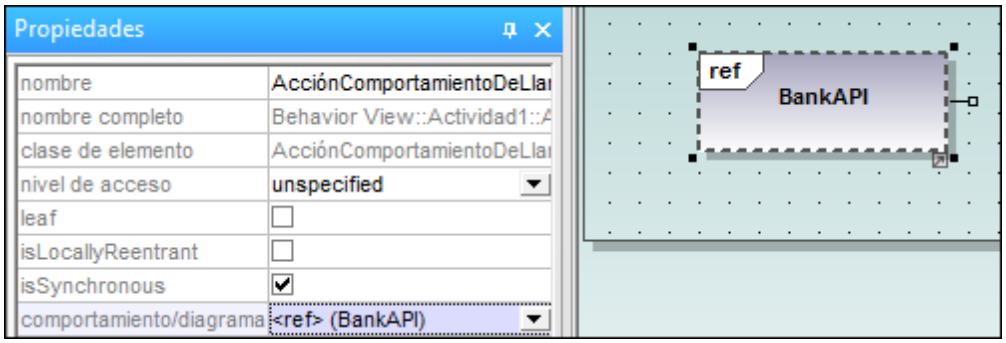
2. Para cambiar el elemento **Interacción** predeterminado, haga clic en el cuadro combinado `comportamiento/diagrama` del panel Propiedades. La lista desplegable incluye todos los elementos que se pueden insertar.



3. Haga clic en el elemento que desea insertar (p. ej. `Connect to BankAPI`).



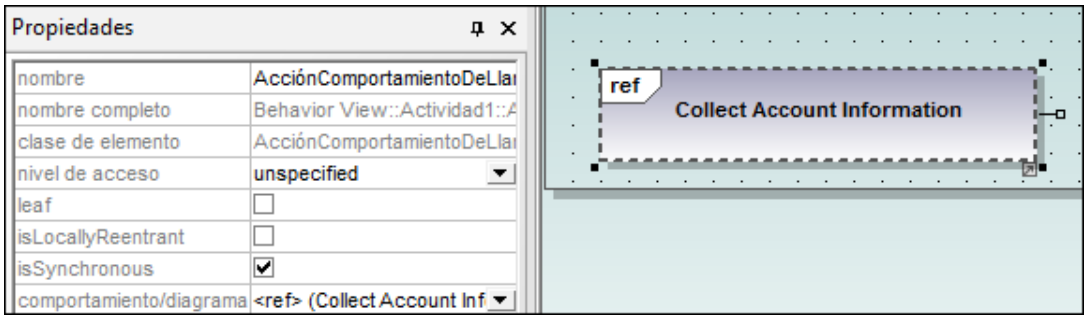
Como este también es un diagrama de secuencia, el elemento **Interacción** aparece como un icono que representa el diagrama de secuencia.
Si selecciona `<ref> BankAPI`, aparece la instancia del elemento **Interacción**.



Insertar una instancia del elemento Interacción:

1. Haga clic en el icono **AcciónComportamientoDeLlamada (UsoDeInteracción)**  de la barra de herramientas y haga clic en el área de trabajo del diagrama para insertar la instancia.

Si usa el archivo de ejemplo `Bank_MultiLanguage.ump` de la carpeta `...\UModelExamples`, UModel inserta **`Collect Account Information`** automáticamente como instancia de interacción. El primer diagrama de secuencia disponible se selecciona por defecto.



2. Para cambiar de elemento **Interacción** haga clic en el cuadro combinado `comportamiento/diagrama` del panel Propiedades. La lista desplegable incluye todos los elementos disponibles que se pueden insertar.
3. Seleccione la instancia que desea insertar.
   Recuerde que todos los elementos que se insertan de esta manera aparecen como en la imagen anterior, es decir, con la abreviatura **ref** en el marco de título.

 **NodoDeDecisión**
Inserta un nodo de decisión que tiene una sola transición entrante y varias transiciones salientes protegidas con guardas. Para más información consulte el apartado Crear una rama.

 **NodoDeCombinación**
Inserta un nodo de combinación que une las transiciones alternas definidas por el nodo de decisión. El nodo de combinación no sincroniza los procesos simultáneos, sino que selecciona uno de los procesos.

 **NodoInicial**
El principio del proceso. Una interacción puede tener más de un nodo inicial.

 **NodoFinalDeActividad**
El final del proceso de interacción. Una interacción puede tener más de un nodo final. Todos los flujos se detienen cuando se encuentra el primer nodo final.

 **NodoDeBifurcación**
Inserta un nodo de bifurcación vertical. Sirve para dividir flujos en varios flujos simultáneos.

 **NodoDeBifurcación (Horizontal)**
Inserta un nodo de bifurcación horizontal. Sirve para dividir flujos en varios flujos simultáneos.

 **NodoDeReunión**
Inserta un nodo de reunión vertical. Sirve para sincronizar varios flujos definidos por el nodo de bifurcación.

 **NodoDeReunión (horizontal)**
Inserta un nodo de reunión horizontal. Sirve para sincronizar varios flujos definidos por el nodo de bifurcación.

 **RestricciónDeDuración**
Una duración define una EspecificaciónDeValor que denota una duración entre un punto inicial y un punto final. Las duraciones suelen ser expresiones que representan el tiempo que puede pasar.

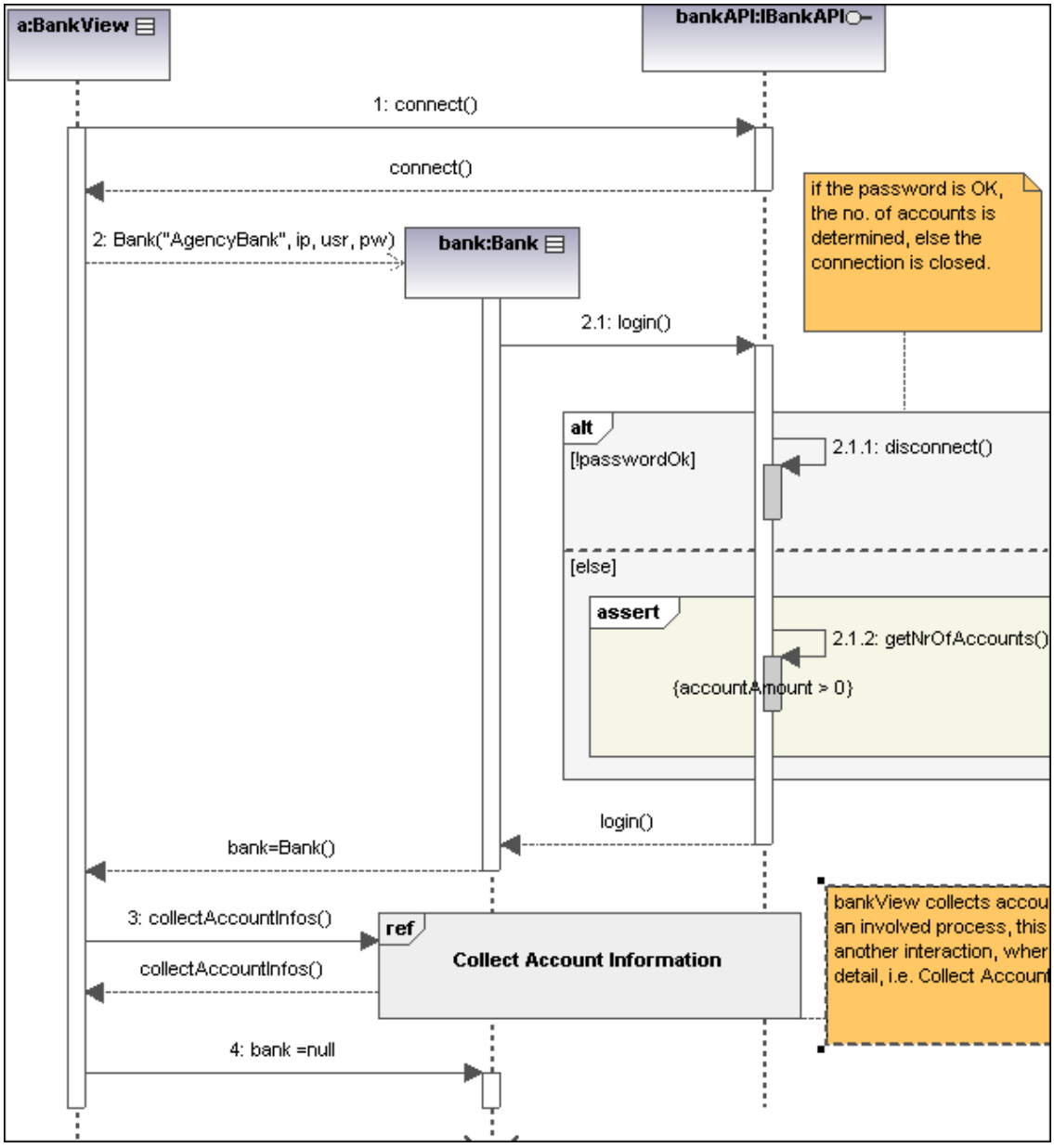
 **FlujoDeControl**
Un flujo de control es una línea con una flecha que conecta dos comportamientos e inicia una interacción después de que finalice la anterior.

9.1.7 Diagrama de secuencia

Sitio web de Altova:  Diagramas de secuencia UML

En UModel puede crear los diagramas de secuencia estándar definidos por UML y manipular objetos y mensajes con total facilidad para modelar casos de uso. Los diagramas de secuencia que aparecen en esta sección proceden de los proyectos de ejemplo `Bank_Java.ump`, `Bank_CSharp.ump` y `Bank_MultiLanguage.ump` del directorio `...\UModelExamples`.

Nota: también puede generar diagramas de secuencia a partir de código fuente. Para más información consulte el apartado Generar diagramas de secuencia .



9.1.7.1 Insertar elementos

Los diagramas de secuencia modelan las interacciones dinámicas de los objetos en tiempo de ejecución por medio de mensajes. Suelen utilizarse para explicar casos de uso.

- Las **líneas de vida** son recuadros alineados horizontalmente en la parte superior del diagrama y tienen una línea de puntos vertical que representa la vida del objeto durante la interacción. Los mensajes se dibujan como flechas entre las líneas de vida de los objetos.

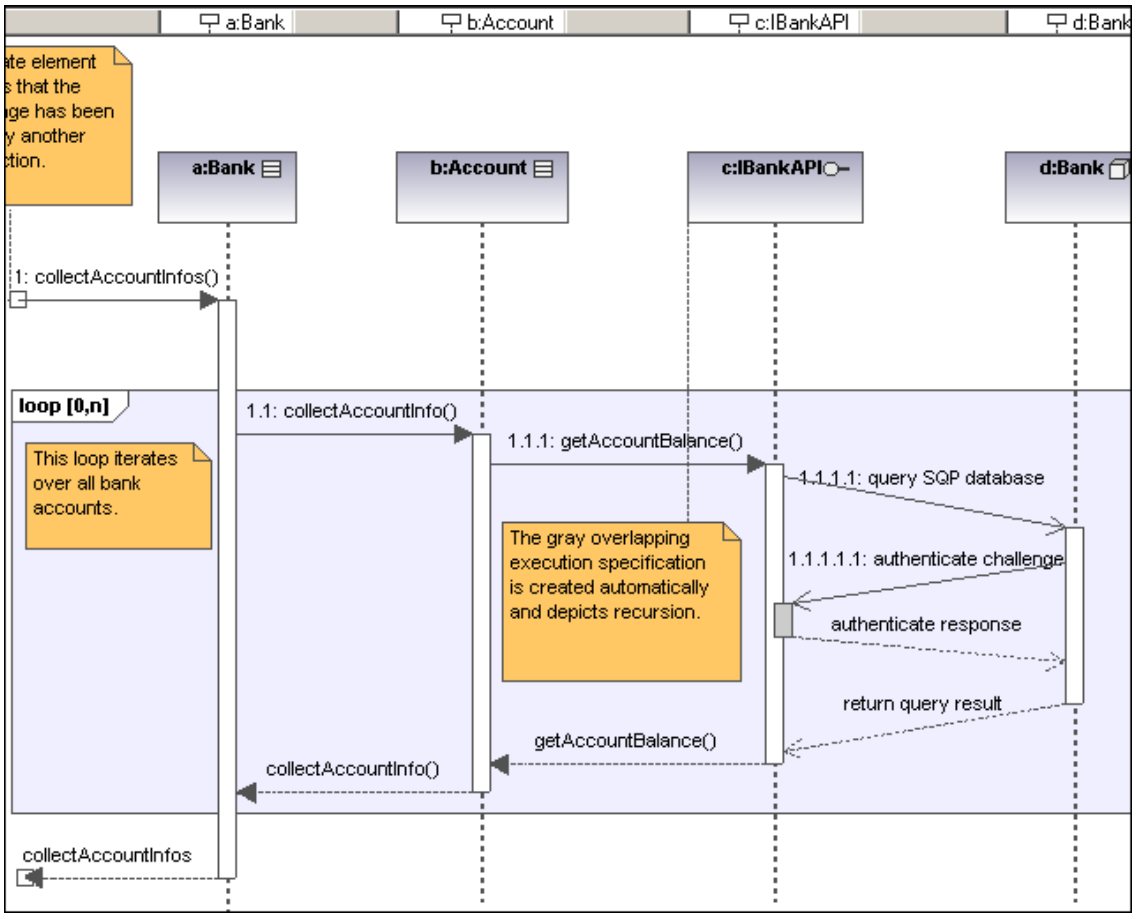
- Los **mensajes** se envían de un objeto a otro, se dibujan en forma de flecha y tienen una etiqueta de texto. Pueden tener un número secuencial y otros atributos opcionales como listas de argumentos, etc. Los mensajes pueden ser condicionales, opcionales y

alternativos. Para más información consulte el apartado Fragmentos combinados.

Esta sección se divide en varios apartados:

- Líneas de vida
- Fragmentos combinados
- Usos de interacción
- Puertas
- Invariantes de estado
- Mensajes

En UModel hay varias maneras de insertar elementos en los diagramas de secuencia.



Usar iconos de la barra de herramientas:

1. Haga clic en un icono de la barra de herramientas Diagrama de secuencia.
2. Haga clic en el área de trabajo del diagrama en el que desea insertar el elemento. Para insertar varios elementos del tipo seleccionado, mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo.

Arrastrar elementos desde la Estructura del modelo hasta el diagrama:

1. En la Estructura del modelo busque el elemento que quiere insertar en el diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).

2. Arrastre el elemento hasta el diagrama de secuencia.

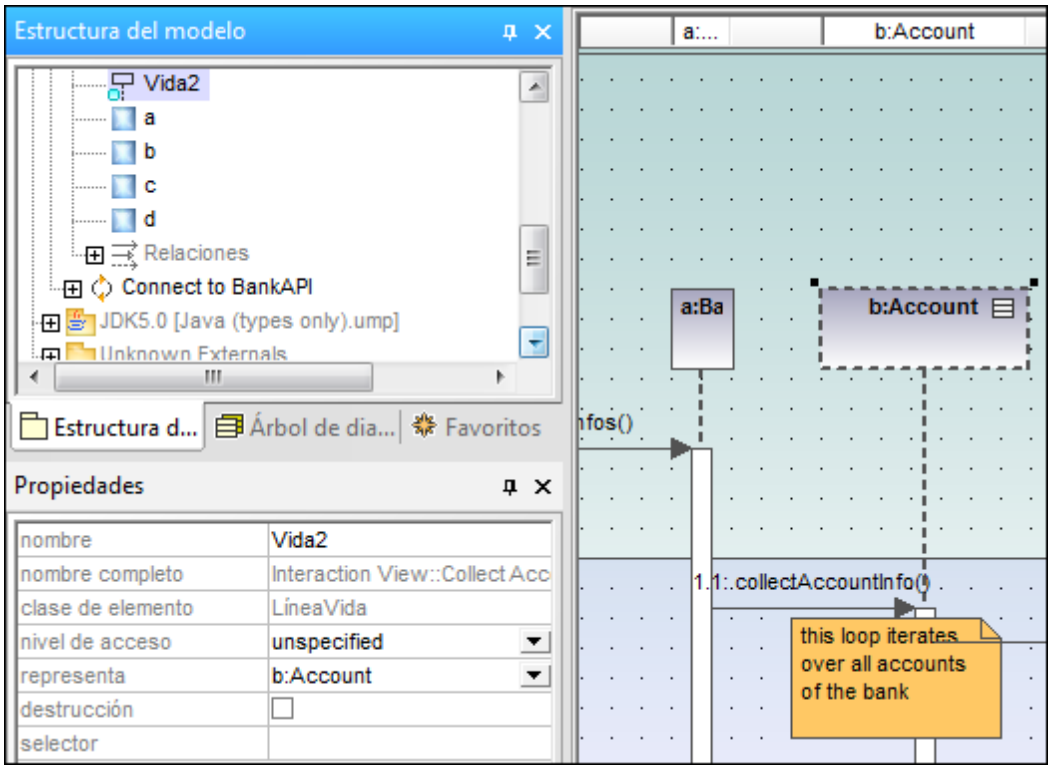
9.1.7.1.1 Líneas de vida

Línea de vida

Una línea de vida es un participante de una interacción. En los diagramas de secuencia de UModel también puede insertar elementos como clases y actores. Estos elementos se representan como una línea de vida nueva.

La etiqueta de la línea de vida aparece en una barra situada en la parte superior del diagrama. Puede cambiar la posición de las etiquetas y también su tamaño. Además puede redefinir el color y el degradado de las etiquetas (en el cuadro combinado `Título - color de degradado` del panel Estilos). Pulse **Ctrl+Entrar** para crear una línea nueva en el nombre de la línea de vida.

En el diagrama de secuencia también puede insertar clasificadores. El campo `representa` del panel Propiedades muestra el tipo de elemento que actúa como línea de vida. Si arrastra una propiedad **con tipo** hasta el diagrama de secuencia, también se crea una línea de vida.



Especificación de ejecución (activación de objetos):
Una especificación de ejecución (activación) se representa en forma de cuadro (rectángulo) en la línea de vida del objeto. Una activación es la ejecución de un procedimiento y el tiempo necesario para ejecutar los procedimientos anidados correspondientes.

Cuando se crea un mensaje entre dos líneas de vida, se crean automáticamente los cuadros de activación.

Y un mensaje recursivo o automensaje (es decir, uno que llama a otro método de la misma clase) crea cuadros de activación apilados.

Para mostrar/ocultar los cuadros de activación:

1. Abra el panel Estilos y desplácese hasta el cuadro combinado Mostrar especificaciones de ejecución.
   En este estilo puede definir si los cuadros de activación se muestran o se ocultan en el diagrama de secuencia.

Atributos de las líneas de vida:
La casilla destrucción sirve para añadir un marcador de destrucción (o freno) a la línea de vida sin necesidad de usar un mensaje de destrucción.

En el campo selector puede insertar una expresión que indique la parte que representa la línea de vida si el `ElementoConectable` tiene más de un valor (es decir, si su multiplicidad es mayor que 1).

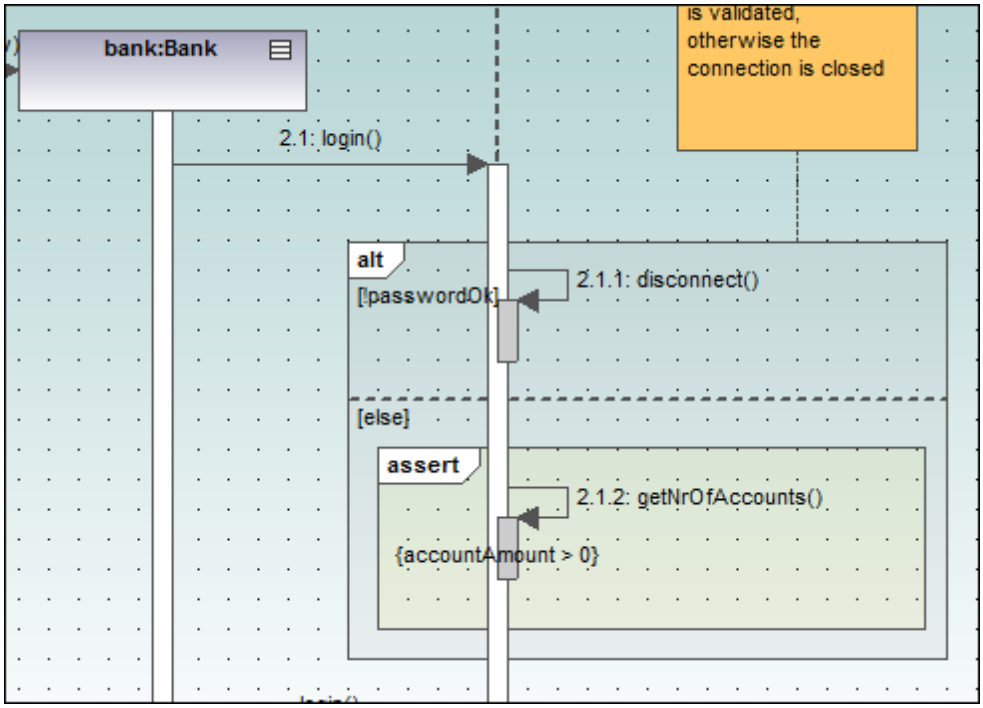
Ir a una línea de vida

Haga clic con el botón derecho en una línea de vida y en el menú contextual elija la opción **Ir a XXX** (XXX es el tipo de línea de vida seleccionada). El elemento se resalta en el panel Estructura del modelo.

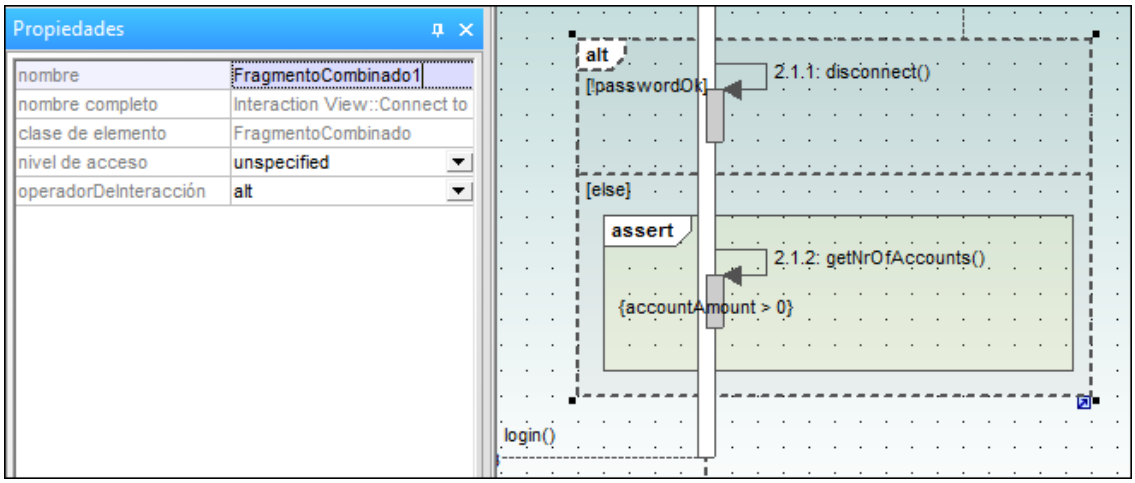
9.1.7.1.2 Fragmentos combinados

FragmentoCombinado

Los fragmentos combinados son subunidades o secciones de una interacción. El operador de interacción que aparece en el pentágono de la esquina superior izquierda define el tipo de fragmento combinado. Por tanto, la restricción define el tipo de fragmento (p. ej. de bucle, alternativo, etc.) utilizado en la interacción.



La barra de herramientas de los diagramas de secuencia también incluye iconos para insertar fragmentos combinados en el diagrama: **`seq`** (secuencia), **`alt`** (alternativo) o **`loop`** (bucle). Haga clic en el cuadro combinado `operadorDeInteracción` para definir el tipo de fragmento de interacción.



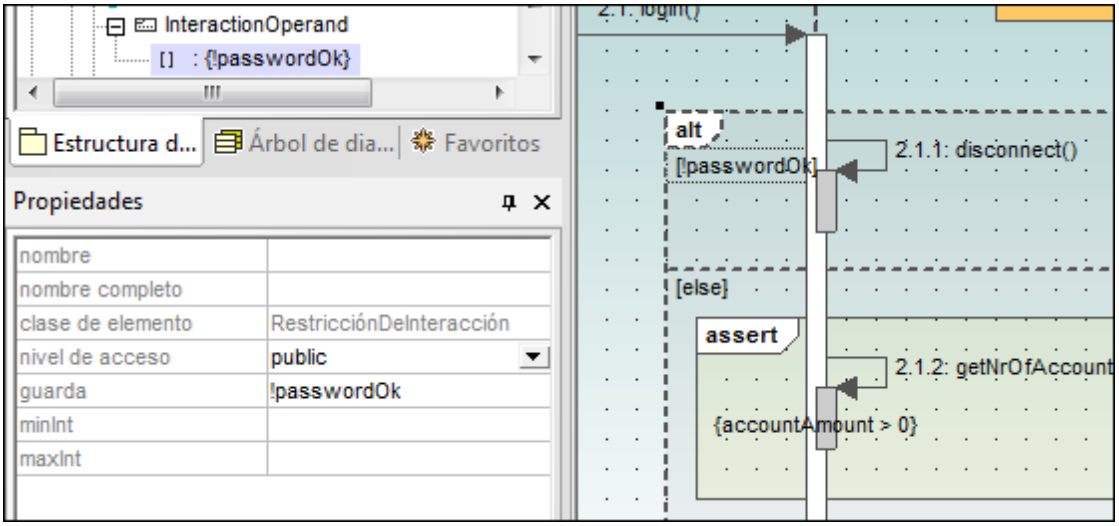
OperadoresDeInteracción

Secuencias débiles **seq** 

El fragmento combinado representa secuencias débiles entre los comportamientos de los operandos.

Alternativas **alt** 
Solo se eligirá uno de los operandos definidos. El operando debe tener una expresión de guarda cuyo resultado sea `true`.

Si uno de los operandos utiliza el guarda "else", el operando se ejecuta si todos los demás guardas devuelven `false`. La expresión de guarda se puede introducir inmediatamente después de la inserción (y aparecerá entre corchetes).



La **RestricciónDeInteracción** es de hecho la expresión de guarda que aparece entre corchetes.

Opción **opt**
Representa una opción entre ejecutar el operando o no hacer nada.

Pausa **break**
El operador **break** se elige cuando el guarda es `true`. El resto del fragmento se ignora.

Paralelo **par**
Indica que el fragmento combinado representa una combinación paralela de operandos.

Secuencias estrictas **strict**
El fragmento combinado representa una secuencia estricta entre los comportamientos de los operandos.

Bucle **loop** 
El operando **loop** se repetirá tantas veces como defina la expresión de guarda.

Tras seleccionar este operando puede editar la expresión directamente (en el pentágono `loop`) haciendo doble clic.

Región crítica **critical**

El fragmento combinado representa una región crítica. La secuencia no se puede interrumpir ni intercalar con otros procesos.

Negativo **neg**
El fragmento no es válido y los demás se suponen válidos.

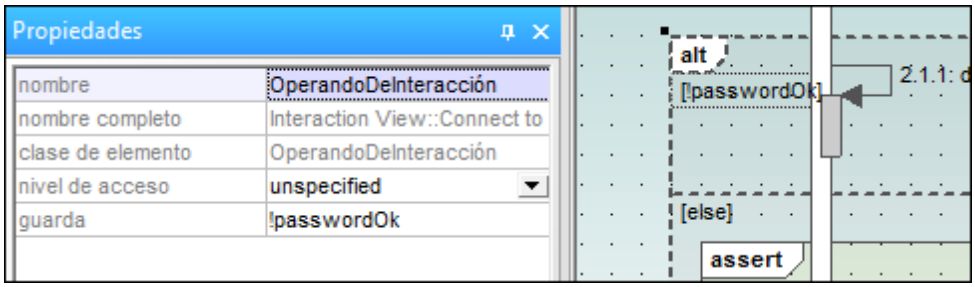
Aserción **assert**
Designa el fragmento combinado válido y sus secuencias. Se suele usar junto con los operandos **consider** o **ignore**.

Ignorar **ignore**
Define qué mensajes deben ignorarse en la interacción. Se suele usar junto con los operandos **assert** o **consider**.

Considerar **consider**
Define qué mensajes se deben tener en cuenta en la interacción. Se suele usar junto con los operandos **assert** o **ignore**.

Agregar operandosDeInteracción a un fragmento combinado:

1. Haga clic con el botón derecho en el fragmento combinado y seleccione **Nuevo/a | OperandoDeInteracción**.
   La condición de guarda se puede editar inmediatamente.
2. Inserte la condición de guarda para el **OperandoDeInteracción** (p. ej. `!passwordOK`) y pulse **Entrar** para confirmar.



   Pulse **Ctrl+Entrar** para crear una línea nueva en el nombre del operando de interacción.
3. Use el mismo método para añadir otro operando de interacción con la condición de guarda "else".
   Los operandos aparecen separados por líneas de puntos en el fragmento.

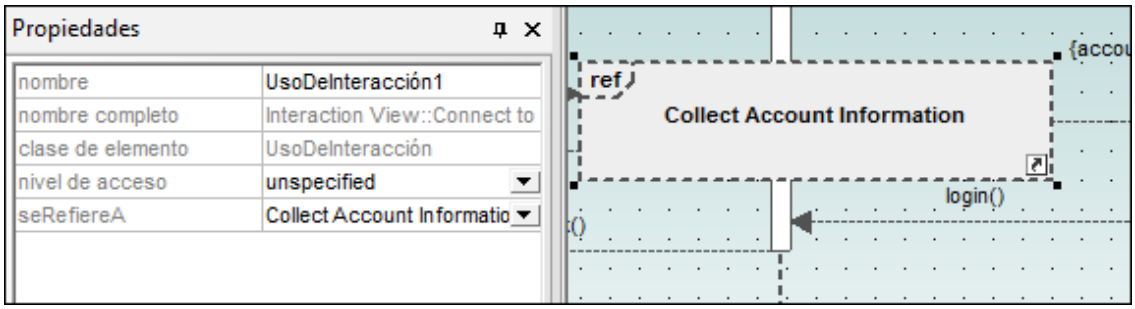
Eliminar operandos de interacción:

1. Haga doble clic en la expresión de guarda del fragmento combinado en el área de trabajo del diagrama (no en el panel Propiedades).
2. Elimine la expresión de guarda y pulse **Entrar** para confirmar.
   Como resultado se elimina la expresión de guarda / el operando de interacción y el tamaño del fragmento combinado se ajusta automáticamente.

9.1.7.1.3 Usos de interacción

UsoDeInteracción

El elemento **UsoDeInteracción** es una referencia a un elemento de interacción y sirve para compartir porciones de una interacción con otras interacciones.



La casilla `seRefiereA` del panel Propiedades sirve para seleccionar la interacción a la que desea hacer referencia. El nombre del uso de interacción seleccionado aparecerá en el elemento.

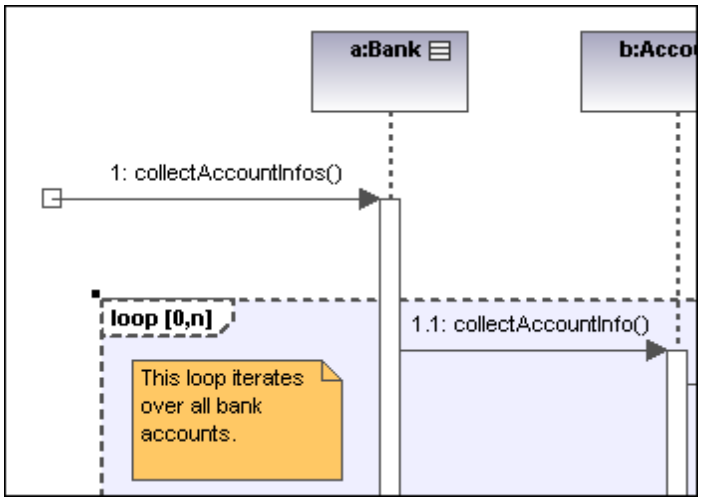
Nota: también puede arrastrar un **`UsoDeInteracción`** desde la Estructura del modelo hasta el área de trabajo del diagrama.

9.1.7.1.4 Puertas

Puerta

Una puerta es un punto de conexión que permite transmitir mensajes de un fragmento a otro de la interacción. Las puertas se conectan por medio de mensajes.

1. Inserte una puerta en el diagrama.
2. Cree un mensaje nuevo, haga clic en la puerta y arrastre el puntero hasta la línea de vida (o desde la línea de vida hasta la puerta).
   Esto conecta los dos elementos. El cuadrado pequeño representa la puerta.



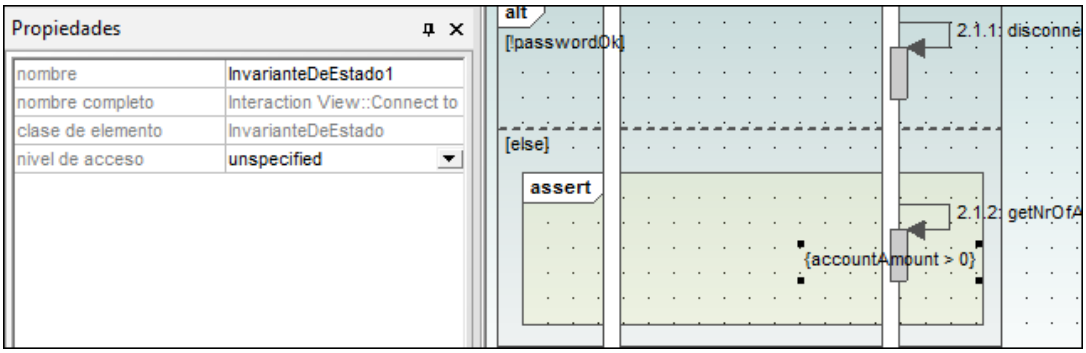
9.1.7.1.5 Invariantes de estado

InvarianteDeEstado

Una invariante de estado es una condición o restricción aplicada a una línea de vida. Para que exista la línea de vida es obligatorio que la condición se cumpla.

Para definir una InvarianteDeEstado:

1. Haga clic en el icono **InvarianteDeEstado** de la barra de herramientas y después en la línea de vida o en la activación de objetos.
2. Inserte la condición/restricción que desea aplicar (p. ej. `accountAmount > 0`) y pulse **Entrar** para confirmar.



9.1.7.1.6 Mensajes

Las líneas de vida envían y reciben mensajes que se representan en forma de flechas etiquetadas. Los mensajes pueden estar numerados de forma secuencial y tener atributos opcionales (como listas de argumentos, etc.). Los mensajes se presentan de arriba a abajo, es decir, el eje vertical es el componente de tiempo del diagrama de secuencia.

- Una **llamada** es una comunicación síncrona o asíncrona que invoca una operación que permite al control volver al objeto remitente. La flecha de una llamada apunta a la parte superior de la activación que inicia la llamada.
- La **recursión** (o llamada a otra operación del mismo objeto) se representa apilando recuadros de activación (`EspecificacionesDeEjecución`).

Para insertar un mensaje:

1. En la barra de herramientas Diagrama de secuencia haga clic en el icono del mensaje que desea insertar.
2. Haga clic en la línea de vida o cuadro de activación del objeto remitente.
3. Arrastre la línea del mensaje hasta la línea de vida o el cuadro de activación del objeto destinatario.
   Si el mensaje se puede colocar en una línea de vida, esta se resalta.

- La dirección en la que se arrastra la flecha define la dirección del mensaje. Los mensajes de respuesta pueden apuntar en ambas direcciones.
- Mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo para insertar varios mensajes.

- En los objetos remitentes/destinatarios se crean automáticamente cuadros de activación. Para ajustar el tamaño de los cuadros a mano haga clic en los controladores de tamaño y arrástrelos.
- La secuencia de numeración se actualiza, dependiendo de las opciones de numeración que estén activas.

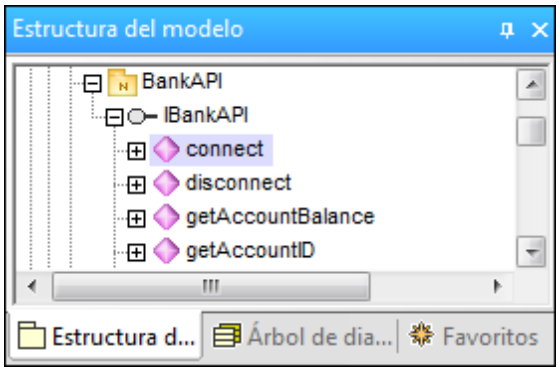
Para eliminar un mensaje:

1. Haga clic en el mensaje.
2. Pulse la tecla **Supr** para eliminar el mensaje del modelo. Para eliminarlo del diagrama solamente, haga clic con el botón derecho en el mensaje y elija **Eliminar solo en el diagrama**.
   La numeración de los mensajes y los cuadros de activación de los demás objetos se actualizan.

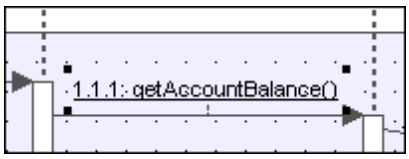
"Ir a la operación" para mensajes de llamada:

En los diagramas de secuencia y de comunicación puede buscar las operaciones a las que hacen referencia los mensajes de llamada.

1. Haga clic con el botón derecho en un mensaje de llamada y elija la opción **Ir a la operación**.
   La operación aparece resaltada en el panel Estructura del modelo.



Nota: los nombres de operaciones estáticas aparecen subrayados.



Para cambiar la posición de mensajes dependientes:

1. Haga clic en el mensaje que quiere mover y arrástrelo hasta su nueva posición. Cuando cambia la posición de un mensaje, UModel mueve también los mensajes dependientes relacionados con el mensaje activo.

   Para seleccionar varios mensajes utilice **Ctrl+clic**.

Para cambiar la posición de los mensajes uno por uno:

1. Desactive el icono **Activar/desactivar el movimiento de mensajes dependientes**

 de la barra de herramientas.

2. Haga clic en el mensaje que desea mover y arrástrelo hasta su nueva posición.
   En este caso solamente se mueve el mensaje seleccionado. Este mensaje se puede colocar en cualquier posición del eje vertical entre las líneas de vida de los objetos.

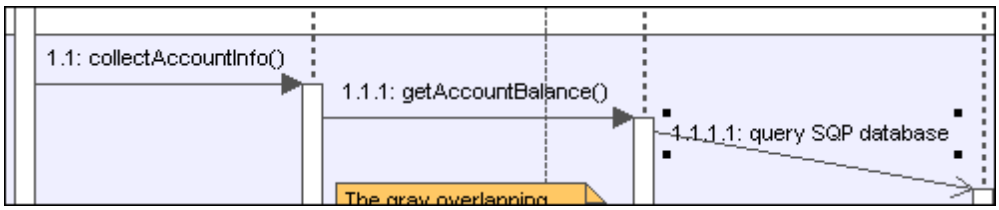
Para crear mensajes de respuesta automáticamente:

1. Active el icono **Activar/desactivar la creación automática de respuestas para mensajes**  .
2. Cree un mensaje nuevo entre dos líneas de vida.
   UModel inserta automáticamente un mensaje de respuesta.

Numeración de los mensajes:

Puede usar tres tipos de numeración para los mensajes: numeración anidada, numeración sencilla o ninguna numeración.

- **Sin numeración**  : este icono elimina la numeración de todos los mensajes.
- **Sencilla**  : asigna una secuencia numérica a todos los mensajes de arriba a abajo, es decir, en el orden en el que aparecen a lo largo del eje temporal del diagrama.
- **Anidada**  : utiliza la notación decimal, que permite ver la estructura jerárquica de los mensajes del diagrama. La secuencia de numeración es una lista de números secuenciales separada por puntos, seguidos de dos puntos y del nombre del mensaje.



Para seleccionar el tipo de numeración:

Hay dos formas de seleccionar el tipo de numeración:

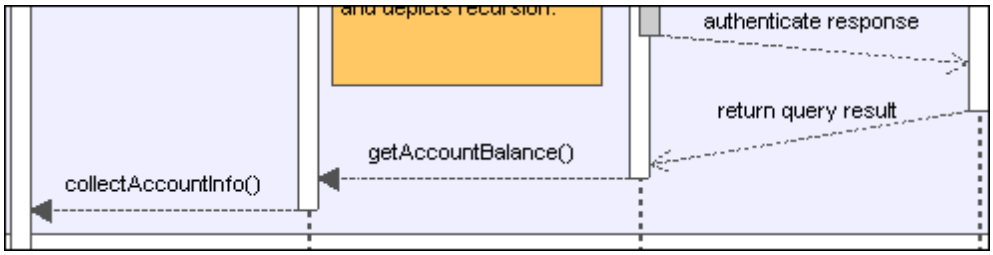
- Con el icono correspondiente de la barra de herramientas Diagrama de secuencia.
- En el panel Estilos.

Para seleccionar el tipo de numeración en el panel Estilos:

1. Haga clic en la pestaña *Estilos* y desplácese hasta el campo `Numeración de los mensajes`.
2. Haga clic en el cuadro combinado y seleccione el tipo de numeración en la lista desplegable.
   La opción de numeración seleccionada aparece en el diagrama de secuencia automáticamente.

   Nota: en ocasiones el tipo de numeración no numera todos los mensajes correctamente si existen ambigüedades. Si esto ocurre, puede solucionar el problema añadiendo mensajes de respuesta.

Mensajes de respuesta:
Los mensajes de respuesta se representan con flechas de línea discontinua.

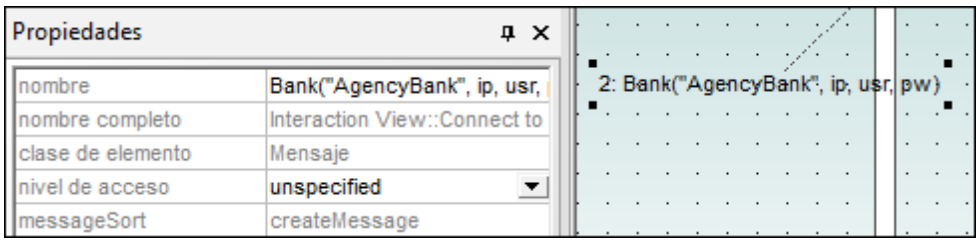


Por lo general, los mensajes de respuesta vienen señalados por la parte inferior del cuadro de activación. Si los cuadros de activación están deshabilitados (panel Estilos, `Mostrar especificaciones de ejecución=false`), entonces se recomienda usar flechas de respuesta para evitar ambigüedades.

Si activa el icono **Activar/desactivar la creación automática de respuestas para mensajes**  , cada vez que cree un mensaje de llamada entre dos líneas de vida/cuadros de activación UModel creará mensajes de respuesta sintácticamente correctos de forma automática.

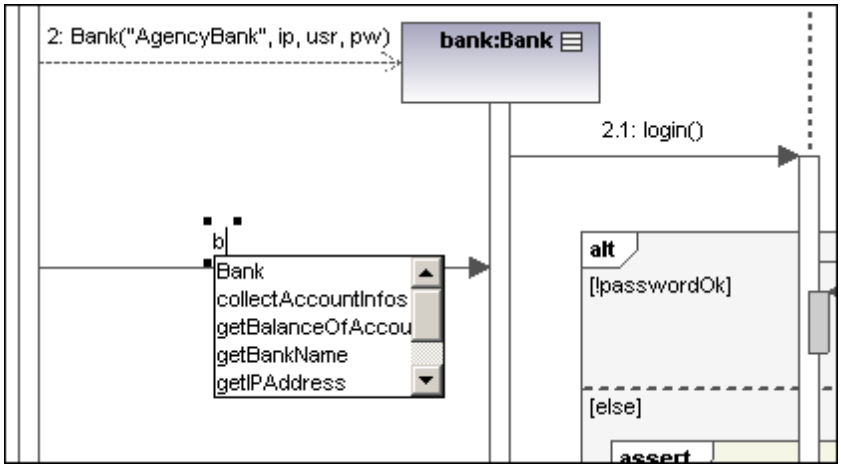
Para crear objetos con mensajes:

1. Los mensajes pueden crear objetos nuevos. Esto se hace con el icono **Mensaje (Creación)**  .
2. Arrastre la flecha del mensaje hasta la línea de vida de un objeto para crear ese objeto. Este tipo de mensaje termina en la mitad del rectángulo del objeto y a menudo pone el cuadro del objeto en posición vertical.

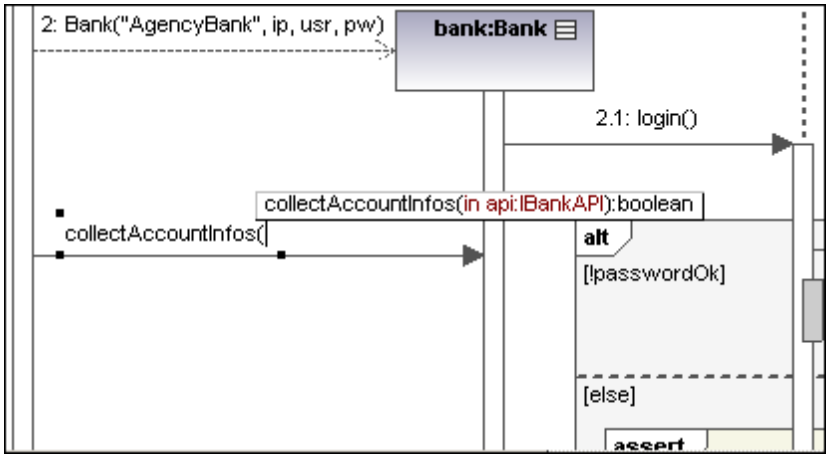


Para enviar mensajes a métodos/operaciones de clases de diagramas de secuencia:
Tras insertar una clase de la Estructura del modelo en un diagrama de secuencia, puede crear un mensaje entre una línea de vida y un método de la clase destinataria (línea de vida). Para ello puede usar la ayuda sintáctica y de las funciones de finalización automática de UModel.

1. Cree un mensaje entre dos líneas de vida (el objeto destinatario debe ser una línea de vida de una clase).
   En cuanto suelte la flecha del mensaje, el nombre del mensaje se resalta automáticamente.
2. Escriba un carácter (p. ej. "b").
   Aparece una ventana emergente que enumera los métodos de clase ya existentes.



3. Seleccione una operación de la lista y pulse **Entrar** para confirmar (p. ej. `collectAccountInfos`).
4. Pulse la barra espaciadora y después la tecla **Entrar** para seleccionar el paréntesis que sugiere automáticamente UModel.
   Ahora aparece una ventana de ayuda sintáctica, que le ayuda a insertar correctamente el parámetro.



Crear operaciones en clases referenciadas:

Si activa el icono **Activar/desactivar la creación automática de operaciones en el destino al escribir el nombre de la operación**  , cada vez que cree un mensaje y escriba un nombre (p. ej. `miOperación()`) UModel creará automáticamente la clase correspondiente en la clase referenciada.

Nota: solamente se pueden crear operaciones automáticamente cuando la línea de vida hace referencia a una clase, a una interfaz...

Iconos de la barra de herramientas para trabajar con mensajes:

 **Mensaje (Llamada)**

 **Mensaje (Respuesta)**

 **Mensaje (Creación)**

 **Mensaje (Destrucción)**

 **Mensaje asíncrono (Llamada)**

 **Mensaje asíncrono (Respuesta)**

 **Mensaje asíncrono (Destrucción)**

 **Activar/desactivar el movimiento de mensajes dependientes**

 **Activar/desactivar la creación automática de respuestas para mensajes**

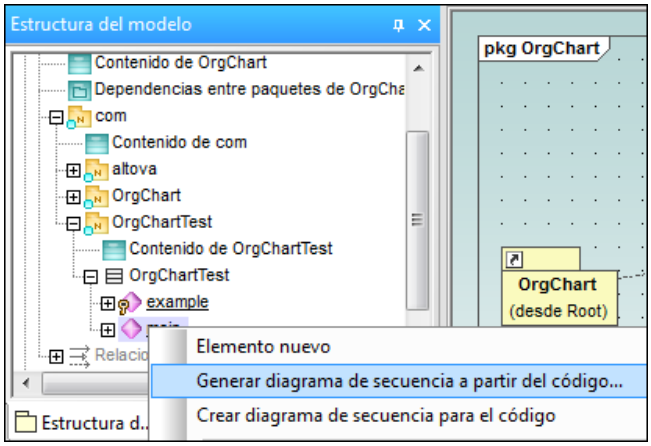
 **Activar/desactivar la creación automática de operaciones en el destino al escribir el nombre de la operación**

9.1.7.2 Generar diagramas de secuencia a partir de código fuente

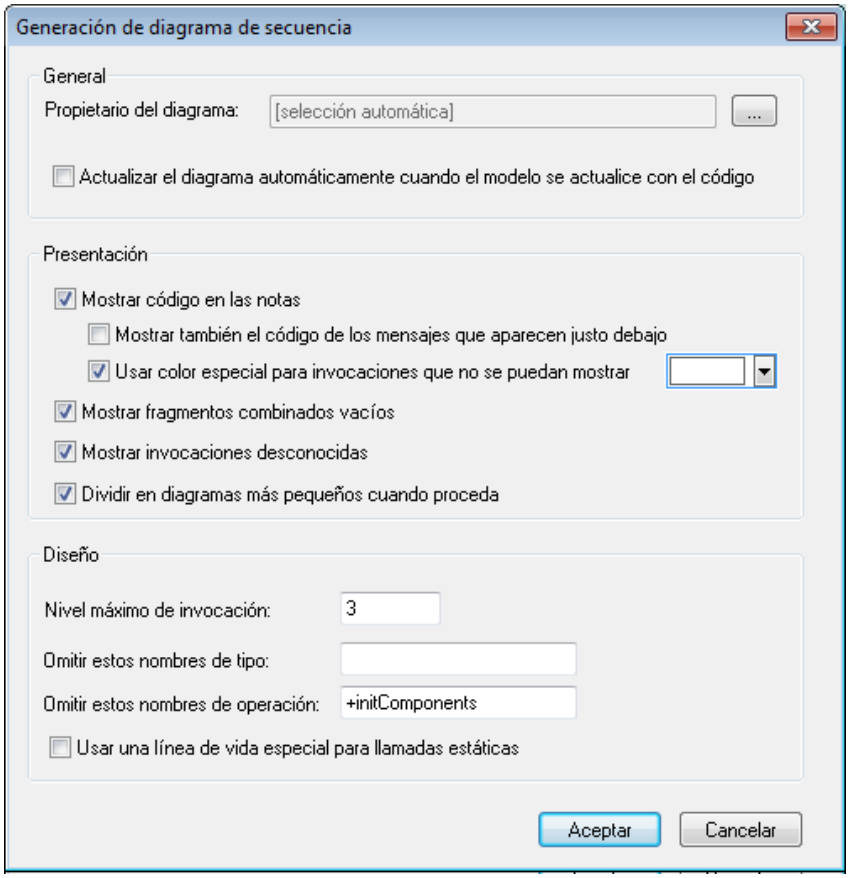
UModel puede modelar diagramas de secuencia a partir de código.

Con el siguiente ejemplo puede aprender a crear un diagrama de secuencia a partir de un método automáticamente. Si quiere, puede hacer lo mismo con su propio código. El método está en el paquete `orgChart`, que se importó al modelo con el comando **Proyecto | Importar directorio de código fuente** (para más información consulte el apartado Ingeniería de ida y vuelta (código - modelo - código).

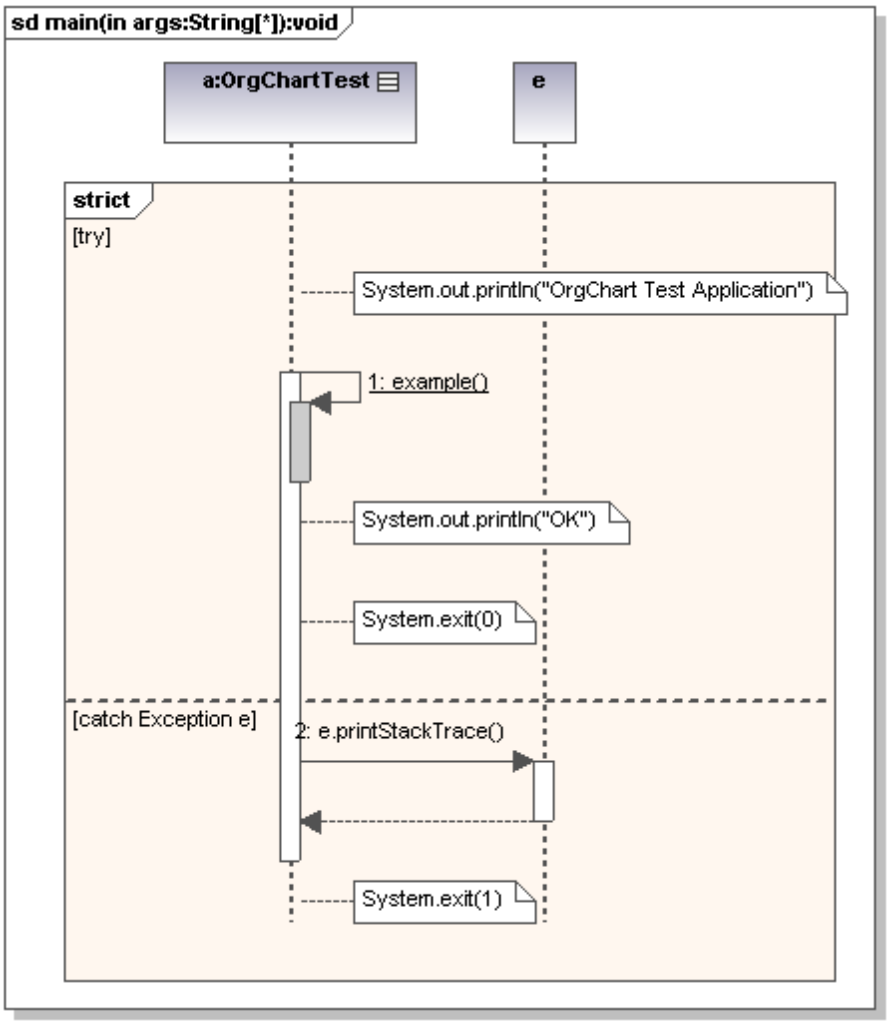
1. Una vez importado el código, haga clic con el botón derecho en el método `main` de la clase `orgChartTest` en la Estructura del modelo. Ahora elija el comando **Generar diagrama de secuencia a partir del código** en el menú contextual.



Esto abre el cuadro de diálogo "Generación de diagrama de secuencia", donde puede configurar la generación.



2. Seleccione las opciones de presentación y diseño y después haga clic en **Aceptar**. La imagen siguiente muestra el diagrama de secuencia que se genera con las opciones seleccionadas en la imagen anterior.

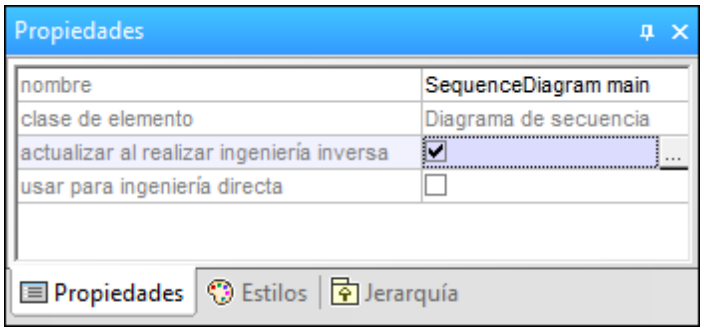


Notas:

- Puede asignar un color distinto a las invocaciones que no se puedan mostrar.
- El nivel máximo de invocación define el nivel de recursión que se debe usar en el diagrama.
- La opción *Omitir estos nombres de tipo:* sirve para crear una lista delimitada por comas de los tipos que no deben aparecer en el diagrama de secuencia generado.
- La opción *Omitir estos nombres de operación:* sirve para crear una lista delimitada por comas de las operaciones que no deben aparecer en el diagrama de secuencia generado (**`initComponents`** se añade automáticamente a la lista).

  Al añadir nombres de operación a la lista (p. ej. **`InitComponents`**), se omite la operación completa. Si añade el carácter + delante del nombre de la operación (p. ej. **`+InitComponent`**), las llamadas de la operación aparecen en el diagrama, pero sin su contenido.
- La opción *Dividir en diagramas más pequeños cuando proceda* divide automáticamente los diagramas de secuencia en varios subdiagramas y genera hipervínculos entre ellos para poder navegar por ellos con facilidad.

Los diagramas de secuencia se actualizan automáticamente cuando se actualiza el proyecto de UModel entero. Si, por el contrario, solo se actualizan las clases o algunos archivos desde una aplicación externa, los diagramas no se actualizan. No obstante, esto puede configurarse marcando la casilla del campo `Actualización de código automática`.



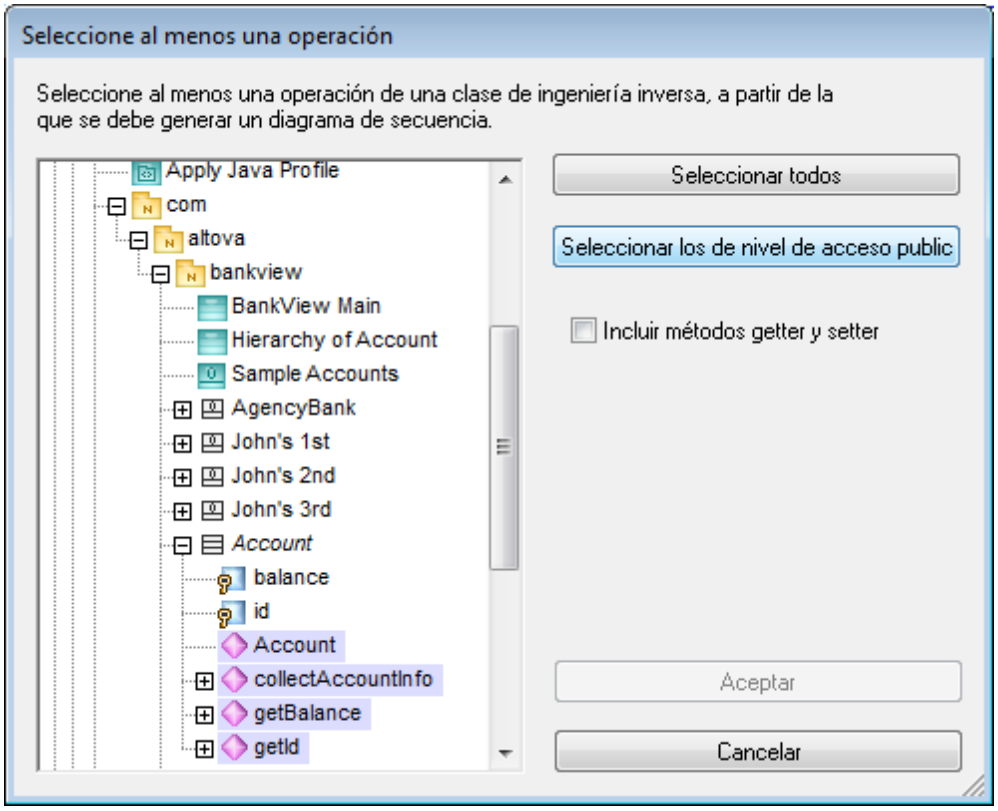
Haga clic en el icono ... del campo `Actualización de código automática` para abrir el cuadro de diálogo "Generación de diagrama de secuencia" y cambiar las opciones de generación.

9.1.7.3 Generar varios diagramas de secuencia a partir de propiedades

Además de crear diagramas de secuencia a partir de operaciones, UModel puede crear diagramas de secuencia a partir de propiedades Getter/Setter.

Crear varios diagramas de secuencia a partir de varias operaciones:

1. Seleccione la opción de menú **Proyecto | Generar diagramas de secuencia a partir del código**.
   Aparece el cuadro de diálogo "Seleccione al menos una operación".

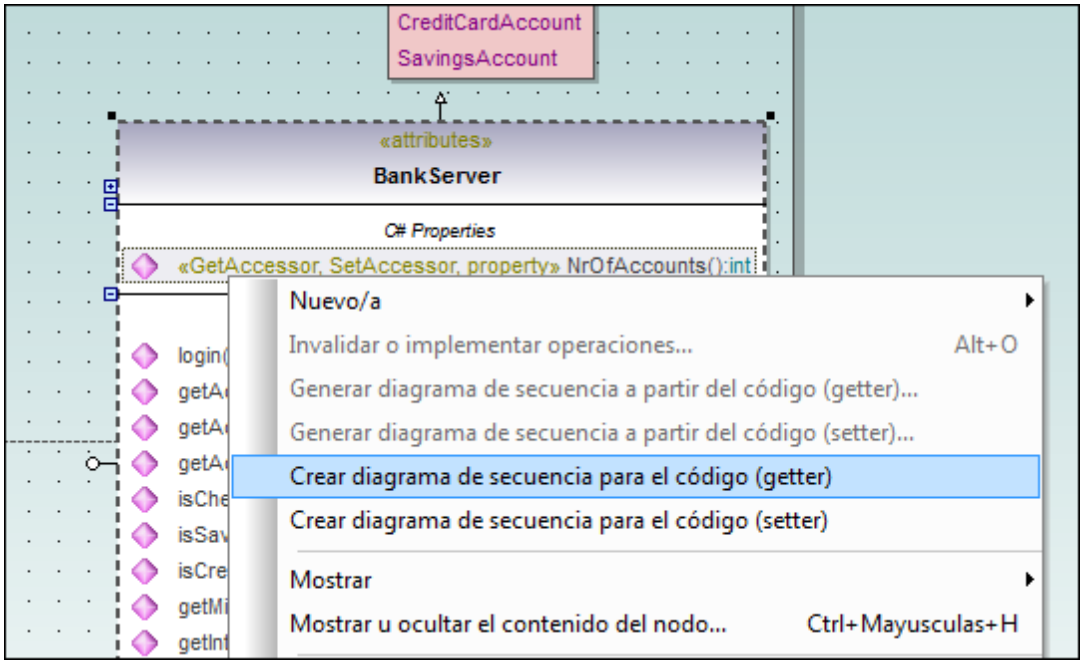


2. Seleccione las operaciones para las que desea generar un diagrama de secuencia y haga clic en **Aceptar** (si quiere puede usar los botones **Seleccionar todos** y **Seleccionar los de nivel de acceso public**). Al hacer clic en **Aceptar** se abre el cuadro de diálogo "Generación de diagrama de secuencia", donde puede elegir las opciones de generación.
3. Haga clic en **Aceptar** para generar los diagramas de secuencia. Por cada operación seleccionada se genera un diagrama de secuencia distinto.

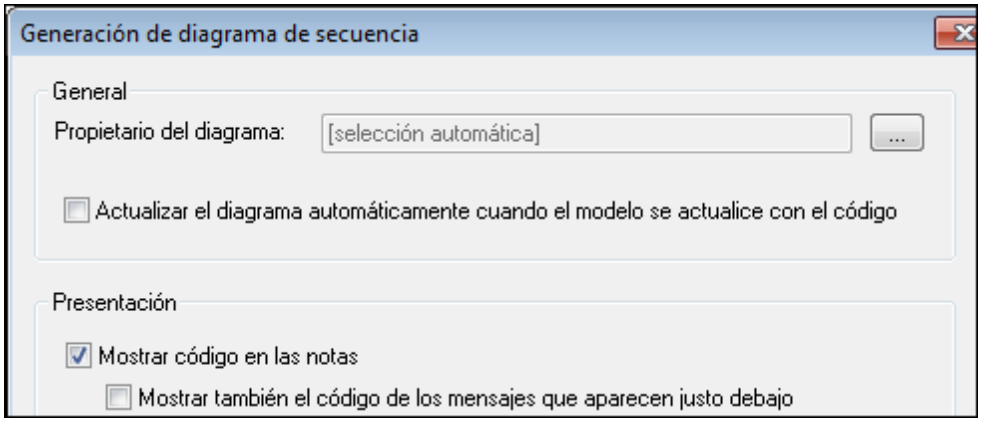
   Nota: cuando genere diagramas de secuencia también puede elegir si se incluyen o excluyen los getter y setter.

Crear un diagrama de secuencia a partir de propiedades getter/setter (C#, VB .NET)

1. Haga clic con el botón derecho en una operación que tenga el estereotipo `GetAccessor/SetAccessor`.



2. Seleccione la opción del menú contextual correspondiente (p. ej. **Crear diagrama de secuencia para el código (getter)**).
   Esto abre el cuadro de diálogo "Generación de diagrama de secuencia", donde puede configurar la presentación del diagrama de secuencia que se generará.



3. Haga clic en **Aceptar** para generar el diagrama de secuencia.

9.1.7.4 Generar código a partir de diagramas de secuencia

UModel puede crear código a partir de un diagrama de secuencia que esté vinculado a una operación o a varias.

A partir de diagramas de secuencia puede generar código para:

- VB.NET, C# y Java.
- UModel y la edición Eclipse y Visual Studio de UModel.
- las tres ediciones de UModel.

Hay dos maneras de crear código a partir de diagramas de secuencia:

- Comenzando con una operación creada con ingeniería inversa (consulte el apartado Generar diagramas de secuencia a partir de código fuente).
- Creando desde cero un diagrama de secuencia **nuevo** que esté vinculado a una operación (haciendo clic con el botón derecho en la Estructura del modelo y seleccionando **Crear diagrama de secuencia para el código**).

Nota: cuando use como base un diagrama de secuencia creado por ingeniería inversa, compruebe que la opción *Mostrar código en las notas* esté activa durante el proceso de ingeniería inversa. Así no perderá código cuando vuelva a iniciar el proceso de ingeniería directa.

Esto se debe a que UML no puede mostrar todas las características de los lenguajes VB.NET, Java y C# en el diagrama de secuencia y las características que no puede mostrar se presentan como notas.

Para agregar texto sin formato como código durante la creación de diagramas de secuencia:

1. Anexe una nota a una línea de vida del diagrama de secuencia.
2. Escriba el código que se debe escribir en el código fuente final.
   Marque la casilla `Es código` (panel Propiedades) de la nota para poder acceder a ella.

   Para ver un ejemplo consulte el apartado Agregar código a los diagramas de secuencia.

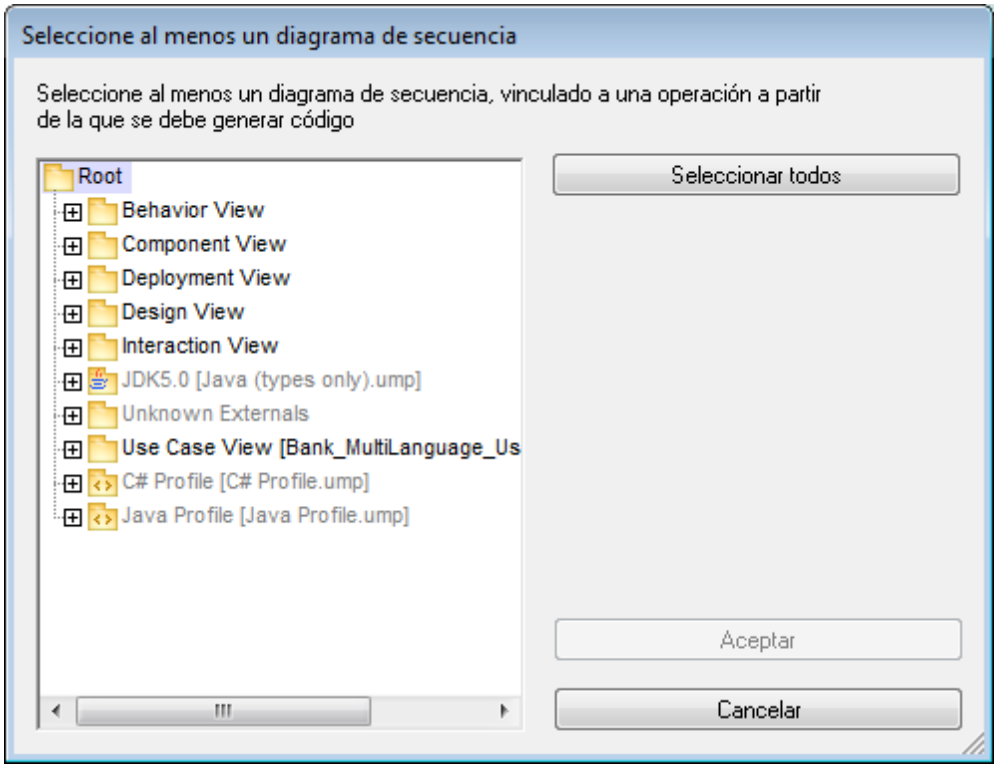
Si quiere que un diagrama de secuencia se utilice automáticamente para ingeniería de código cada vez que se inicie la ingeniería de código:

- Active la casilla `Usar para ingeniería directa` del panel Propiedades.

Cuando se crea código por ingeniería directa a partir de un diagrama de secuencia, siempre se pierde código antiguo porque lo sobrescribe el código nuevo.

Menú Proyecto:

1. Seleccione la opción de menú **Proyecto | Generar código a partir de diagramas de secuencia**.
   Aparece un diálogo donde debe seleccionar el diagrama de secuencia.



Con el botón **Seleccionar todos** se seleccionan todos los diagramas de secuencia del proyecto de UModel.

2. Haga clic en **Aceptar** para generar el código.
   La ventana Mensajes muestra el estado del proceso de generación de código.

Panel Estructural del modelo:

- Haga clic con el botón derecho en un diagrama de secuencia y elija **Generar código a partir del diagrama de secuencia** en el menú contextual.

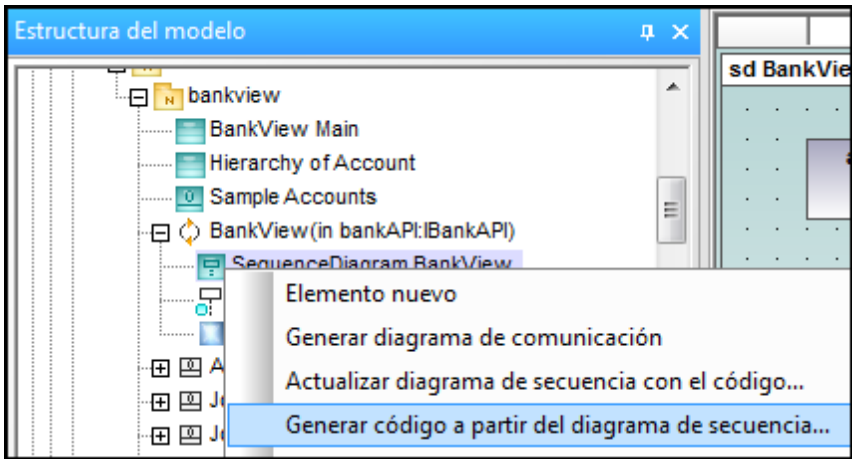
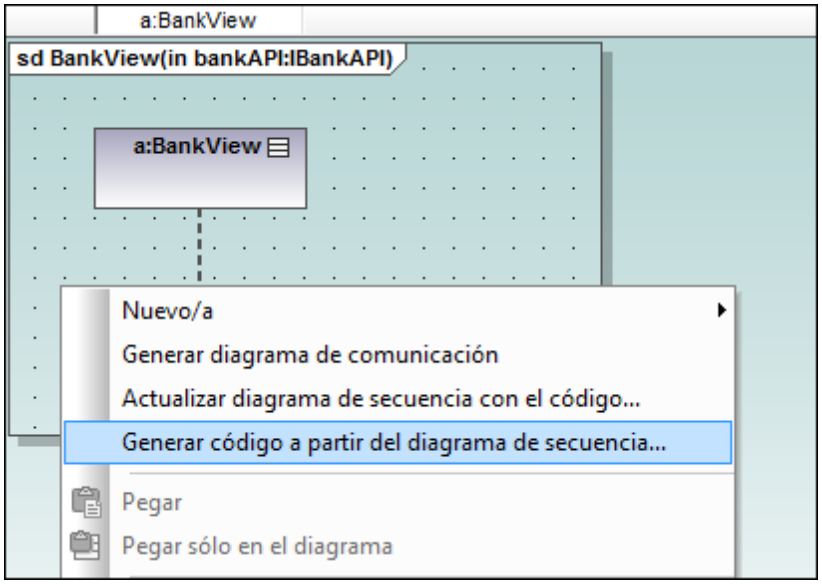


Diagrama de secuencia con código de una operación:

1. Haga clic con el botón derecho en el fondo del diagrama de secuencia que contiene el código de una operación.

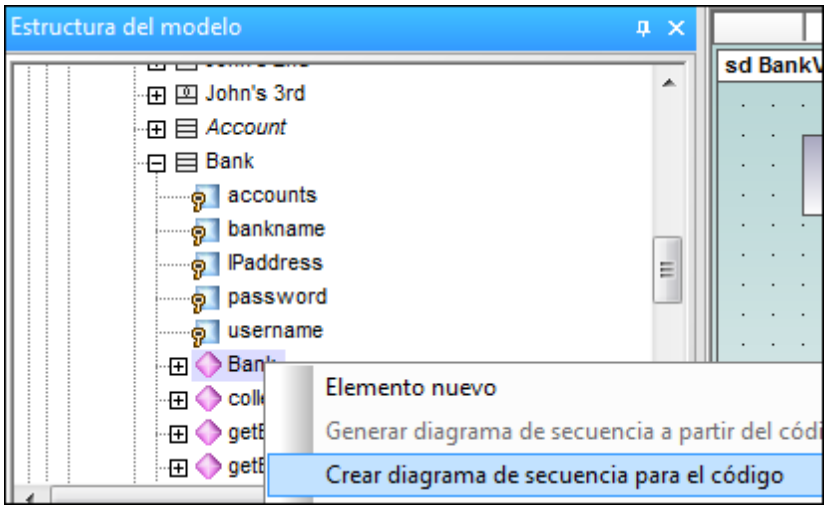
2. Elija la opción **Generar código a partir del diagrama de secuencia**.



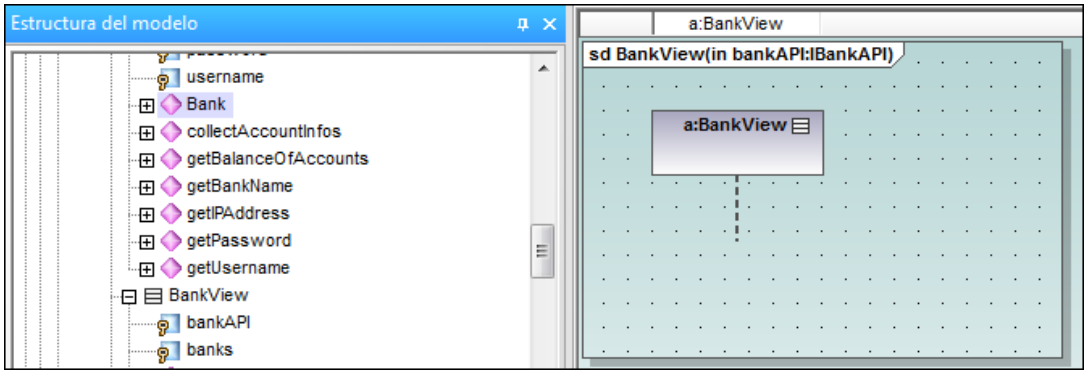
Este comando inicia el proceso de ingeniería directa.

Para crear un diagrama de secuencia para el código (ingeniería):

1. En la Estructura del modelo haga clic con el botón derecho en una operación y elija la opción **Crear diagrama de secuencia para el código**.



UModel le pregunta si quiere usar el diagrama nuevo para la ingeniería directa.



El resultado es un diagrama de secuencia nuevo que contiene la línea de vida de esa clase.

9.1.7.4.1 Agregar código a diagramas de secuencia

En UModel puede generar código a partir de diagramas de secuencia nuevos y de diagramas de secuencia generados por ingeniería inversa, pero solo si el diagrama está vinculado a la operación principal.

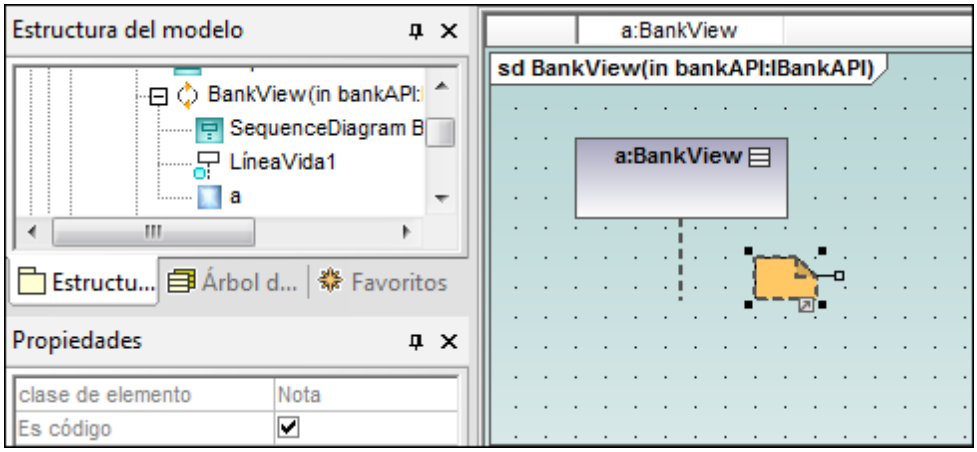
Cuando se aplica ingeniería inversa a código, los elementos estándar de los diagramas de secuencia (p. ej. los `FragmentosCombinados`) se asignan a los elementos del código (p. ej. instrucciones `if`, bucles, etc.).

Para las instrucciones de programación que no tengan elementos equivalentes en el diagrama de secuencia (p. ej. `i = i+1`), UModel utiliza las notas del código y añade código a los diagramas. Estas notas se deben vincular a la línea de vida.

Recuerde que UModel no revisa ni analiza estos fragmentos de código. Por eso es importante comprobar que los fragmentos de código son correctos y se podrán compilar.

Para agregar código a un diagrama de secuencia:

1. Haga clic en el icono **Nota**  y después en el elemento de modelado donde desea insertar la nota (p. ej. `FragmentoCombinado`).
2 Escriba el fragmento de código dentro de la nota (p. ej. `return`).
3. Haga clic en el controlador `Enlace de nota` de la nota que acaba de insertar y arrastre el cursor hasta la línea de vida.
4. Marque la casilla `Es código` en el panel Propiedades para incluir este fragmento de código cuando UModel genere código.



Cuando seleccione una nota de un diagrama de secuencia que se **pueda** usar para generación de código, la propiedad `Es código` aparece en la ventana Propiedades. Esta propiedad permite alternar entre notas normales y corrientes y notas para generación de código.

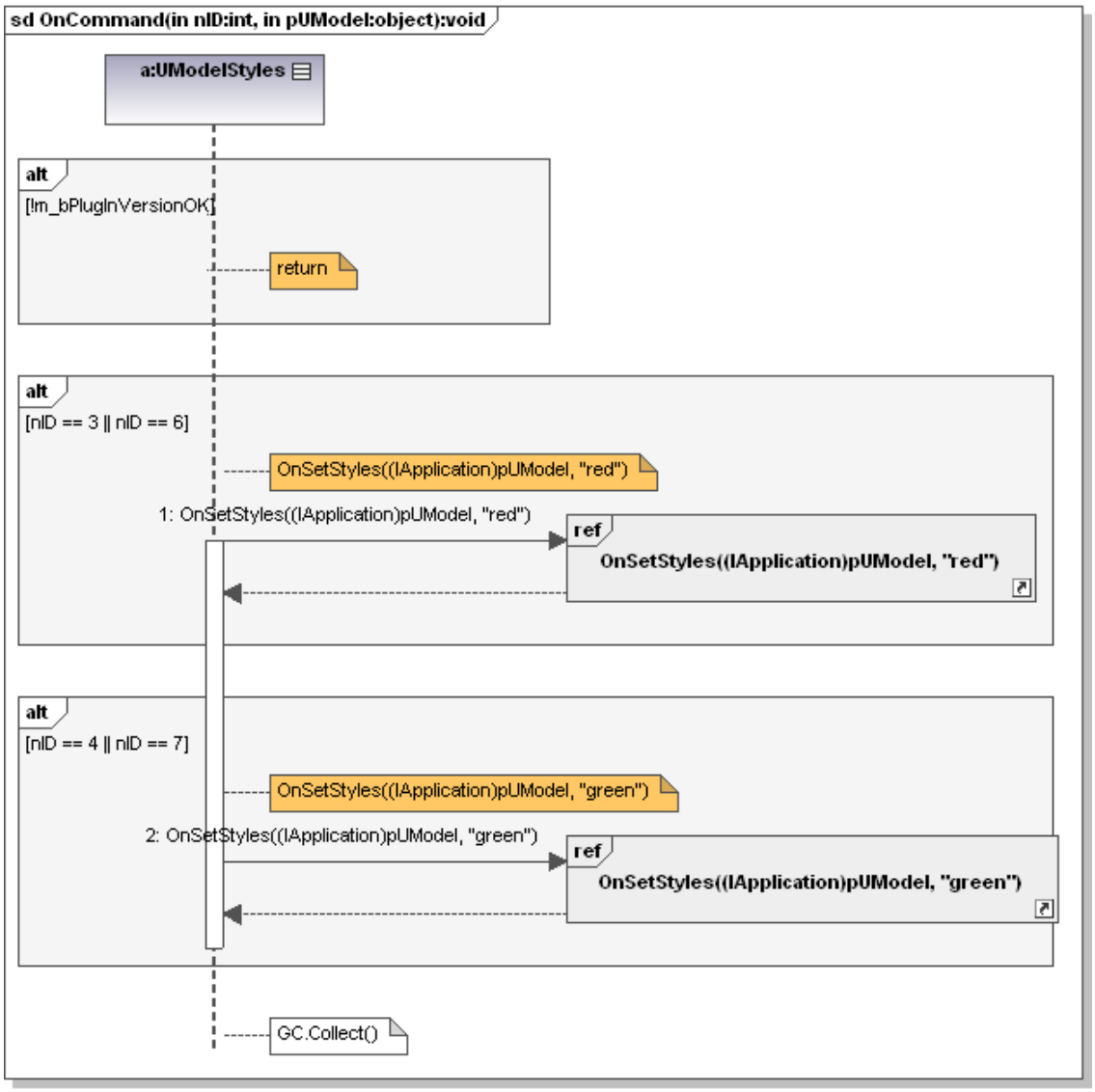
Notas normales y corrientes: 

Notas para generación de código: 

(la pestaña de la esquina superior derecha es más oscura)

Las actualizaciones de código tienen lugar automáticamente en cada proceso de ingeniería directa si está activa la casilla `Usar para ingeniería directa`. Si se realizaron cambios en el diagrama de secuencia, el código de la operación se sobrescribe siempre.

El diagrama de secuencia que aparece más abajo se generó haciendo clic con el botón derecho en la operación **`onCommand`** y seleccionando la opción **Generar diagrama de secuencia a partir del código**. El código C# de este ejemplo está disponible en la carpeta `c:\Documents and Settings\<usuario>\Mis Documentos\Altova\UModel2012\UModelExamples\IDEPlugIn\Styles`. Utilice la opción de menú **Proyecto | Importar proyecto de código fuente** para importar el proyecto.



El código que aparece a continuación se generó a partir del diagrama de secuencia.

```
Public void OnCommand(int nID, object pUModel)
{
   //Generated by UModel. This code will be overwritten when you re-run code
generation.
   if (!m_bPlugINVersionOK)
   {
   return;
   }

   if (nID == 3 || nID == 6)
   {
   OnSetStyles((IApplication)pUModel, "red");
   }

   if (nID == 4 || nID == 7)
   {
```

```
    OnSetStyles((IApplication)pUModel, "green");
    }
    GC.Collect();
}
```

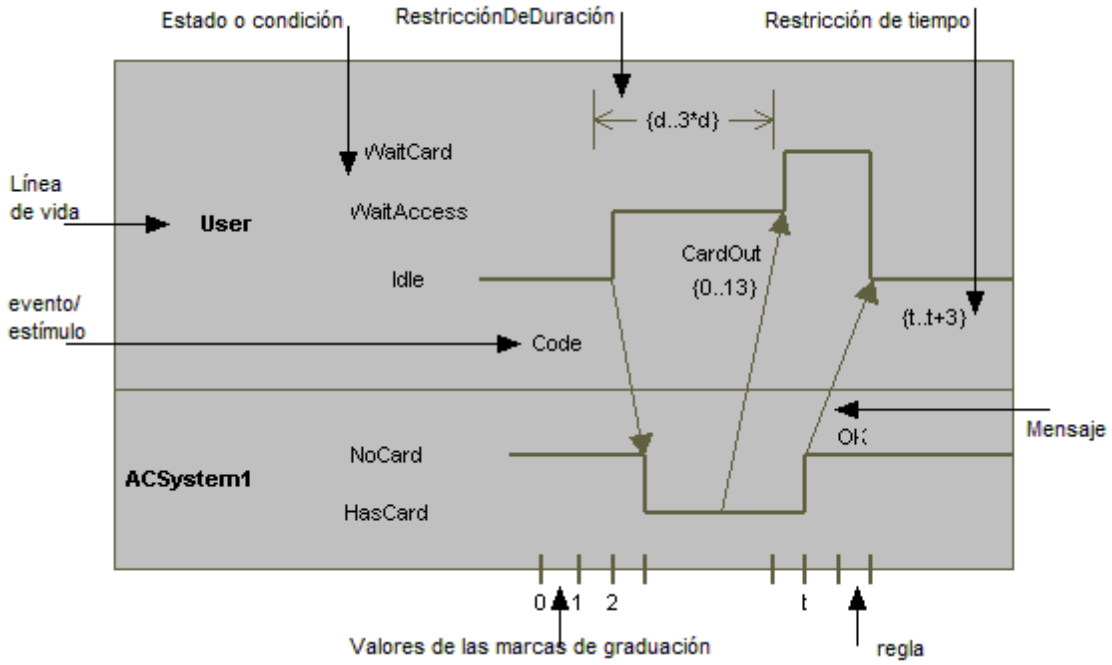
9.1.8 Diagrama de ciclo de vida

Sitio web de Altova: Diagramas de ciclo de vida UML

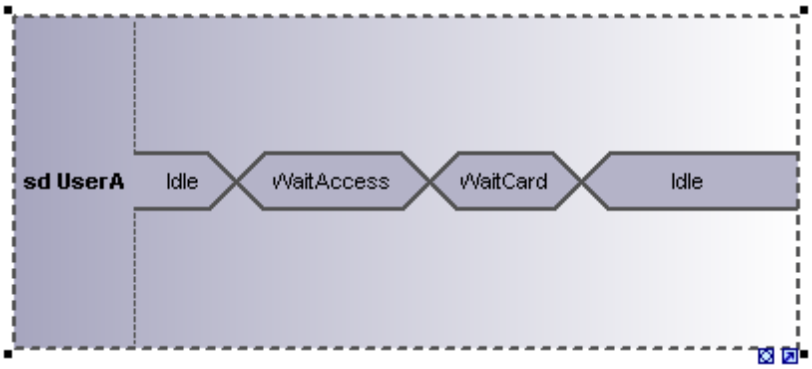
Los diagramas de ciclo de vida modelan los cambios de estado o la condición de los objetos que interactúan entre sí a lo largo de un período de tiempo. Los estados o condiciones se representan como escalas de tiempo que responden a eventos de mensaje y las líneas de vida representan instancias de clasificador y roles clasificador.

Los diagramas de ciclo de vida son un tipo especial de diagrama de secuencia. La diferencia es que los ejes están invertidos, es decir, el tiempo aumenta de izquierda a derecha, y las líneas de vida aparecen por separado en compartimentos apilados verticalmente.

Además, este tipo de diagramas suelen utilizarse para el diseño de software integrado o de sistemas en tiempo real.



Hay dos tipos de diagramas de ciclo de vida: los que contienen la escala de tiempo del estado/de la condición (*imagen anterior*) y los que muestran el ciclo de vida general (*imagen siguiente*).



9.1.8.1 Insertar elementos

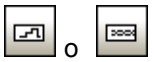
Usar iconos de la barra de herramientas:

1. Haga clic en un icono de la barra de herramientas Diagrama de ciclo de vida.
2. Haga clic en el área de trabajo del diagrama en el que desea insertar el elemento. Para insertar varios elementos del tipo seleccionado, mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo.

Arrastrar elementos desde la Estructura del modelo hasta el diagrama de ciclo de vida:

1. En la Estructura del modelo busque el elemento que quiere insertar en el otro diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).
2. Arrastre el elemento hasta el diagrama de ciclo de vida.

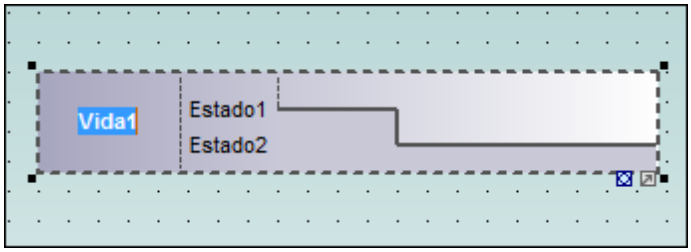
9.1.8.2 Línea de vida

 o **Línea de vida**

La línea de vida es un participante de la interacción y tiene dos representaciones distintas: un **Estado/Condición** o un **Valor general**. Pulse **Ctrl+Entrar** para crear una línea nueva en el nombre de la línea de vida.

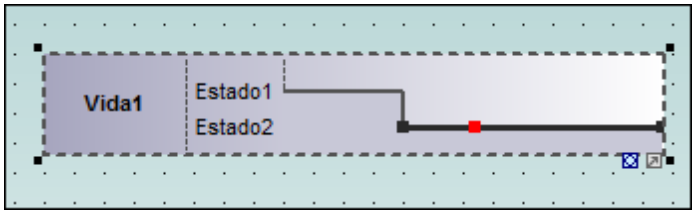
Para insertar un Estado/Condición (InvarianteDeEstado) y definir cambios de estado:

1. Haga clic en el icono **Línea de vida (Estado o Condición)**  de la barra de herramientas y después haga clic en el área de trabajo del diagrama.



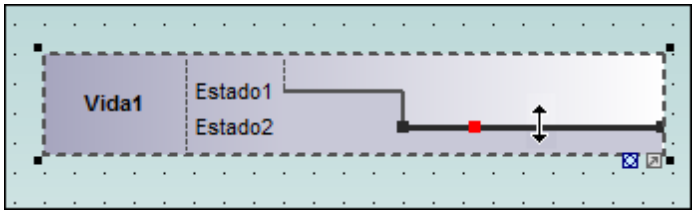
2. Escriba el nombre de la línea de vida o utilice el nombre predeterminado **`Líneadevida1`**.
3. Haga clic en una sección de la línea de tiempo para seleccionarla.
4. Haga clic en la posición de la línea de tiempo donde quiere que se produzca el cambio de estado.

   En este momento aparece una flecha con dos puntas.

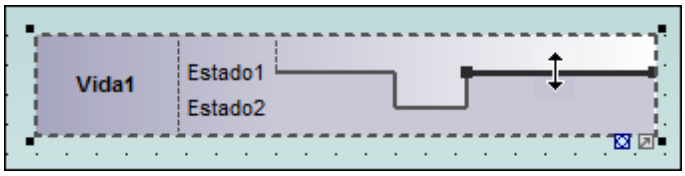


   Y además aparece un recuadro rojo en la posición donde hizo clic, que divide la línea por ese punto.

5. Ponga el cursor en la parte derecha de la línea y arrastre la línea hacia arriba.



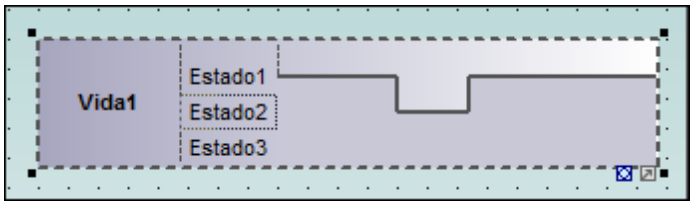
   No olvide que solo puede mover líneas entre estados de la línea de vida actual.



   Puede definir un número ilimitado de cambios de estado en una línea de vida. El recuadro rojo de la línea desaparece al hacer clic en otra parte del diagrama.

Para añadir un estado nuevo a la línea de vida:

1. Haga clic con el botón derecho en la línea de vida y seleccione **Nuevo/a | Estado o Condición (InvarianteDeEstado)**.
   El estado nuevo **`Estado3`** se añade a la línea de vida.



Para cambiar la posición de un estado (dentro de una línea de vida):

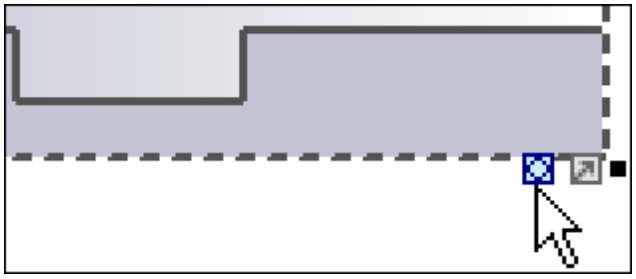
1. Haga clic en la etiqueta del estado.
2. Arrástrela hasta la posición nueva.

Para eliminar un estado de una línea de vida:

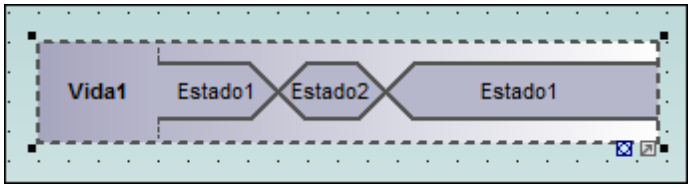
1. Haga clic en la etiqueta del estado y pulse la tecla **Supr**. También puede hacer clic con el botón derecho en el estado y seleccionar el comando **Eliminar**.

Para cambiar de tipo de diagrama de ciclo de vida:

1. Haga clic en el icono **Alternar estilo de notación** que aparece en la esquina inferior derecha de la línea de vida.



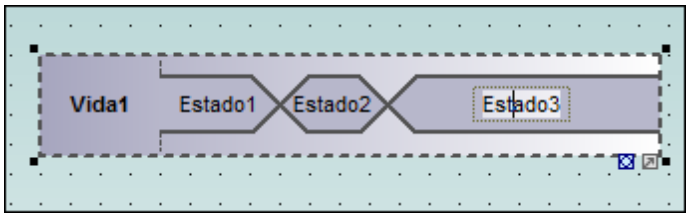
Ahora la línea de vida se presenta con su `Valor general`. Cada punto donde se cruzan las líneas es un cambio de estado/valor.



Recuerde que al hacer clic en el icono **Línea de vida (Valor general)**  se inserta una línea de vida como la de la imagen anterior. Puede cambiar a la otra presentación cuando quiera.

Para añadir un estado nuevo a la línea de vida Valor general:

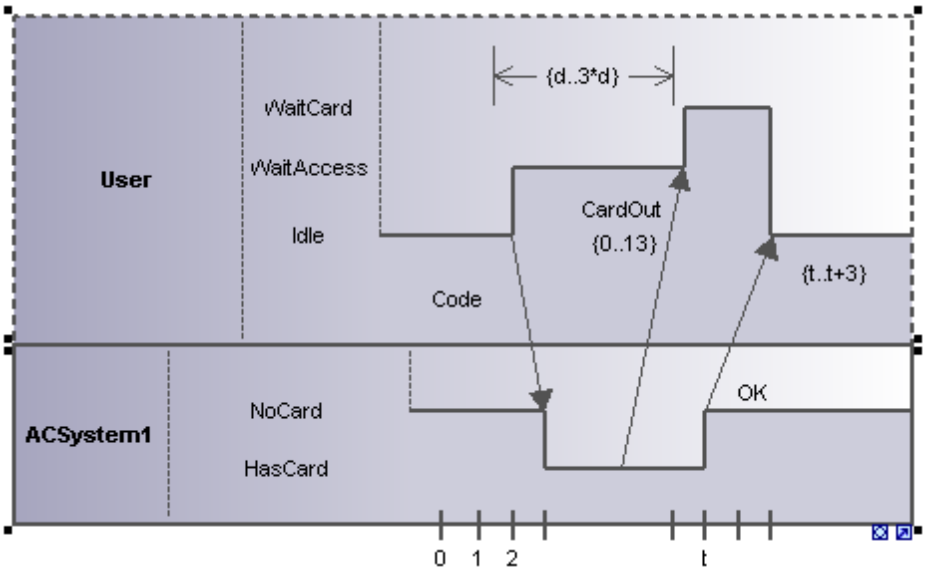
1. Haga clic con el botón derecho en la línea de vida y seleccione **Nuevo/a | Estado o condición (InvarianteDeEstado)**.
2. Edite el nombre del estado nuevo y pulse **Entrar** para confirmar.



Se añade un estado nuevo en la línea de vida.

Agrupar las líneas de vida

Si apila las líneas de vida, UModel las reorganiza automáticamente de forma correcta y conserva las marcas de graduación disponibles hasta ese momento. También puede crear mensajes entre las líneas de vida. Para ello arrastre el objeto de mensaje correspondiente hasta la posición deseada.



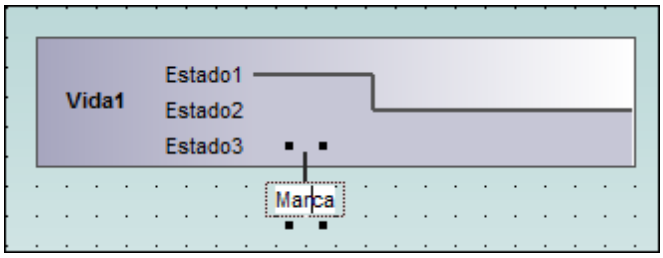
9.1.8.3 *Marca de graduación*

MarcaDeGraduación

Este icono permite insertar las marcas de graduación de una escala de tiempo en la línea de vida.

Para insertar una marca de graduación:

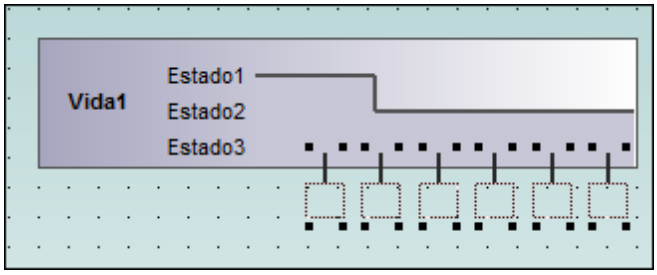
1. Haga clic en el icono de la marca de graduación y después en la línea de vida para insertarla.



2. Para insertar varias marcas, pulse la tecla **Ctrl** mientras hace clic en la línea de vida tantas veces como sea necesario.
3. Escriba el nombre de la marca.
   Si quiere cambiar la posición de una marca de graduación, simplemente arrástrela hasta la posición nueva.

Para espaciar las marcas de graduación uniformemente:

1. Seleccione todas las marcas de graduación.
2. Haga clic en el icono **Espaciar uniformemente en horizontal**  de la barra de herramientas.



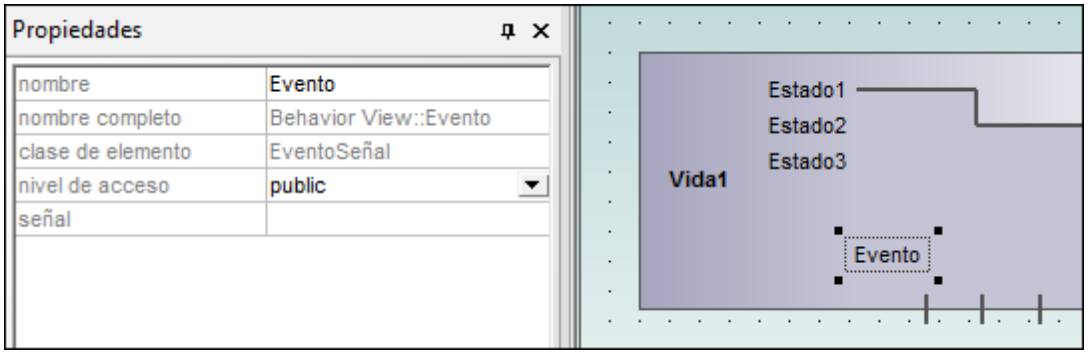
9.1.8.4 Evento/estímulo

Evento/ Estímulo

Este icono sirve para ilustrar el cambio de estado de un objeto causado por el correspondiente evento o estímulo. Los eventos recibidos se anotan para mostrar qué evento provoca el cambio en la condición o en el estado.

Para insertar un evento o estímulo:

1. Haga clic en el icono **Evento o estímulo**  y después haga clic en la posición de la escala de tiempo donde tiene lugar el cambio de estado.



2. Escriba el nombre del evento. Observe que las propiedades del evento aparece en el panel Propiedades.

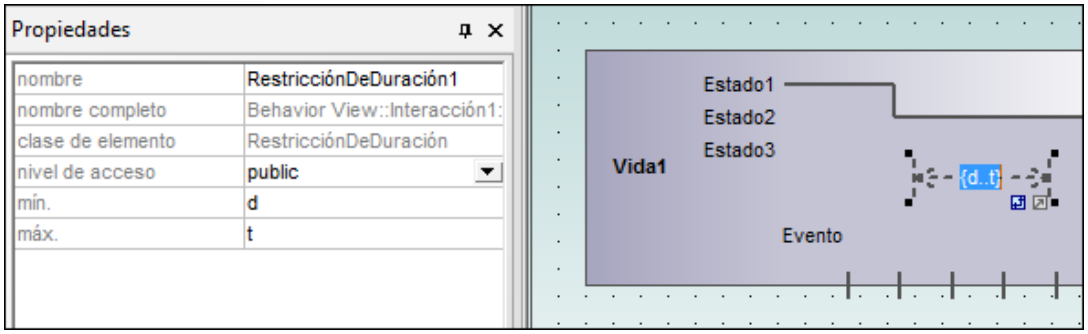
9.1.8.5 Restricción de duración

RestricciónDeDuración

Una duración define una `EspecificaciónDeValor` que denota el período de tiempo comprendido entre el punto de inicio y el punto final. Por lo general una duración es una expresión que representa el tiempo que puede pasar durante este período.

Para insertar una RestricciónDeDuración:

1. Haga clic en el icono **RestricciónDeDuración**  y después haga clic en la posición de la línea de vida donde debe aparecer la restricción.

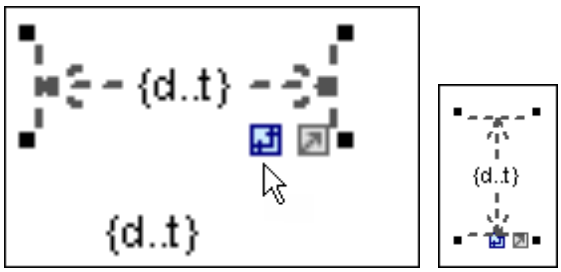


El valor mínimo y máximo predeterminado ("`d..t`") aparece automáticamente. Para editar estos valores haga doble clic en la restricción o edite los valores en el panel Propiedades.

2. Si quiere, use los controladores para cambiar el objeto de tamaño.

Para cambiar la orientación de la RestricciónDeDuración:

1. Haga clic en el icono rotación para poner la restricción en vertical.



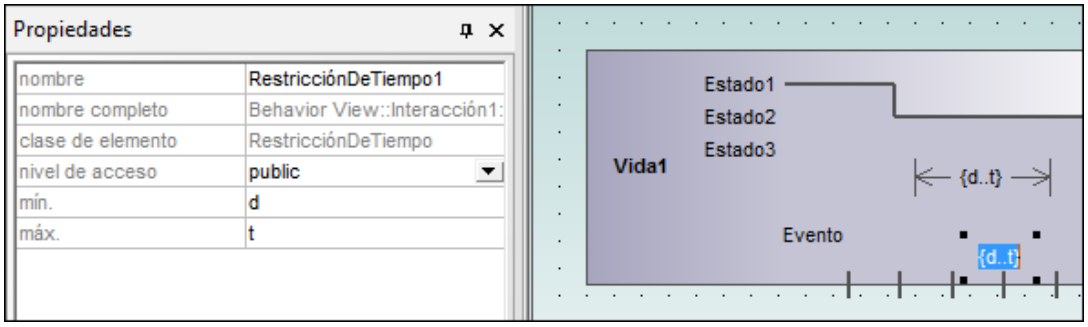
9.1.8.6 Restricción de tiempo

RestricciónDeTiempo

Una restricción de tiempo suele representarse como una asociación gráfica entre un `IntervaloDeTiempo` y la construcción que limita. Lo normal es que sea una asociación gráfica entre un evento y un intervalo de tiempo.

Para insertar una restricción de tiempo:

1. Haga clic en el icono **RestricciónDeTiempo**  en la barra de herramientas y después haga clic en la posición de la línea de vida donde debe aparecer la restricción.



El valor mínimo y máximo predeterminado ("`d..t`") aparece automáticamente. Para editar estos valores haga doble clic en la restricción o edite los valores en el panel Propiedades.

9.1.8.7 Mensaje

Mensaje (Llamada) **Mensaje (Respuesta)** **Mensaje asincrónico (Llamada)**

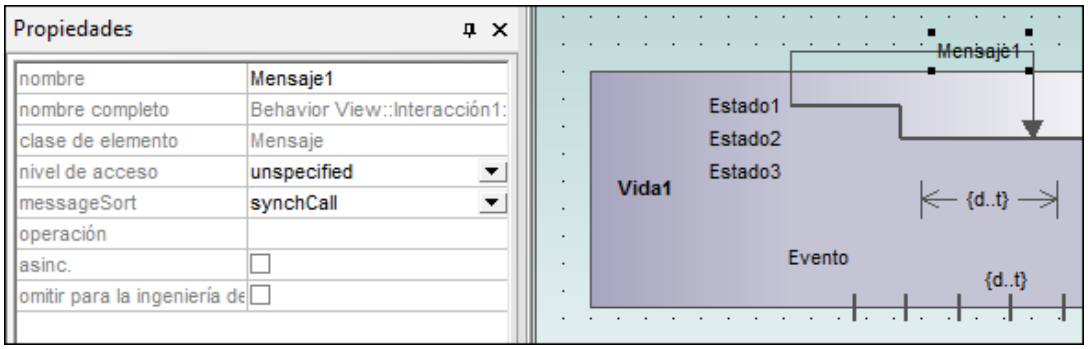
Un mensaje es un elemento de modelado que define un tipo concreto de comunicación dentro de una interacción. Una comunicación puede lanzar una señal, invocar una operación, crear o destruir una instancia, etc. El mensaje especifica el tipo de comunicación definida por la `EspecificaciónDeEjecución` emisora, así como el remitente y el destinatario.

Los mensajes se transmiten entre la escala de tiempo remitente y la escala de tiempo destinataria y se representan en forma de flechas con etiqueta.

Para insertar un mensaje:

1. Haga clic en el icono del mensaje que desea insertar en la barra de herramientas.
2. Haga clic en el objeto remitente.
3. Arrastre la línea del mensaje y suéltela encima del objeto destinatario.

   La línea de vida se resalta cuando el mensaje se puede soltar.



- La dirección en la que se arrastra la flecha define la dirección del mensaje. Los mensajes de respuesta pueden apuntar en ambas direcciones.
- Mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo para insertar varios mensajes.

Para eliminar un mensaje:

1. Haga clic en el mensaje que desea eliminar.
2. Pulse la tecla **Supr** para eliminarlo del modelo. También puede hacer clic con el botón derecho en el mensaje y elegir el comando **Eliminar solo en el diagrama**.

9.2 Diagramas de estructura

Los diagramas de estructura muestran qué elementos estructurales componen un sistema o una función. Pueden representar tanto las relaciones estáticas (p. ej. diagramas de clases) como las dinámicas (p. ej. diagramas de objetos).

Diagramas de estructura

- Diagramas de clases
- Diagramas de componentes
- Diagramas de estructura de un compuesto
- Diagramas de implementación
- Diagramas de objetos
- Diagramas de paquetes
- Diagramas de perfil

9.2.1 Diagrama de clases

Esta sección se ocupa de describir las tareas y los conceptos relacionados con los diagramas de clases:

- Personalizar diagramas de clases
- Invalidar operaciones de clases base e implementar operaciones de interfaz
- Crear métodos getter y setter
- Notaciones de forma esférica (Ball and socket)
- Agregar excepciones emitidas a los métodos de una clase
- Generar diagramas de clases

Para obtener información general sobre los diagramas de clases, consulte el apartado *Diagramas de clases* del tutorial que acompaña a esta documentación.

9.2.1.1 Personalizar diagramas de clases

Expandir/ocultar los compartimentos de las clases en un diagrama UML:
Hay varias maneras de expandir los compartimentos de los diagramas de clases.

- Haga clic en los botones **+** o **-** de la clase activa para expandir/contraer el compartimento correspondiente.
- Use el recuadro de selección (arrastrando el puntero por el diagrama) para marcar varias clases y después haga clic en el botón expandir/ocultar. También puede usar **Ctrl+clic** para seleccionar varias clases.
- Pulse **Ctrl+A** para seleccionar **todas las clases** y después haga clic en el botón expandir/ocultar en una de las clases para expandir/contraer los compartimentos correspondientes.

Expandir/ocultar los compartimentos de las clases en la Estructura del

modelo:

En la Estructura del modelo, las clases son subelementos de los paquetes y la acción expandir/ocultar se puede ejecutar en los paquetes o en las clases.

Haga clic en el paquete / en la clase que desea **expandir** y:
- Pulse la tecla ***** para expandir el paquete/la clase actual y todos los subelementos.
- Pulse la tecla **+** para abrir el paquete/la clase actual.

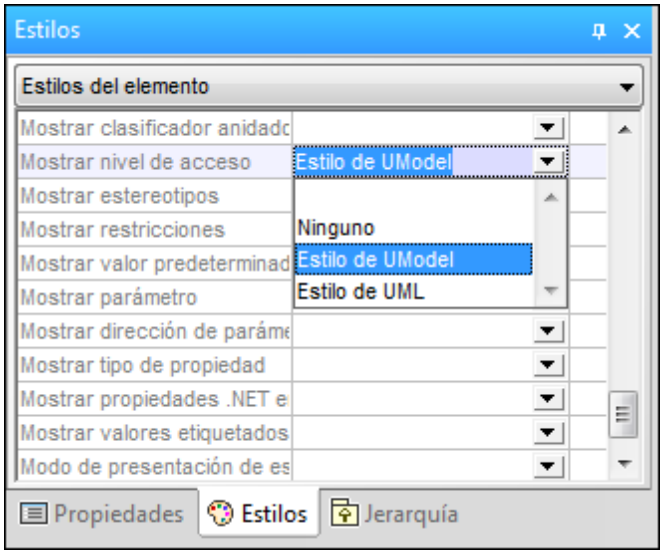
Para **contraer** los paquetes/las clases, pulse la tecla del teclado **-**.
Recuerde que puede usar las teclas del teclado estándar o del teclado numérico para hacer esto.

Cambiar el icono de nivel de acceso

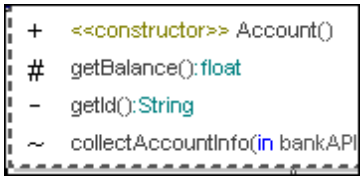
Haga clic en el icono de nivel de acceso situado a la izquierda de una operación o propiedad para abrir una lista desplegable en la que puede elegir el nivel de acceso del elemento.

En UModel también puede elegir qué tipo de símbolo se utiliza para identificar los niveles de acceso nivel de acceso:

- Haga clic en una clase del diagrama y abra la pestaña *Estilos*. Desplácese hasta la el estilo `Mostrar nivel de acceso`.



Aquí puede elegir entre usar el estilo de UModel, el estilo de UML (*imagen siguiente*) o no utilizar ninguno.



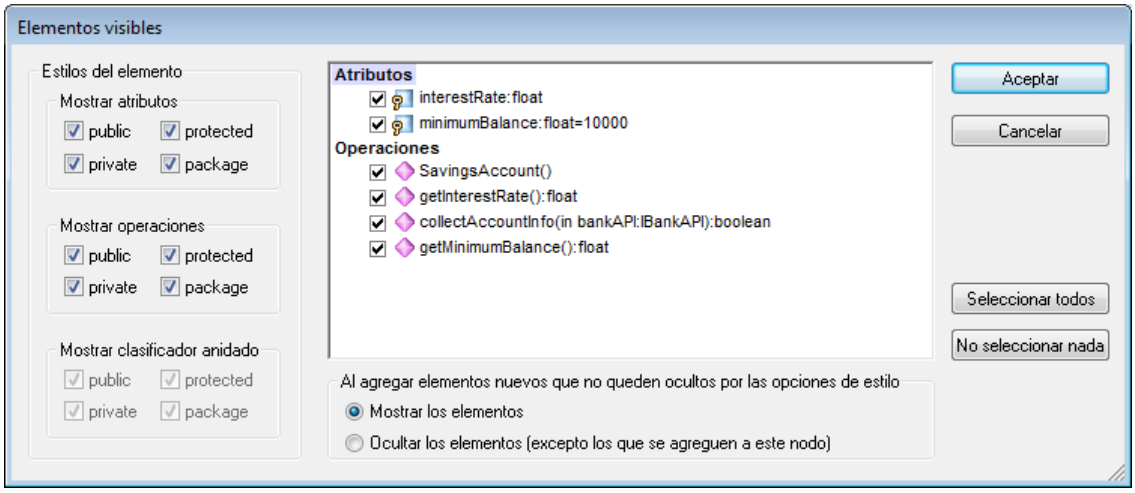
Mostrar/ocultar el contenido de los nodos (atributos de clase, operaciones, slots)

En UModel también puede elegir qué atributos y operaciones de una clase aparecen en el diagrama y elegir qué nuevos atributos y operaciones se muestran cuando se añaden.

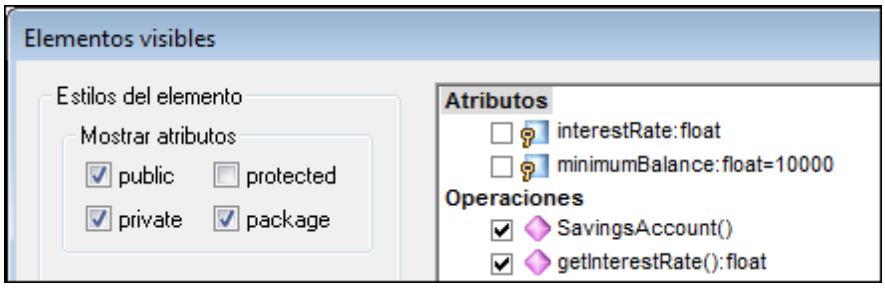
No olvide que los slots de objetos (es decir, EspecificacionesDeInstancia) se pueden mostrar/ocultar de la misma manera.

Haga clic con el botón derecho en una clase (p. ej. `SavingsAccount`) y elija el comando **Mostrar u ocultar el contenido del nodo** del menú contextual.

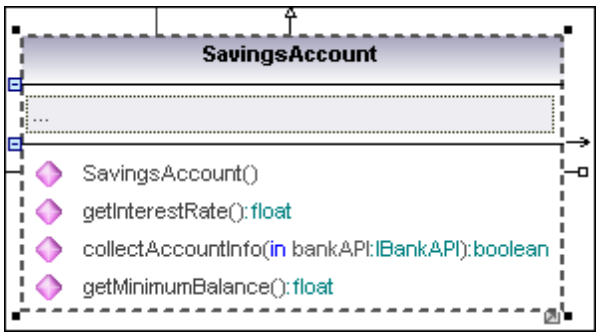
Esto abre el cuadro de diálogo "Elementos visibles".



Si desactiva la casilla *protected* del grupo *Mostrar atributos*, los atributos que tengan el nivel `protected` se desactivan en la vista previa del cuadro de diálogo.



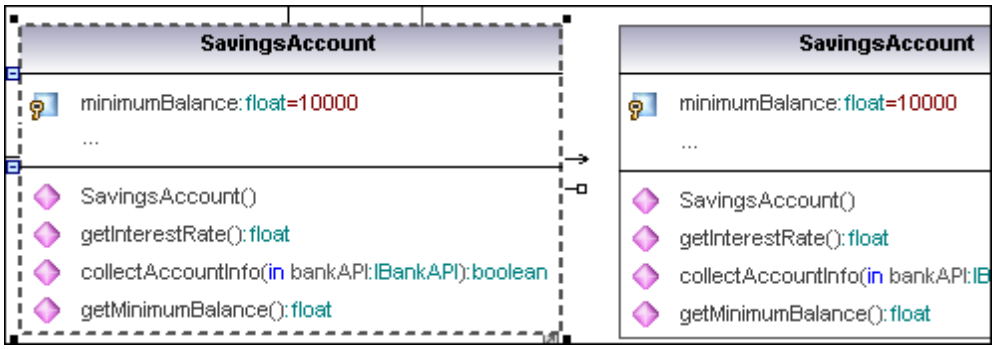
Haga clic en **Aceptar** para cerrar el cuadro de diálogo. Los atributos de la clase que tienen el nivel `protected` se reemplazan con tres puntos (...). Haga doble clic en los tres puntos para abrir otra vez el cuadro de diálogo "Elementos visibles".



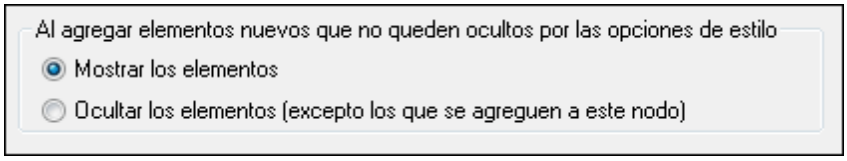
No olvide que en la vista previa del cuadro de diálogo puede desactivar los atributos uno a uno.

Mostrar/ocultar atributos y operaciones (Estilos del elemento)

En UModel puede insertar varias instancias de la misma clase en un el mismo diagrama o incluso en diagramas distintos. Cada una de las vistas de esta clase puede tener niveles de acceso diferentes. Por ejemplo, la imagen siguiente muestra dos vistas de la misma clase: `SavingsAccount`.

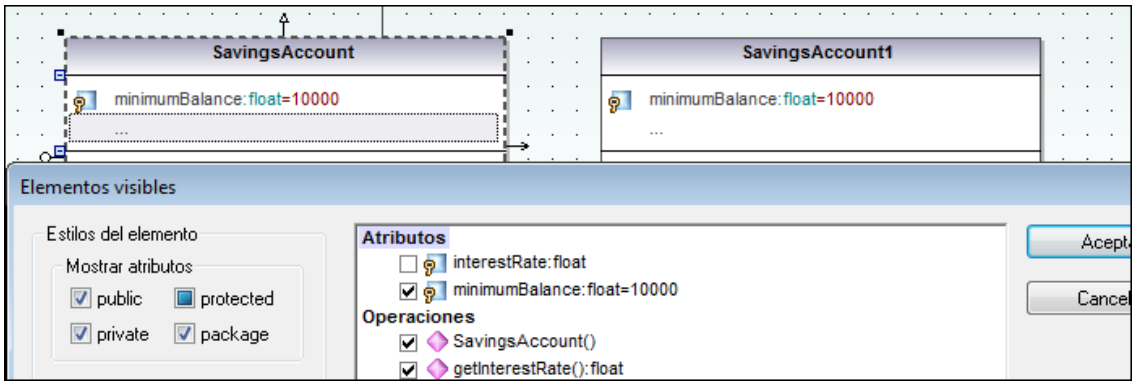


En el cuadro de diálogo "Elementos visibles" las opciones del grupo *Al agregar elementos nuevos que no queden ocultos por las opciones de estilo* sirven para definir qué será visible cuando se añadan elementos nuevos a la clase. Los elementos se pueden añadir a mano en el diagrama o en la Estructura del modelo o automáticamente durante el proceso de ingeniería de código.

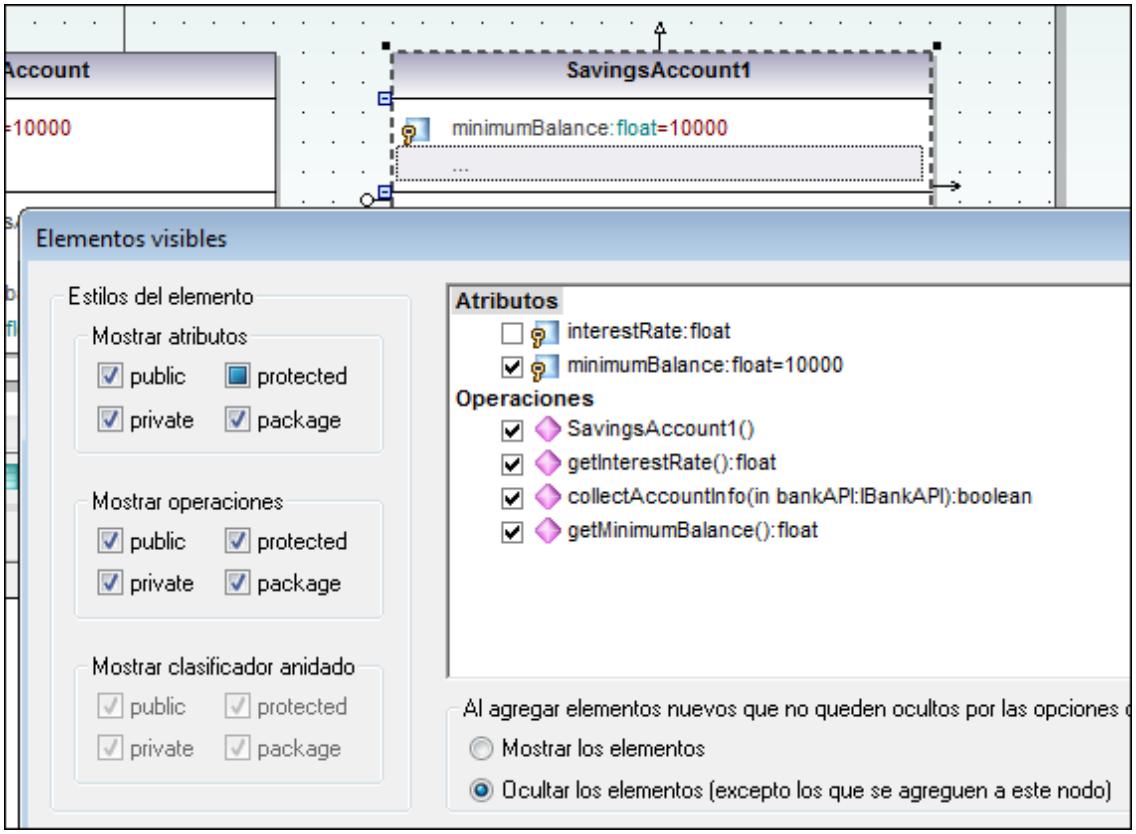


Mostrar los elementos: elija esta opción si quiere que se muestren todos los elementos que se añadan a cualquier vista de la clase.

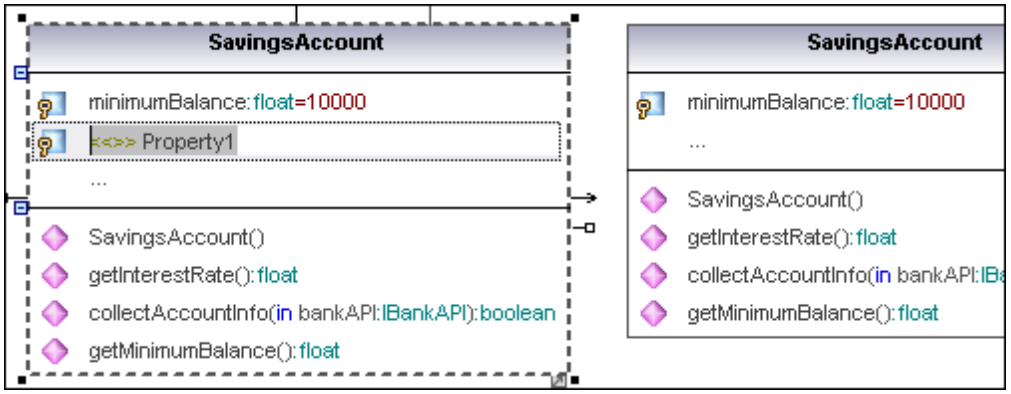
P. ej. el atributo `interestRate:float` se ocultó en ambas vistas de **`SavingsAccount`** y se dejó visible el atributo `minimumBalance`. El botón de opción *Mostrar los elementos* está activo para la vista izquierda de la clase. Al hacer doble clic en los tres puntos (...) del compartimento del atributo de la vista **izquierda** de la clase, se abre el cuadro de diálogo "Elementos visibles", donde podemos ver que el botón de opción *Mostrar los elementos* está activo.



Al hacer doble clic en los tres puntos (...) del compartimento del atributo de la vista **derecha** de la clase, se abre el cuadro de diálogo "Elementos visibles", donde podemos ver que está activo el botón de opción *Ocultar los elementos (excepto los que se agreguen a este nodo)*.

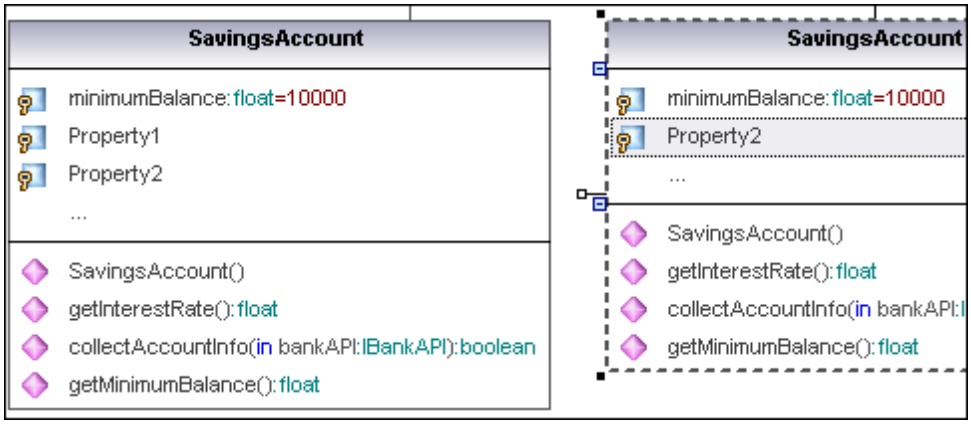


Si hacemos clic en la vista **izquierda** de la clase y pulsamos **F7** (o si hacemos clic en la clase en la Estructura del modelo y pulsamos **F7**), en la clase se añade un atributo nuevo (`Propiedad1`).



El nuevo elemento solamente está visible en la vista **izquierda** de la clase (porque en esta vista está activa la opción *Mostrar los elementos*). En la vista **derecha** de la clase el elemento está oculto (porque en esa vista está activa la opción *Ocultar los elementos*).

Si hacemos clic en la vista derecha de la clase y pulsamos F7, en la clase se añade un atributo nuevo (`Property2`). Este atributo nuevo está visible en la vista derecha porque la opción oculta los elementos *excepto los que se agreguen a este nodo*. Aquí *nodo* significa esta clase o elemento de modelado.

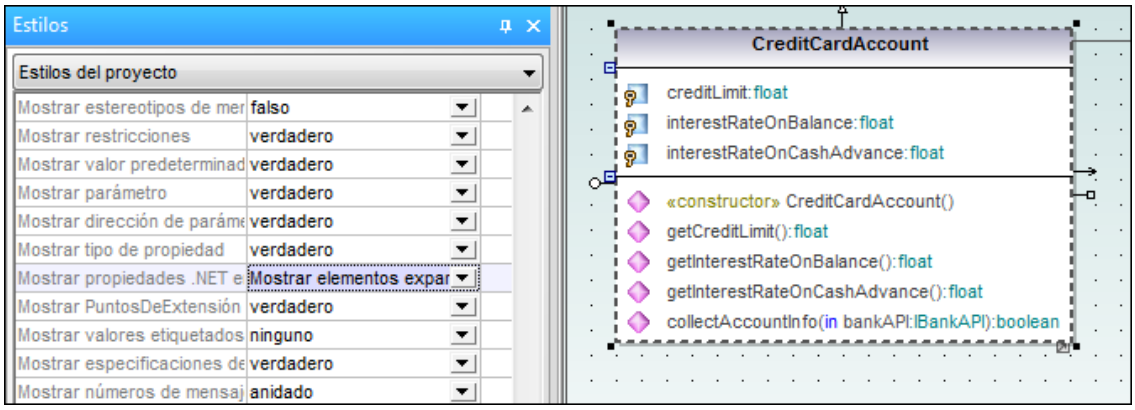


Ahora el atributo `Property2` está visible en la vista izquierda de la clase (porque en esta vista está activa la opción *Mostrar los elementos*).

Nota: los valores etiquetados de los elementos ocultos también se ocultan si se elige la opción *Ocultar los elementos*.

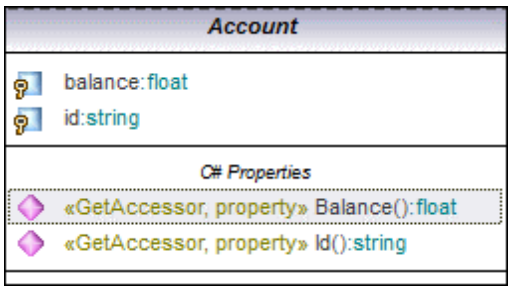
Mostrar/ocultar compartimentos VS .NET:

UModel puede mostrar las propiedades .NET en un compartimento separado. Para ello, habilite el estilo `Mostrar propiedades .NET en un compartimento propio` del panel Estilos.



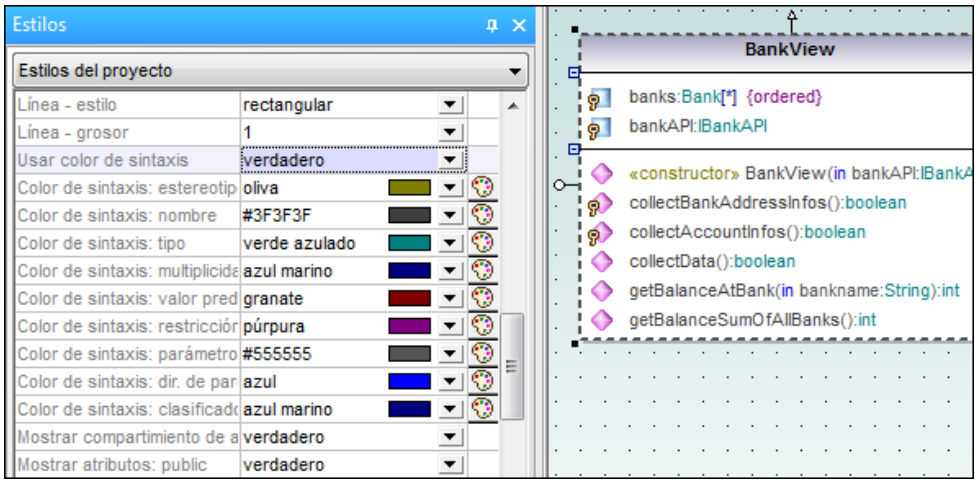
Mostrar las propiedades VS .NET como asociaciones

UModel también puede mostrar las propiedades .NET como asociaciones.
Haga clic con el botón derecho en una propiedad C# y elija **Mostrar | Todas las propiedades .NET como asociaciones** en el menú contextual.



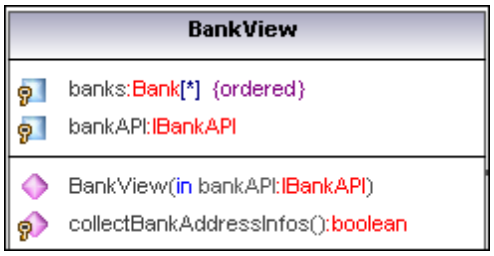
Cambiar el color de sintaxis de las operaciones y propiedades

UModel habilita automáticamente la función de color de sintaxis, pero esta función se puede personalizar. A continuación puede ver la configuración predeterminada.



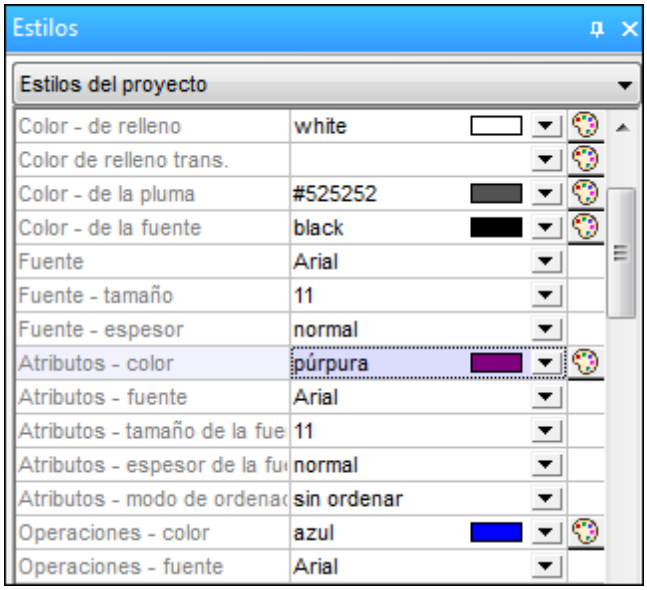
Para cambiar las opciones predeterminadas de color de sintaxis:

1. Abra la pestaña *Estilos* y desplácese hasta los estilos que empiezan por `Color de sintaxis`.
2. Cambie el valor de uno de estos estilos. Por ejemplo: cambie el valor de `Color de sintaxis: tipo` a `red`.



Para deshabilitar el color de sintaxis:

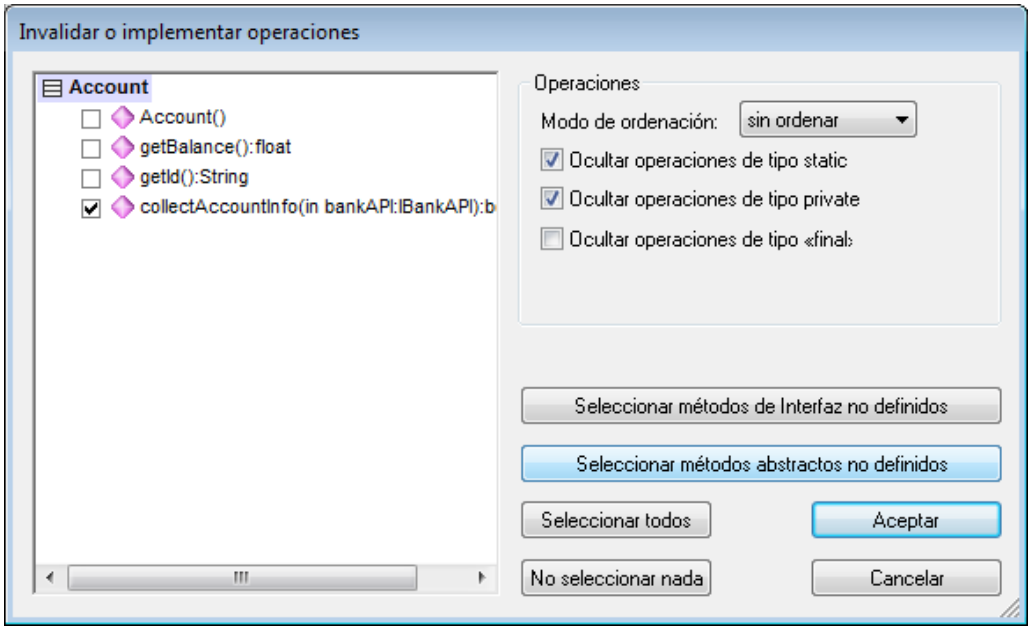
1. Abra la pestaña Estilos y asigne el valor `false` al estilo `Usar color de sintaxis`.
2. Y ahora use los estilos `Atributos - color` o los estilos `Operaciones - color` para personalizar estos elementos en las clases.



9.2.1.2 Invalidar operaciones de clases base e implementar operaciones de interfaz

Invalidar operaciones de la clase base e implementar operaciones de interfaz
UModel ofrece la posibilidad de invalidar las operaciones de la clase base o implementar operaciones de interfaz de una clase. Esto se puede hacer desde la Estructura del modelo, desde Favoritos o desde los diagramas de clases.

1. Haga clic con el botón derecho en una de las clases derivadas del diagrama (p. ej. `CheckingAccount`) y elija **Invalidar o implementar operaciones** en el menú contextual. Esto abre el cuadro de diálogo "Invalidar o implementar operaciones" (*imagen siguiente*).



2. Seleccione las operaciones que desea invalidar y haga clic en **Aceptar** para confirmar. Con los botones **Seleccionar métodos...** puede seleccionar esos tipos de métodos en la vista previa de la izquierda.

   Nota: cuando se abre este cuadro de diálogo, se marcan (se activan) las operaciones de las clases base y de las interfaces implementadas que tiene la misma firma que las operaciones disponibles.

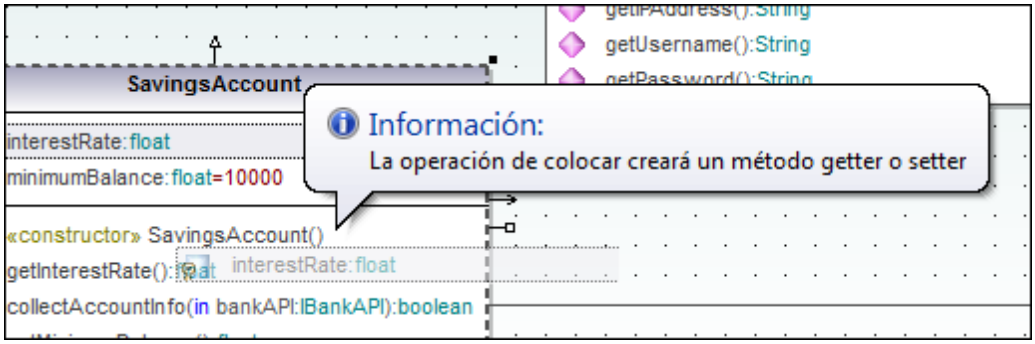
9.2.1.3 Crear métodos getter y setter

Durante el proceso de modelado a veces es necesario crear métodos getter/setter para los atributos existentes. UModel ofrece dos métodos para crear métodos getter/setter:

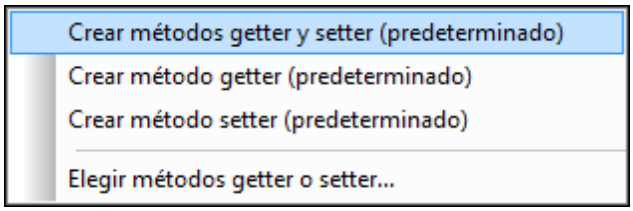
- Arrastrando y colocando un atributo en el compartimento de una operación.
- Usando el menú contextual para abrir un cuadro de diálogo donde puede gestionar los métodos getter/setter.

Para crear métodos getter/setter mediante operaciones arrastrar y colocar:

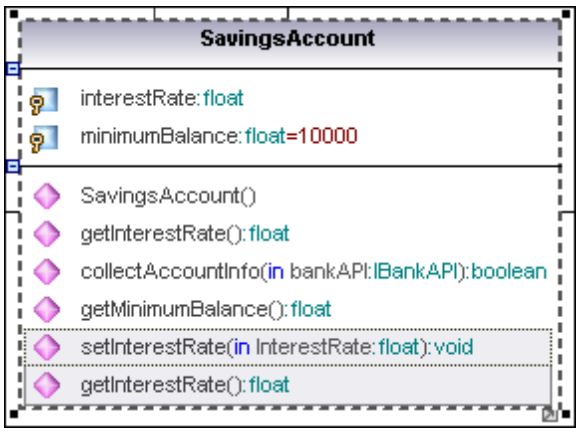
1. Arrastre un atributo desde el compartimento de atributos hasta el compartimento de operaciones.



Aparece un menú contextual donde puede elegir el tipo de método getter/setter que se debe crear.

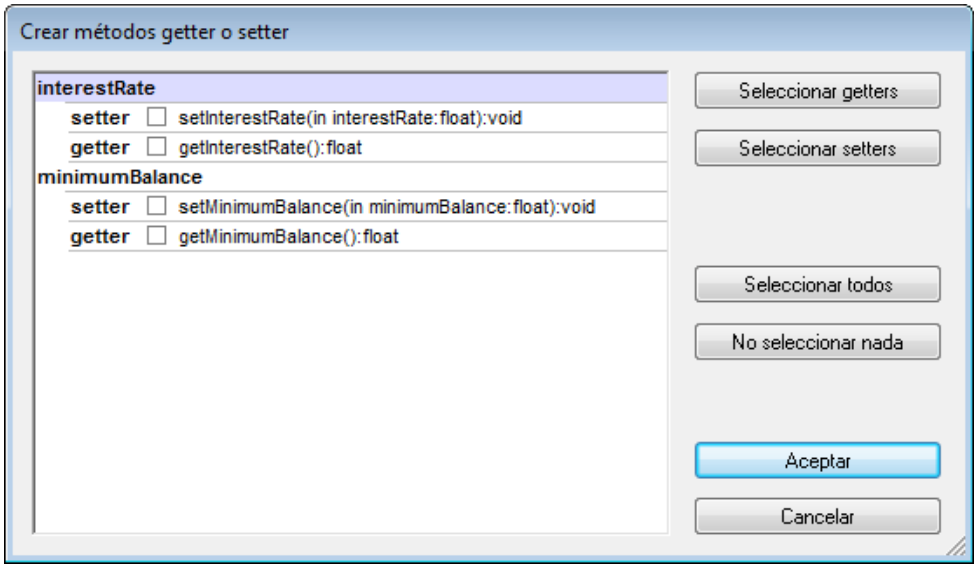


2. Seleccione el primer comando del menú para crear un método getter y setter para interestRate:float.



Para crear métodos getter/setter con el menú contextual:

1. Haga clic con el botón derecho en el título de una clase (p. ej. **`SavingsAccount`**) y elija la opción **Crear operaciones de método getter o setter...** del menú contextual. Esto abre el cuadro de diálogo "Crear métodos getter o setter", que enumera los atributos disponibles en la clase activa.

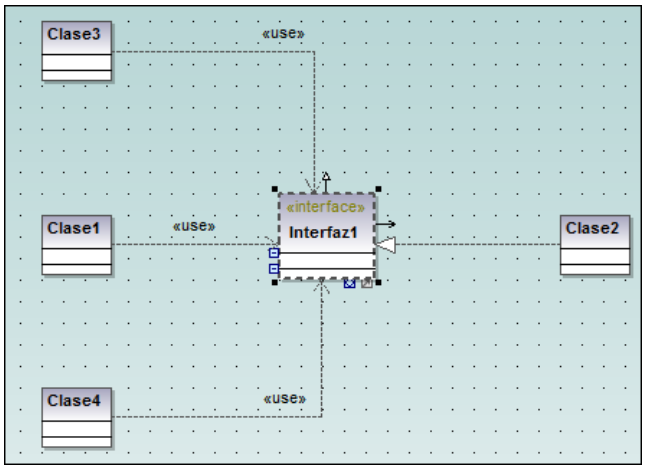


2. Use los botones o marque las casillas para seleccionar los métodos que desea crear.

   Nota: también puede hacer clic con el botón derecho en un solo atributo y usar el mismo método para crear una operación para el atributo.

9.2.1.4 Notaciones de forma esférica (Ball and socket)

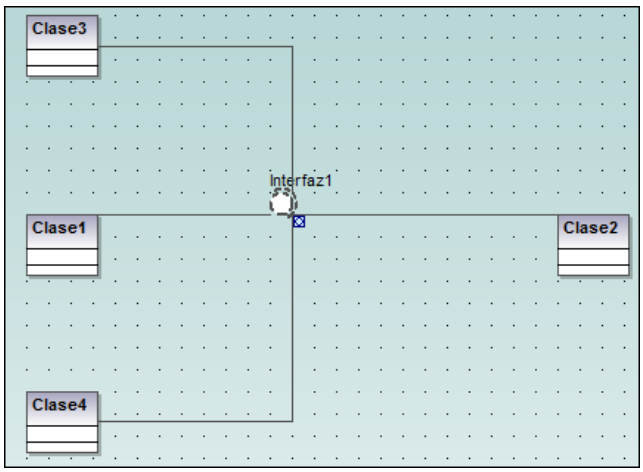
UModel es compatible con la notación en forma esférica de UML. Las clases que necesitan una interfaz muestran un círculo y el nombre de la interfaz, mientras que las clases que implementan la interfaz muestran una esfera.



En la imagen anterior, la `Clase2` realiza la `Interfaz1`, que es utilizada por las clases `clase1`, `clase3` y `clase4`. Para crear la relación de utilización entre las clases e interfaz se utilizó el icono **Utilización** de la barra de herramientas Diagrama de clases.

Para cambiar entre la vista estándar y la vista esférica:

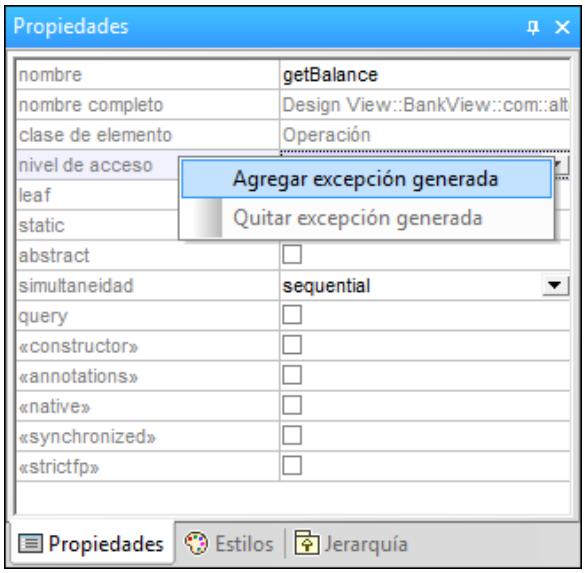
- Haga clic en el icono **Alternar estilo de notación de interfaz** situado en la parte inferior del elemento interfaz.



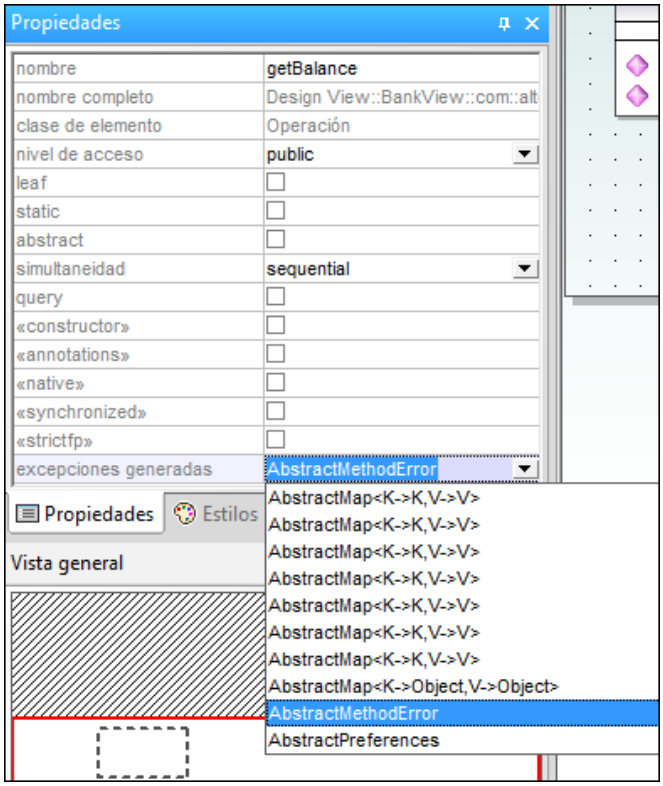
9.2.1.5 *Agregar excepciones emitidas a los métodos de una clase*

Para agregar excepciones generadas a los métodos de una clase

1. En la Estructura del modelo haga clic en el método de la clase en la que desea agregar la excepción generada (p. ej. el método **`getBalance`** de la clase **`Account`**).
2. Ahora haga clic con el botón derecho dentro del panel Propiedades y elija la opción **Agregar excepción generada** del menú contextual.



   Esto añade el campo `Excepciones generadas` al panel Propiedades y selecciona la primera opción de la lista desplegable.
3. Seleccione una opción de la lista despegable del campo `Excepciones generadas` o escriba el nombre que quiera.

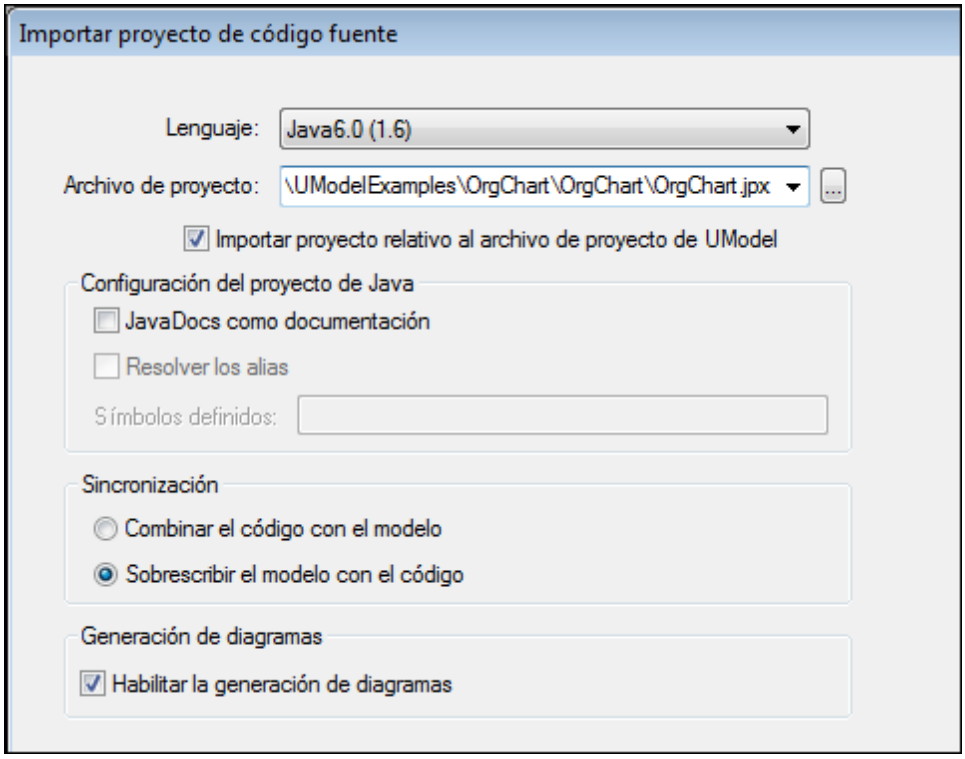


9.2.1.6 Generar diagramas de clases

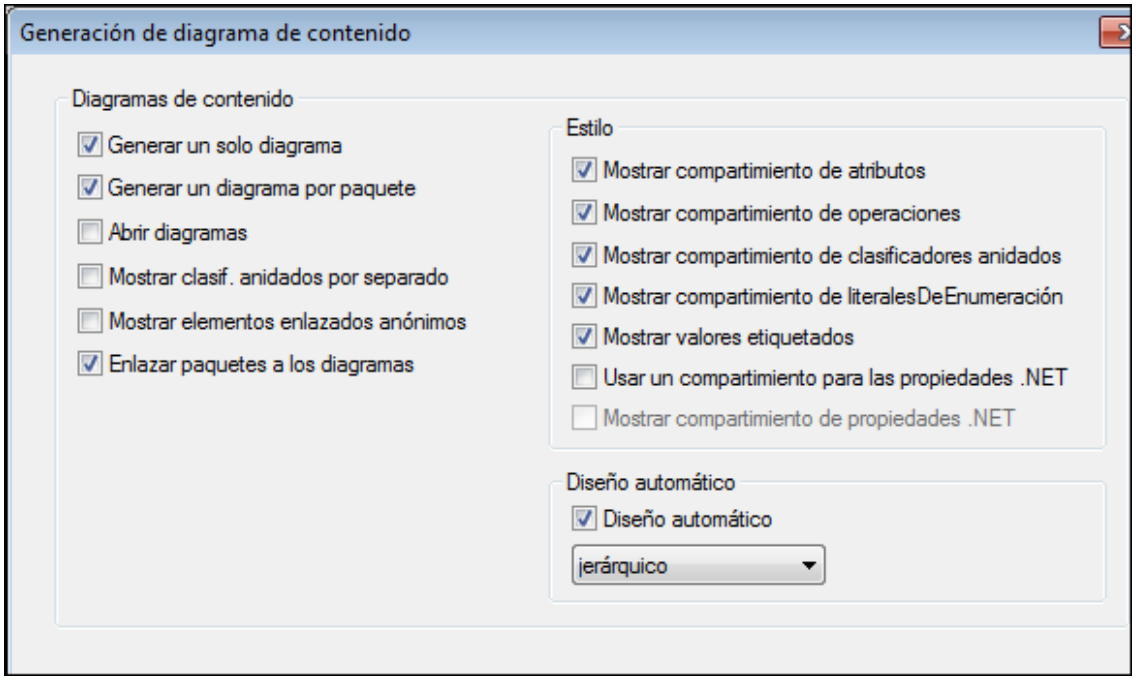
Si lo prefiere, en lugar de diseñar diagramas de clases en UModel directamente, también puede generarlos automáticamente cuando importe código fuente o binarios en sus proyectos de UModel (consulte los apartados Importar código fuente en proyectos e Importar binarios Java, C# y VB).

Siga las instrucciones del asistente para la importación y asegúrese de que:

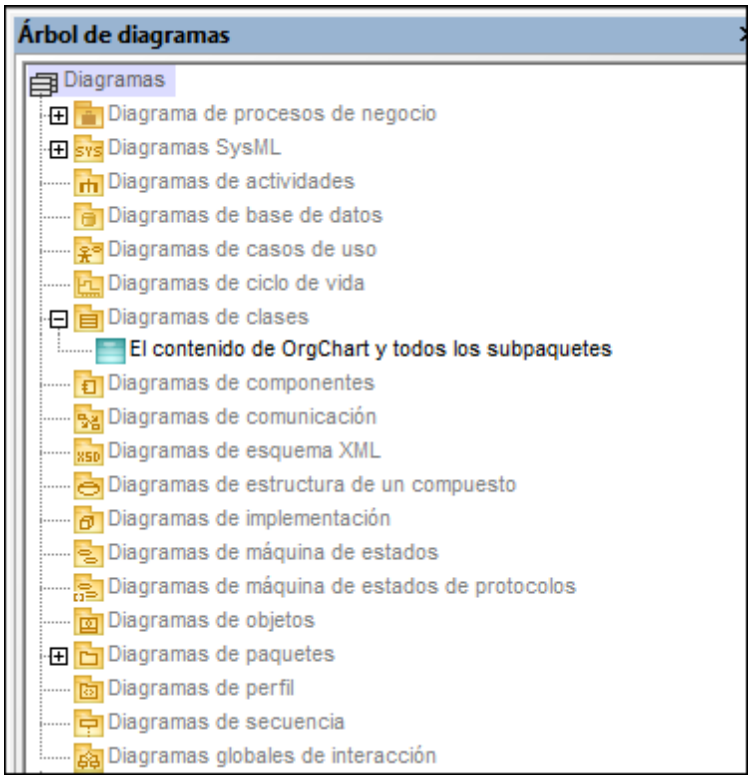
1) La casilla *Habilitar generación de diagramas* está marcada en el cuadro de diálogo "Importar proyecto de código fuente", "Importar tipos binarios" o "Importar directorio de código fuente".



2) Las opciones *Generar un solo diagrama* y *Generar un diagrama por paquete* están marcadas en el cuadro de diálogo "Generación de diagrama de contenido".



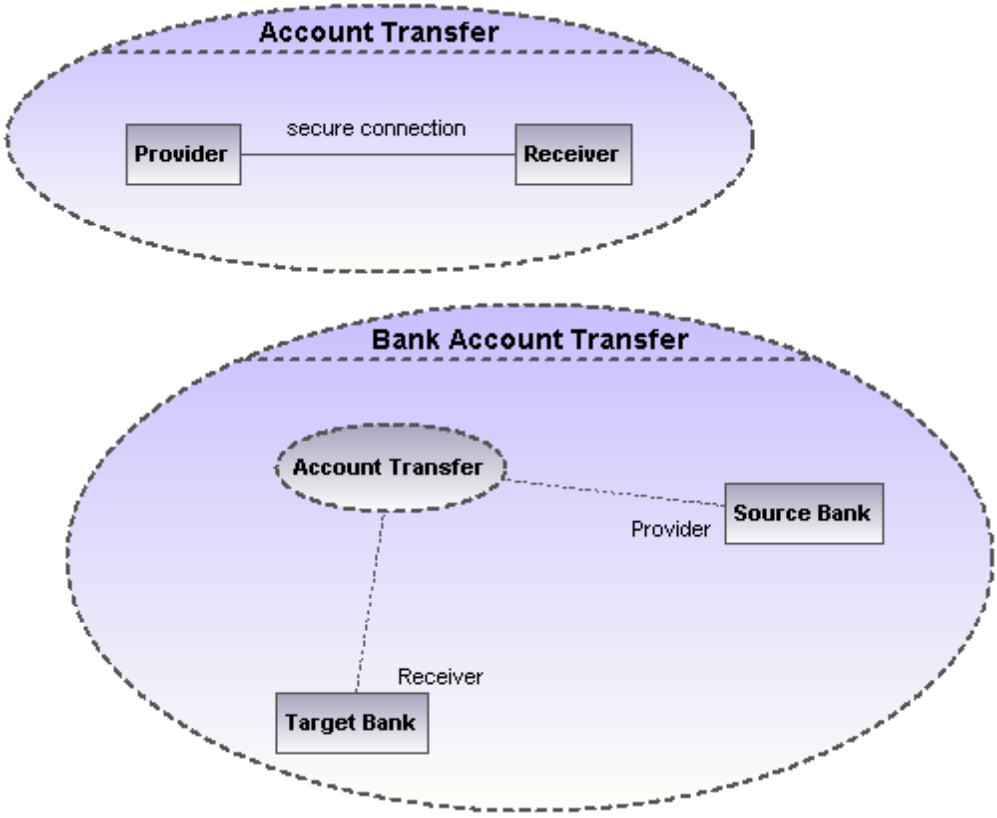
3) Una vez finalizada la operación de importación, los diagramas de clases generados estarán disponibles bajo el nodo `Diagramas de clases` del Árbol de diagramas.



9.2.2 Diagrama de estructura de un compuesto

Sitio web de Altova: diagramas de estructura de un compuesto UML

Los diagramas de estructura de un compuesto son un tipo de diagrama añadido en la especificación UML 2.0 y sirven para mostrar la estructura interna (partes, puertos, conectores...) de un clasificador estructurado o colaboración.



9.2.2.1 Insertar elementos

Usar iconos de la barra de herramientas:

1. Haga clic en un icono de la barra de herramientas Diagrama de estructura de un compuesto.
2. Haga clic en el área de trabajo del diagrama en el que desea insertar el elemento. Para insertar varios elementos del tipo seleccionado, mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo.

Arrastrar elementos desde la Estructura del modelo hasta el diagrama de estructura de un compuesto:

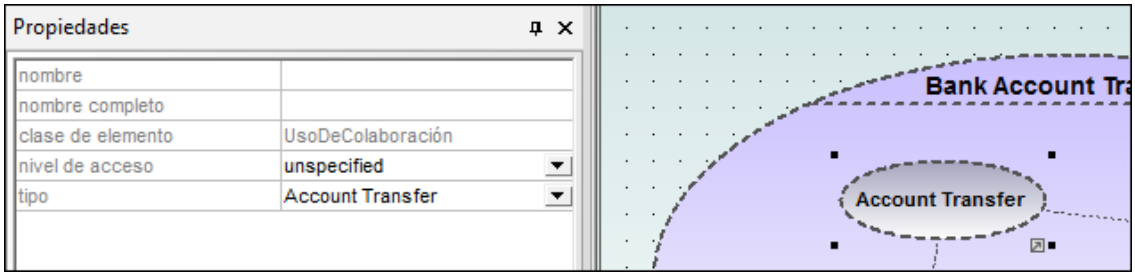
1. En la Estructura del modelo busque el elemento que quiere insertar en el otro diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).
2. Arrastre el elemento hasta el diagrama de estructura de un compuesto.

Colaboración

Este icono inserta un elemento `colaboración`, que es un tipo de clasificador/instancia que se comunica con otras instancias para producir el comportamiento del sistema.

UsoDeColaboración

Este icono inserta un elemento `usoDeColaboración`, que representa un uso determinado de una colaboración en la que participan determinadas clases o instancias que desempeñan el papel de la colaboración. Un uso de colaboración se representa por medio de una elipse con una línea de puntos. Dentro de la elipse aparece el nombre de la instancia, un punto y coma y el nombre del tipo de colaboración.



Cuando cree dependencias entre elementos uso de colaboración, es necesario rellenar el campo tipo para poder crear el enlace de roles y la colaboración de destino debe tener una parte/un rol como mínimo.

Parte (Propiedad)

Inserta un elemento `parte`, que representa un conjunto de instancias propiedad de un clasificador contenedor. Los elementos parte pueden añadirse a colaboraciones y a clases.

Puerto

Inserta un elemento `puerto`, que define el punto de interacción entre un clasificador y su entorno. Los elementos puerto pueden añadirse en partes con un tipo definido.

Clase

Inserta un elemento `clase`, que es el clasificador que tiene lugar en ese uso concreto de la colaboración.

Conector

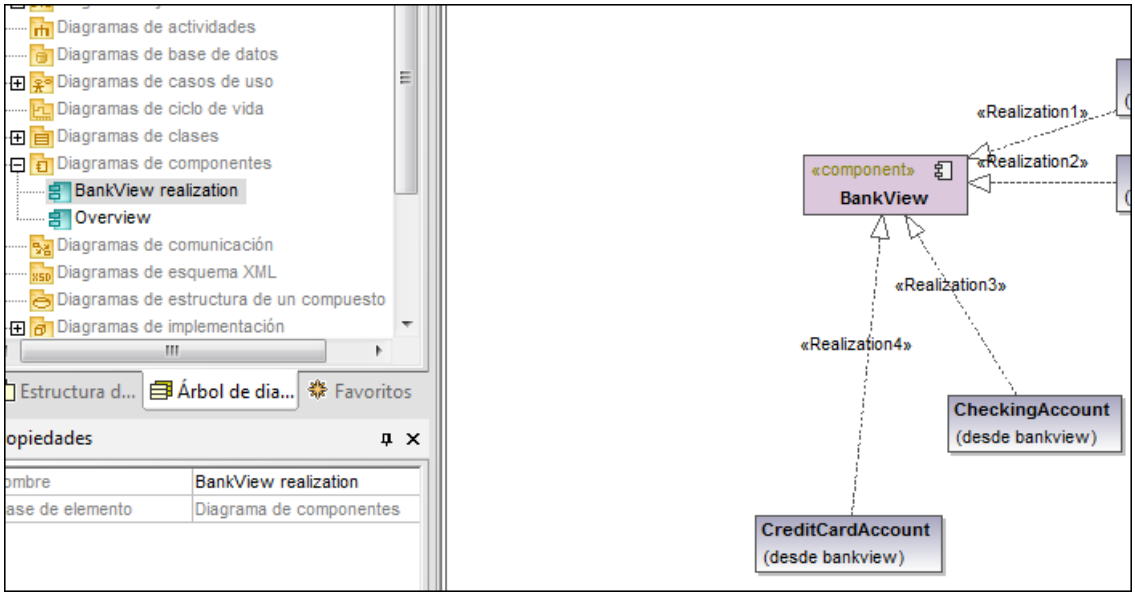
Inserta un elemento `conector`, que se puede usar para conectar dos o más instancias de una parte o de un puerto. El conector define la relación entre los objetos e identifica la comunicación

entre los roles.

Dependencia (Enlace de roles)
Inserta el elemento `dependencia`, que indica qué rol desempeña cada elemento conectable del clasificador o de la operación en la colaboración.

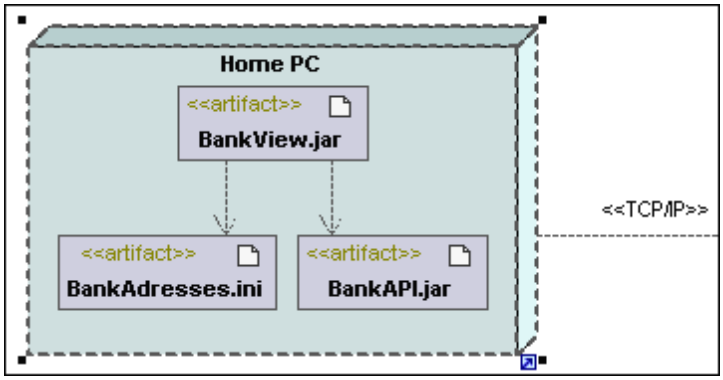
9.2.3 Diagrama de componentes

Para más información sobre cómo añadir componentes al diagrama consulte el apartado Diagramas de componentes.



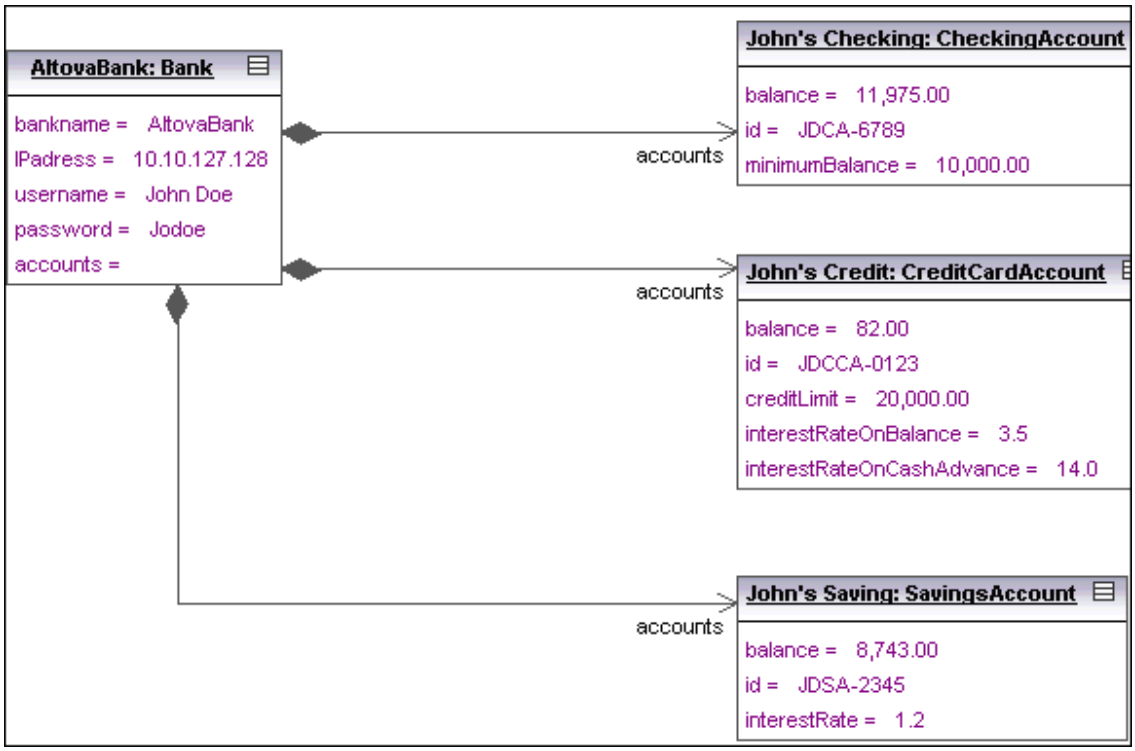
9.2.4 Diagrama de implementación

Para más información sobre cómo agregar nodos y artefactos al diagrama consulte el apartado Diagramas de implementación del tutorial.



9.2.5 Diagrama de objetos

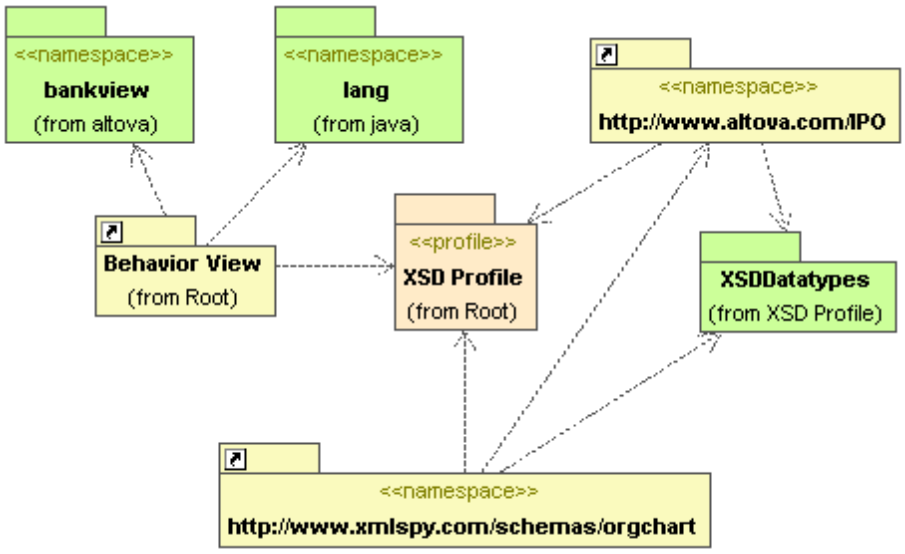
Para más información sobre cómo agregar objetos/instancias nuevos al diagrama consulte el apartado Diagramas de objetos del tutorial.



9.2.6 Diagrama de paquetes

Los diagramas de paquetes ilustran la organización de los paquetes y de sus elementos, así como sus correspondientes espacios de nombres. Además en UModel puede crear hipervínculos para navegar hasta el contenido del paquete correspondiente.

Los paquetes se dibujan en forma de carpetas y se pueden usar en cualquier diagrama UML, aunque se usan sobre todo en diagramas de casos de uso y de clases.



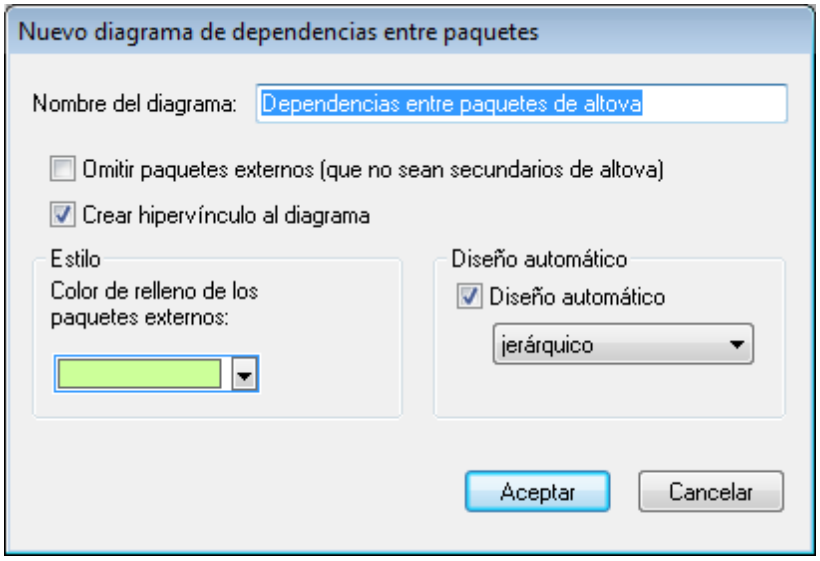
Generación automática de diagramas de dependencias entre paquetes

UModel tiene una función para generar diagramas de dependencias entre paquetes a partir de cualquier paquete del panel Estructura del modelo.

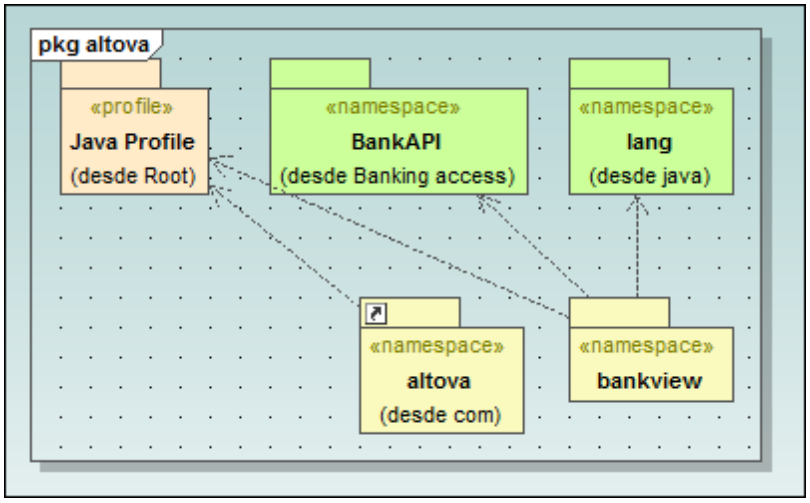
Esta función crea vínculos de dependencia entre los paquetes si existen referencias entre sus elementos de modelado. P. ej. si hay dependencias entre clases, si hay clases derivadas o si los atributos tienen tipos que están definidos en otro paquete.

Para generar un diagrama de dependencias entre paquetes:

1. En la Estructura del modelo haga clic con el botón derecho en un paquete (p. ej. `altova`) y elija **Mostrar en un diagrama nuevo de | Dependencias entre paquetes...**. Esto abre el cuadro de diálogo "Nuevo diagrama de dependencias entre paquetes".



2. Seleccione las opciones que necesite en el cuadro de diálogo y haga clic en **Aceptar**.



UModel genera un diagrama nuevo y muestra las dependencias entre los paquetes del paquete **altova**.

9.2.6.1 *Insertar elementos*

Usar iconos de la barra de herramientas:

1. Haga clic en un icono de la barra de herramientas Diagrama de paquetes.
2. Haga clic en el área de trabajo del diagrama en el que desea insertar el elemento. Para insertar varios elementos del tipo seleccionado, mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo.

Arrastrar elementos desde la Estructura del modelo hasta el diagrama de paquetes:

1. En la Estructura del modelo busque el elemento que quiere insertar en el otro diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).
2. Arrastre el elemento hasta el diagrama de paquetes.

Paquete

Inserta el elemento `paquete` en el diagrama. Los paquetes sirven para agrupar elementos y para aportar un espacio de nombres para los elementos agrupados. Al ser un espacio de nombres, un paquete puede importar elementos concretos de otros paquetes o todos sus elementos. Los paquetes también se pueden combinar con otros paquetes.

Perfil

Inserta el elemento `perfil`, un tipo de paquete que se puede aplicar a otros.

El paquete `perfiles` se usa para extender el metamodelo UML. La construcción de extensión principal es el estereotipo, que a su vez es parte del perfil. Los perfiles siempre deben estar relacionados con un metamodelo de referencia como UML, es decir, no pueden existir por sí mismos.

 **Dependencia**
Inserta el elemento `dependencia`, que indica una relación de proveedor/cliente entre los elementos de modelado (en este caso paquetes o perfiles).

 **ImportaciónDePaquete**
Inserta una relación de importación `<<import>>` que muestra que los elementos del paquete incluido se importarán en el paquete de destino. El espacio de nombres del paquete de destino obtiene acceso al espacio de nombres incluido. El espacio de nombres del paquete incluido no se ve afectado.

Nota: los elementos `private` de un paquete no se pueden combinar ni importar.

 **CombinaciónDePaquete**
Inserta una relación de combinación `<<merge>>` que muestra que los elementos del paquete combinado (de origen) se importarán en el paquete de destino, incluido el contenido importado del paquete combinado (de origen).

Si en el paquete de destino existe el mismo elemento, las definiciones de estos elementos se expanden con las del paquete de destino. Los elementos actualizados o agregados se señalan por medio de una relación de generalización que apunta al paquete de origen.

Nota: los elementos `private` de un paquete no se pueden combinar ni importar.

 **AplicaciónDePerfil**
Inserta una aplicación de perfil que muestra qué perfiles se aplicaron al paquete. Se trata de un tipo de importación de paquete que afirma que un perfil se aplicó a un paquete.

El perfil extiende el paquete al que se aplicó. Al aplicar un perfil (usando el icono **AplicaciónDePerfil**), todos los estereotipos que forman parte del perfil también estarán a disposición del paquete.

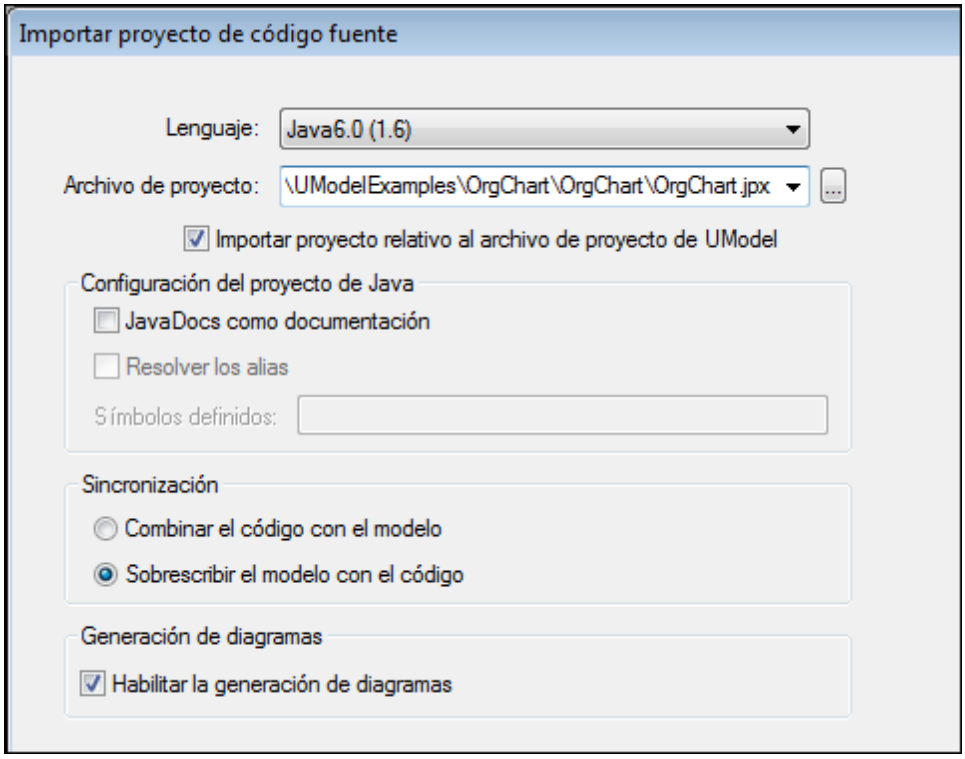
Los nombres de perfil aparecen en forma de líneas discontinuas con puntas de flecha que van del paquete hasta el perfil aplicado y llevan la etiqueta `<<apply>>`.

9.2.6.2 Generar diagramas de paquetes al importar código o binarios

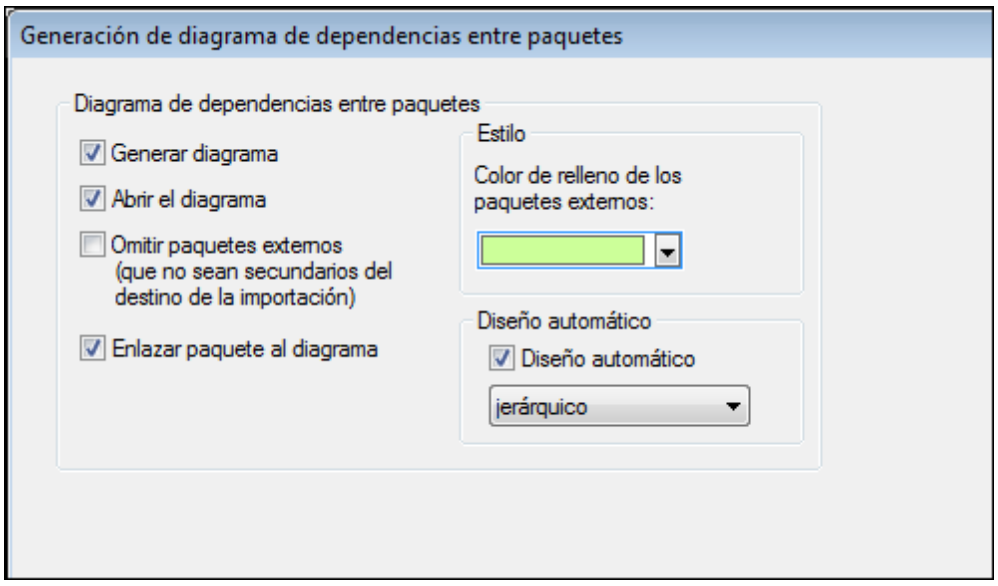
Con UModel puede generar diagramas de paquetes al importar código fuente o binarios en el proyecto de UModel (consulte los apartados Importar código fuente en proyectos e Importar binarios Java, C# y VB).

Siga las instrucciones del asistente para la importación y asegúrese de que:

1) La casilla *Habilitar generación de diagramas* está marcada en el cuadro de diálogo "Importar proyecto de código fuente", "Importar tipos binarios" o "Importar directorio de código fuente".

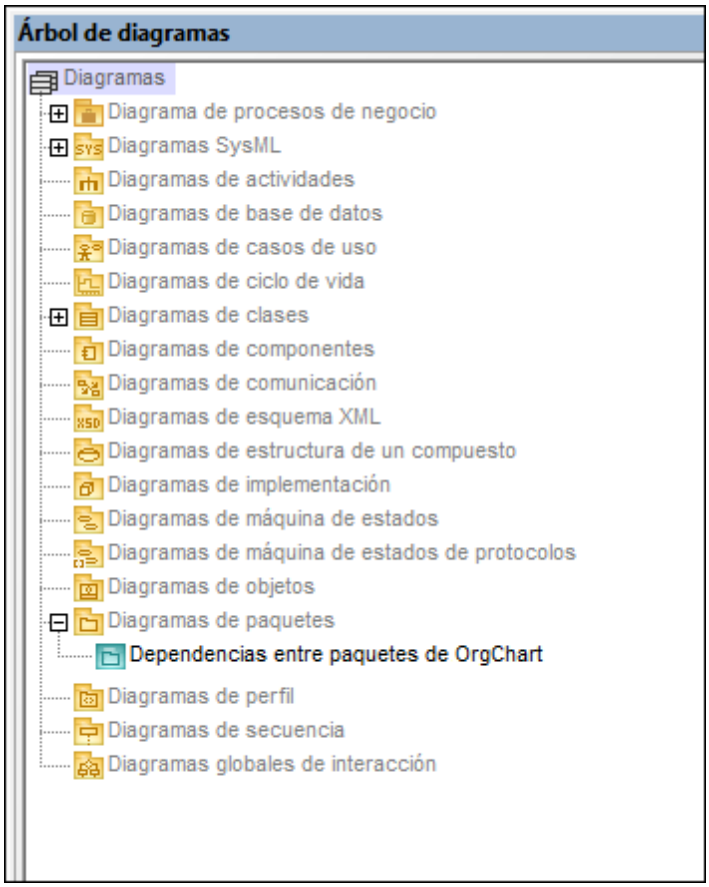


2) La opción *Generar diagrama* está marcada en el cuadro de diálogo "Generación de diagrama de dependencias entre paquetes".



3) Una vez finalizada la operación de importación, los diagramas de clases generados

estarán disponibles bajo el nodo `Diagramas de paquetes` del Árbol de diagramas.



9.2.7 Diagrama de perfil y estereotipos

Sitio web de Altova: Diagramas de perfil UML

Con los diagramas de perfil de UModel puede definir estereotipos, valores etiquetados y restricciones personales en un diagrama especial.

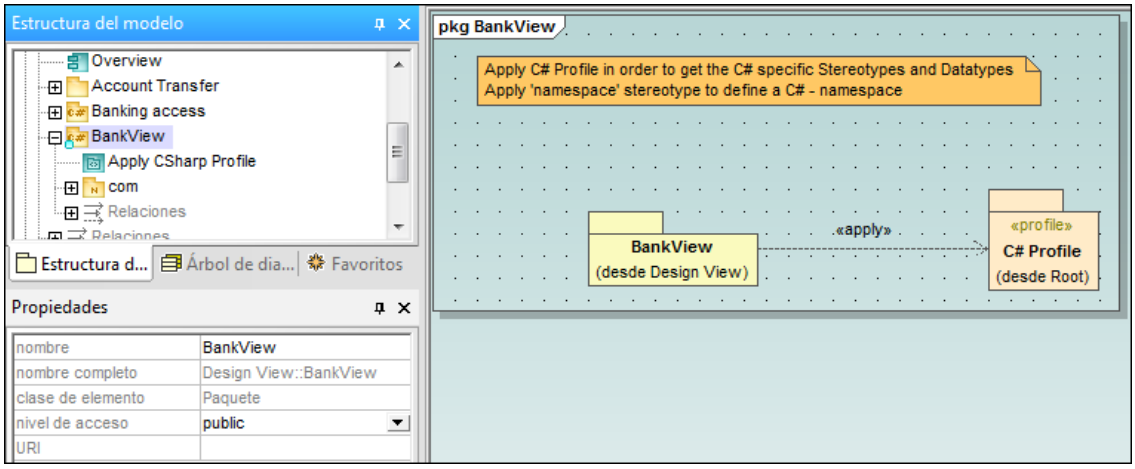
Los perfiles y los estereotipos sirven para extender el metamodelo UML. La construcción de extensión principal es el estereotipo, que forma parte del perfil. Los perfiles siempre deben estar relacionados con un metamodelo de referencia como UML porque no pueden existir por sí solos. UModel ofrece un tipo de diagrama de perfil donde puede definir estereotipos propios.

El archivo de ejemplo `Java Profile.ump` (o `C# Profile.ump` o `VB Profile.ump`) debe aplicarse cuando se creen proyectos nuevos de UModel con el comando de menú **Proyecto | Incluir un subproyecto**. Este perfil aporta los tipos de datos y estereotipos Java y es fundamental para crear código mediante ingeniería de ida y vuelta.

En esta sección explicamos:

- Cómo agregar estereotipos y definir valores etiquetados
- Los estereotipos y las enumeraciones
- Los estilos de estereotipo definidos por el usuario

Para explicar todo ello utilizamos el archivo de ejemplo `Bank_CSharp.ump` (de la carpeta `...\UModelExamples`). El perfil C# se aplicó al paquete **`BankView`**.



- Los perfiles son tipos de paquetes concretos que se aplican a otros paquetes.
- Los estereotipos son metaclases concretas que extienden las clases estándar.
- Los valores etiquetados son los valores de los atributos del estereotipo.

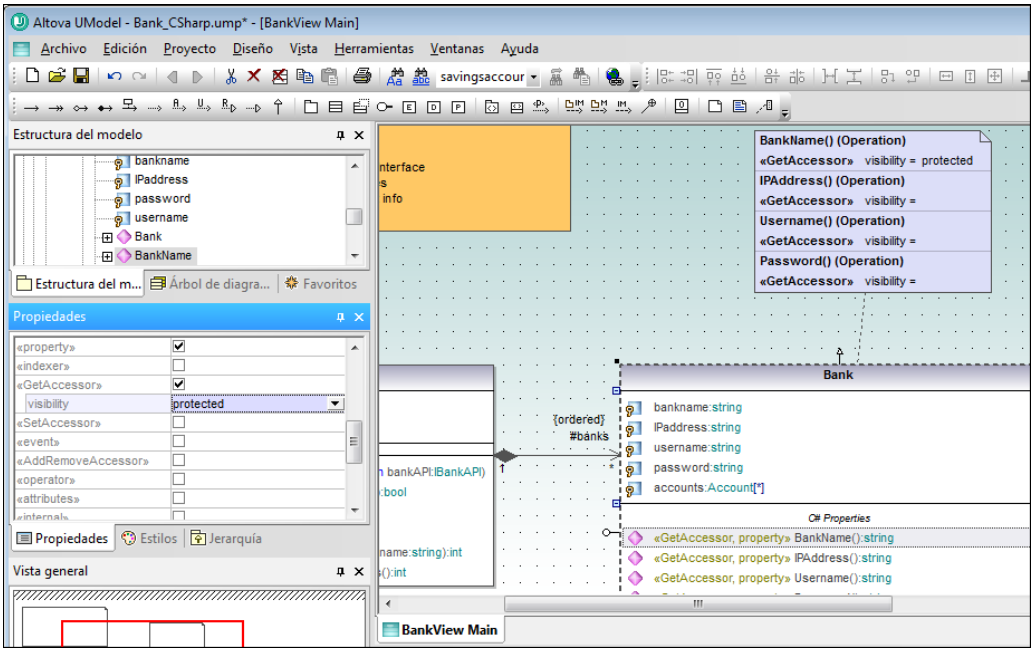
Una `AplicaciónDePerfil` muestra qué perfiles se aplicaron a un paquete y es un tipo de importación de paquete que afirma que un perfil se aplicó a un paquete. El perfil extiende el paquete al que se aplicó. Si se aplica un perfil a un paquete (con ayuda de **AplicaciónDePerfil** ), significa que todos los estereotipos que forman parte del perfil estarán a disposición del paquete.

Las `AplicacionesDePerfil` se dibujan como líneas de puntos con punta de flecha que van **desde el paquete** hasta el perfil aplicado y tienen la etiqueta `<<apply>>`.

,

Estereotipos:
Un estereotipo define cómo se puede extender una metaclase. Se trata de un tipo de clase que extiende otras clases por medio de extensiones. Los estereotipos solamente se pueden crear en perfiles y se representan en diagramas de clases como clases estándar, pero con la palabra clave `<<stereotype>>` encima del nombre de la clase.

- Los estereotipos pueden tener **propiedades**, llamadas **definiciones de etiquetas**.
- Cuando se aplica el estereotipo a un elemento de modelado, los valores de las propiedades reciben el nombre de **valores etiquetados**.
- Cuando se aplican estereotipos que tienen propiedades, los valores etiquetados aparecen automáticamente en un comentario. Para más información sobre cómo personalizar la vista de valores etiquetados consulte este apartado.
- Los estereotipos tienen su propia familia de estilos
- Si el atributo es de tipo `enumeration`, entonces podrá seleccionar uno de los valores predefinidos. También puede insertar/seleccionar el valor concreto en el panel Propiedades (p. ej. `<<GetAccessor>> nivel de acceso = public`).

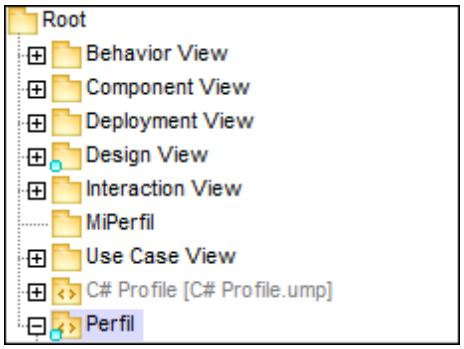


9.2.7.1 *Agregar estereotipos y definir valores etiquetados*

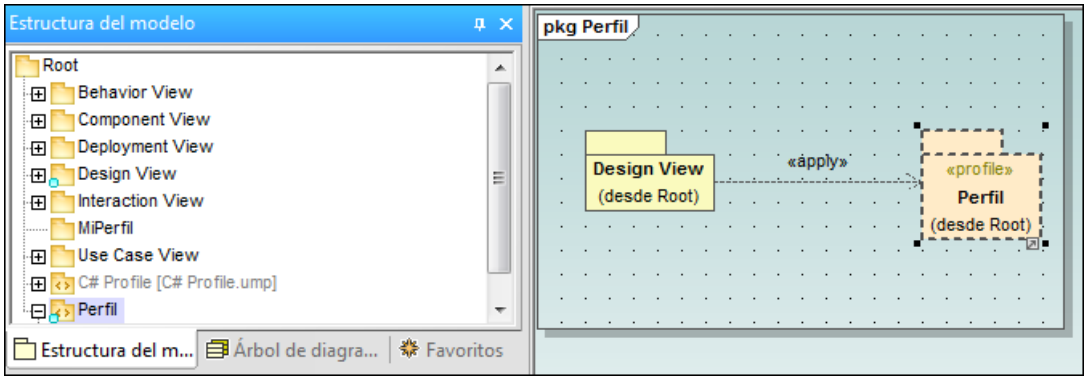
Para los ejemplos que aparecen a continuación utilizamos el archivo `Bank_MultiLanguage.ump` de la carpeta `...\UModelExamples`.

Crear un diagrama de perfil y un estereotipo

1. Cree un perfil nuevo desde el panel Estructura del modelo. Por ejemplo: haga clic con el botón derecho en el paquete **`Root`** y elija **Elemento nuevo | Perfil**. Póngale el nombre **`Perfil`** al nuevo perfil.

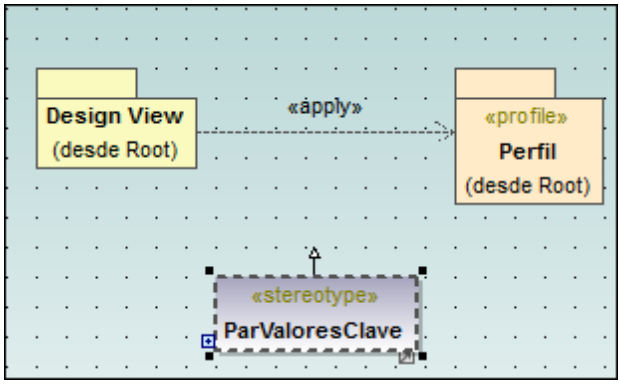


2. Ahora haga clic con el botón derecho en **`Perfil`** y elija **Diagrama nuevo | Diagrama de perfil**.
   Esto añade un diagrama de perfil al paquete seleccionado.
2. Arrastre el perfil recién creado **`Perfil`** desde la Estructura del modelo hasta el área de trabajo del diagrama de perfil.
3. Arrastre también el paquete **`DesignView`** hasta el área de trabajo del diagrama de perfil.
4. Ahora haga clic en el icono **AplicaciónDePerfil**  de la barra de herramientas. Haga clic en el paquete **`DesignView`** y arrastre el puntero del ratón hasta el paquete **`Perfil`**.

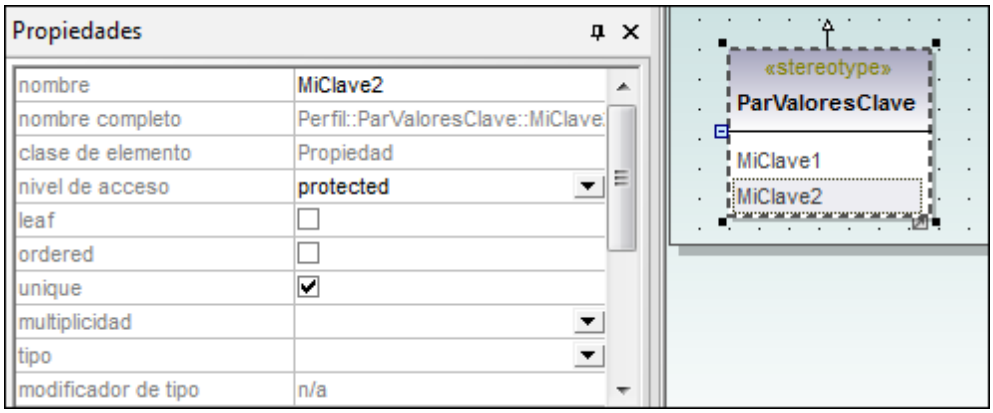


Esto permite usar en el paquete **`DesignView`** y en sus subpaquetes todos los estereotipos que estén definidos en el perfil **`Perfil`**.

5. Haga clic en el icono **Estereotipo** de la barra de herramientas e inserte un estereotipo de clase (p. ej. `ParValoresClave`).



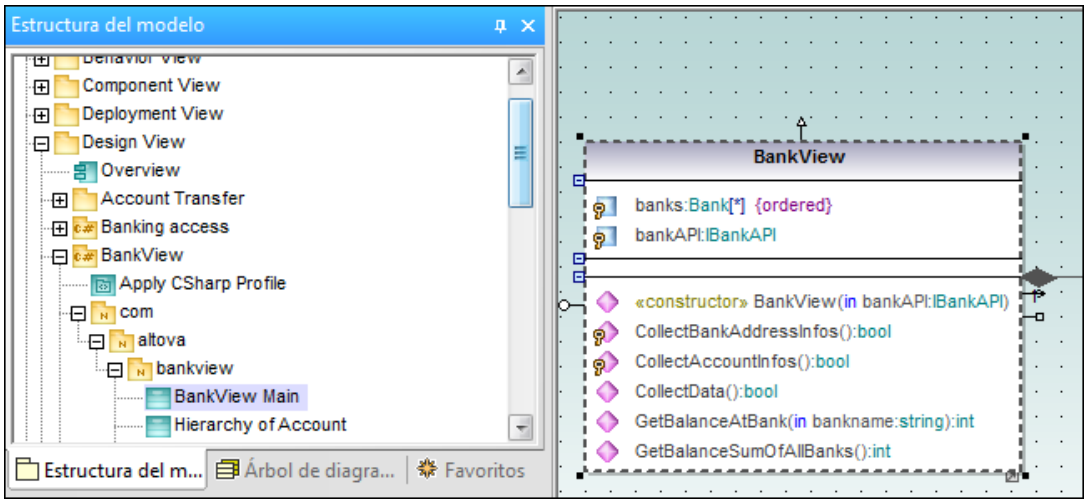
6. Pulse **F7** para agregar un atributo al estereotipo (p. ej. `MiClave1`). Añada otro atributo llamado `MiClave2`.



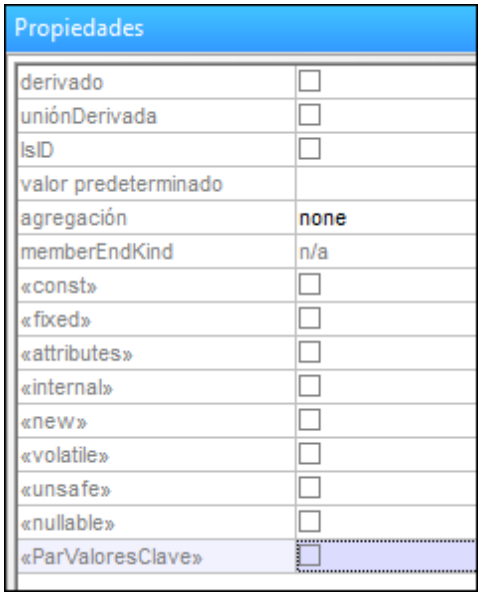
Por ahora hemos terminado de **definir** el estereotipo. En el siguiente paso podemos usar/asignar el estereotipo cuando añadamos atributos a una clase que forme parte del paquete **`BankView`**.

Usar y asignar estereotipos

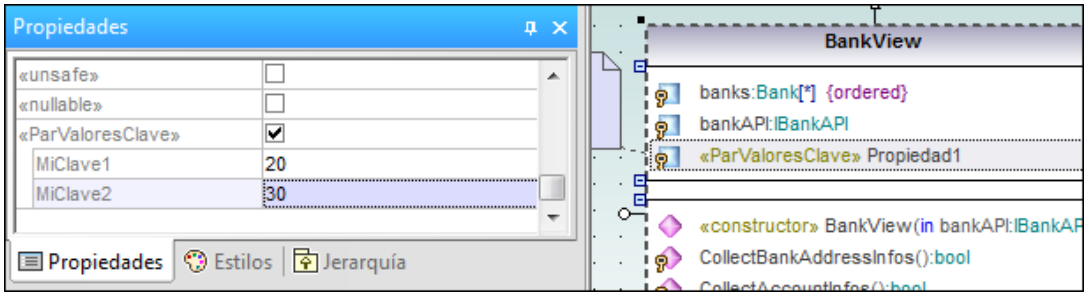
1. En la Estructura del modelo haga doble clic en el icono del diagrama de clases **`BankView Main`**.
   Esto abre el diagrama de clases y muestra las asociaciones que existen entre las clases.



   Ahora vamos a añadir un atributo a la clase **`BankView`** y asignar el estereotipo definido previamente.
2. Haga clic en la clase **`BankView`** y pulse **F7** para añadir un atributo.
3. En el panel Propiedades desplácese hasta el final de la lista. Observe que el estereotipo `ParValoresClave` está disponible en el panel de propiedades.

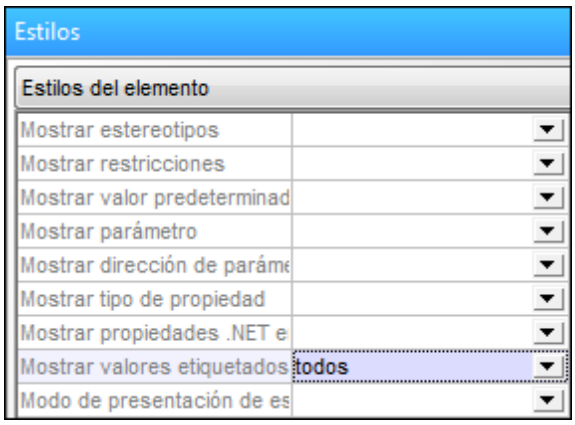


4. Marque la casilla `ParValoresClave` para aplicarlo. Los dos valores etiquetados `MiClave1` y `MiClave2` aparecen ahora bajo el estereotipo en el panel Propiedades.
5. Haga doble clic en el campo de los valores etiquetados e inserte un valor para cada uno.

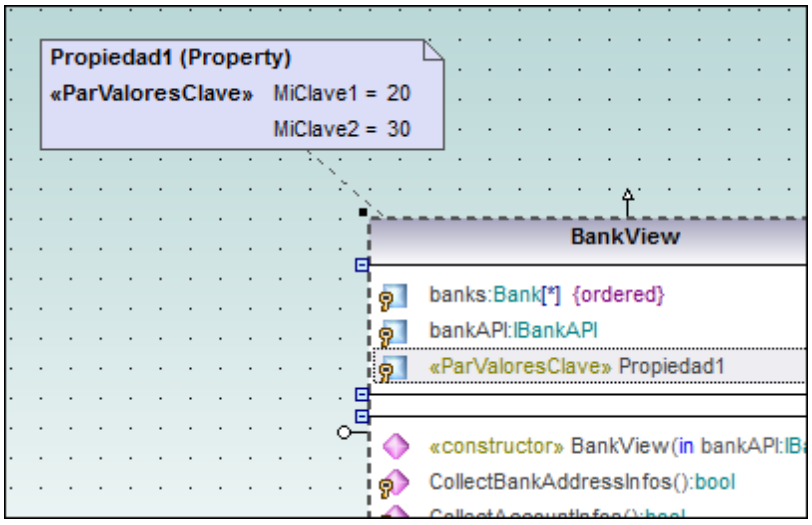


Mostrar valores etiquetados en un diagrama

1. Abra la pestaña *Estilos* y desplácese hasta la entrada `Mostrar valores etiquetados`. Seleccione el valor `todos`.



Ahora el diagrama muestra los valores etiquetados en el elemento `Nota`. Para editar un valor del elemento `Nota`, haga doble clic.



Mostrar valores etiquetados | en compartimento

Este comando del menú contextual muestra los valores etiquetados en un compartimento separado de la clase.

Nota: cuando oculte atributos y operaciones con el comando **Mostrar u ocultar el contenido del nodo...** del menú contextual, los valores etiquetados también se muestran/ocultan junto con el elemento de modelado.

Los extremos de asociación pueden mostrar estereotipos si asigna el valor `verdadero` a la opción `Mostrar estereotipos de memberEnd` en el panel Estilos.

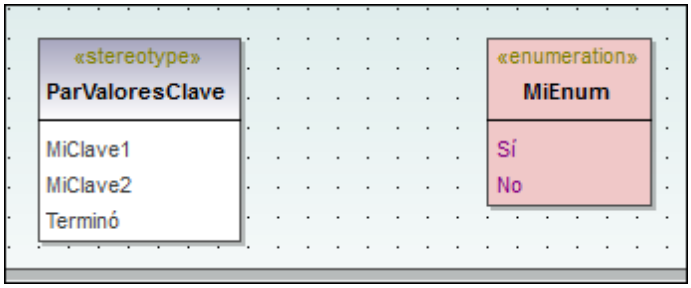
Para más información consulte el apartado Ver valores etiquetados.

9.2.7.2 Estereotipos y enumeraciones

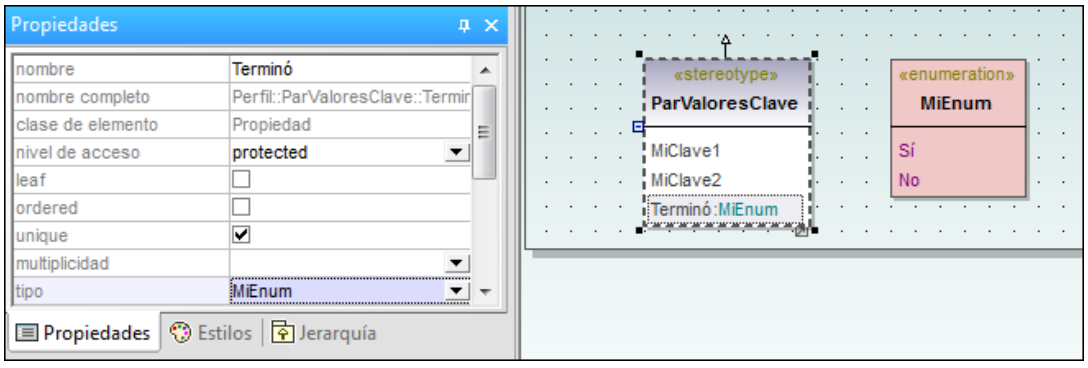
UModel ofrece un método muy eficaz para seleccionar valores enumerados de estereotipos.

Haga clic en la pestaña del diagrama que contiene la definición de estereotipo (es decir, el diagrama de perfil que añadimos en el primer paso):

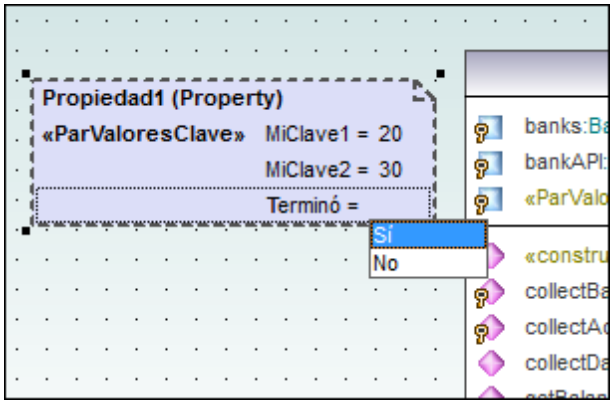
1. Haga clic en el icono **Enumeración** de la barra de herramientas para insertar una enumeración en el diagrama (que contiene el estereotipo previamente definido).
2. Pulse **Mayús+F7** para añadir los LiteralesDeEnumeración `Sí` y `No` a la enumeración. Si lo prefiere puede añadirlos desde el menú contextual.



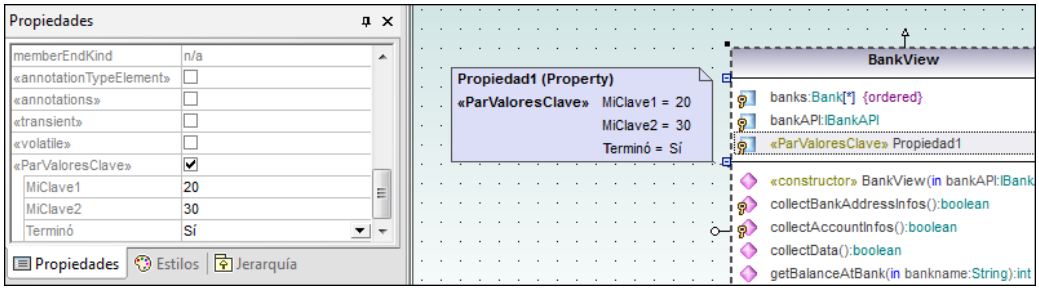
3. Haga clic en la clase estereotipo y pulse **F7** para añadir un atributo/una propiedad nueva (p. ej. `Terminó`).
4. Ahora en el panel Propiedades seleccione el tipo `MiEnum`.



5. Ahora vuelva al diagrama de clases **`BankView Main`**.
6. Observe que la propiedad `Terminó` aparece ahora como valor etiquetado en la nota.

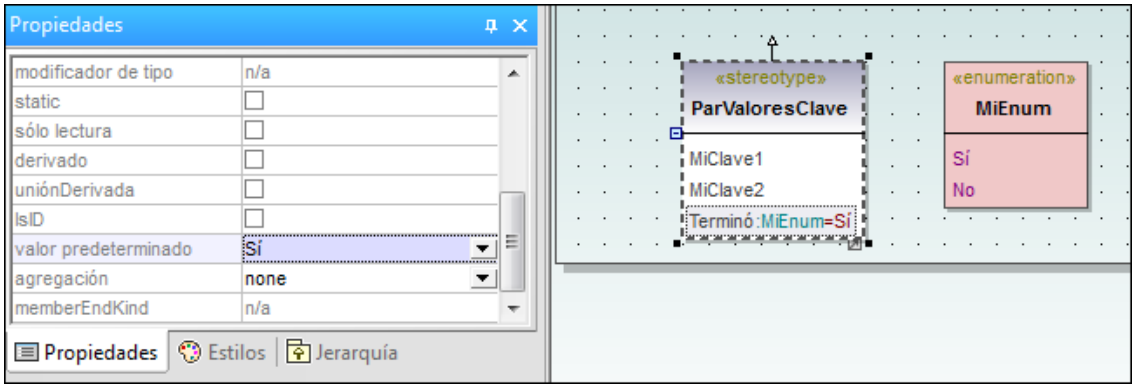


Haga doble clic en el valor etiquetado `Terminó` para ver los valores de enumeración predefinidos. Haga clic en una enumeración para seleccionarla.

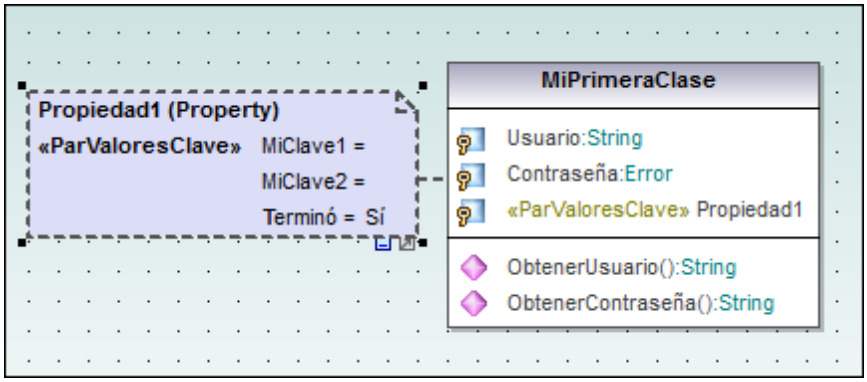


Valores de enumeración predeterminados

UModel también permite definir valores etiquetados **predeterminados**. Cuando añada un atributo al estereotipo, haga doble clic en el campo predeterminado e inserte una de las enumeraciones disponibles como valor predeterminado.



En el ejemplo de la imagen anterior se insertó el valor predeterminado `Sí`.
Cuando añada una propiedad a una clase y esté seleccionado el tipo `MiEnum`, el valor predeterminado se inserta automáticamente como valor etiquetado (es decir, `Terminó = Sí`).

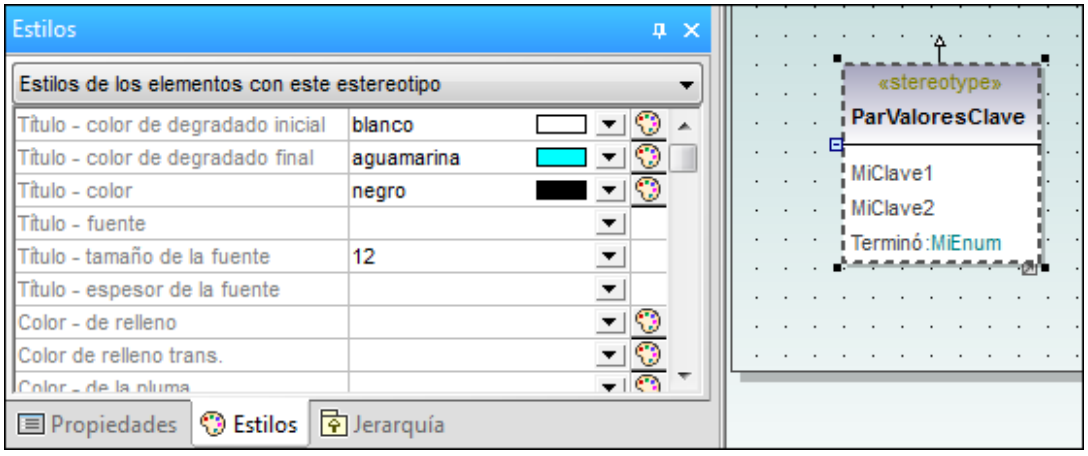


9.2.7.3 Estilos de estereotipo definidos por el usuario

En UModel puede crear estilos definidos por el usuario para cada estereotipo. Esto significa que puede tener fuentes y colores diferentes para cada estereotipo y aplicarlos a las clases que sean del tipo de estereotipo.

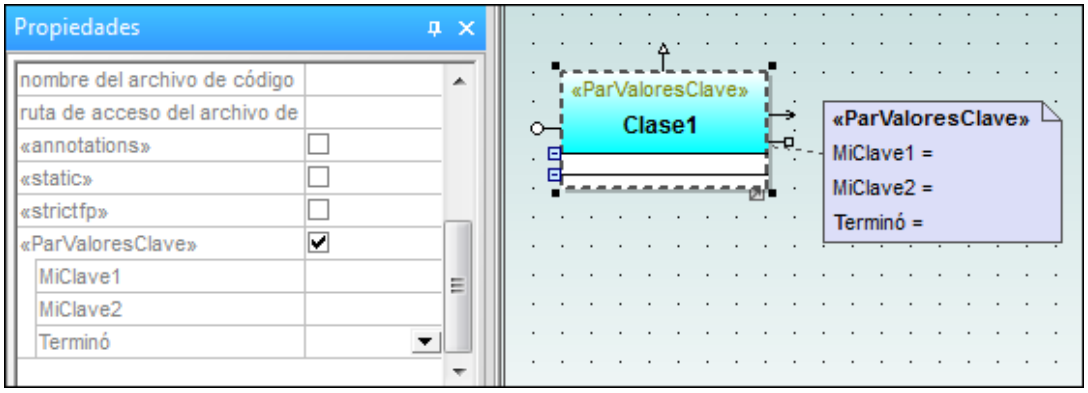
Para crear estilos de estereotipo definidos por el usuario:

1. Haga clic en un estereotipo (p. ej. `ParValoresClave`).
2. Abra la pestaña *Estilos* y, en el cuadro combinado de la parte superior, seleccione la opción **Estilos de elementos con este estereotipo**.
3. Defina los estilos de este estereotipo (p. ej. `Título - color de degradado final = aguamarina`).



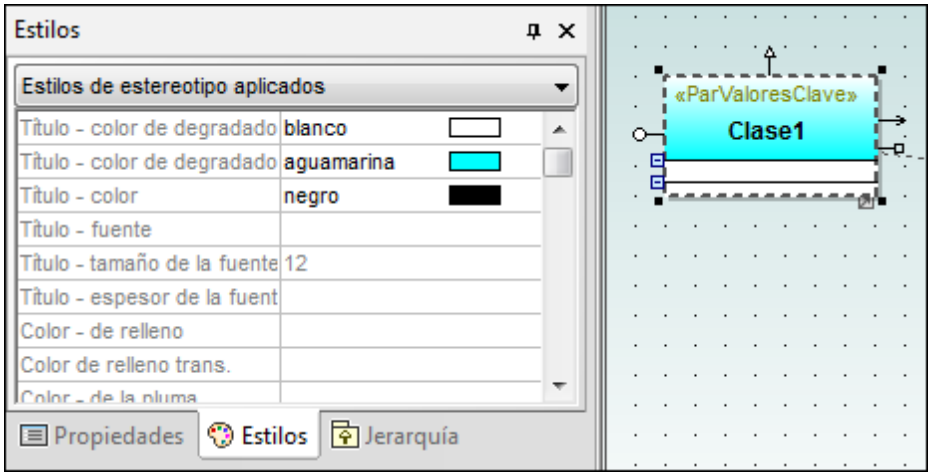
Al hacer clic en la clase del estereotipo, aparecen automáticamente los estilos definidos para el estereotipo en el panel Estilos.

4. Ahora pase al diagrama de clases e inerte una clase nueva.
5. Haga clic en el título o encabezado de la clase y marque la casilla `ParValoresClave` en el panel Propiedades.



La clase nueva tiene los estilos que se asignaron al estereotipo (el color de degradado del título es aguamarina, p. ej.). No olvide que los estilos del estereotipo **no se aplican** si el estereotipo se aplica a una propiedad u operación de la clase.

6. Haga clic en la nueva clase estereotipo del diagrama y después abra la pestaña *Estilos*.
7. En el cuadro combinado situado en la parte superior seleccione la opción **Estilos de estereotipo aplicados**.



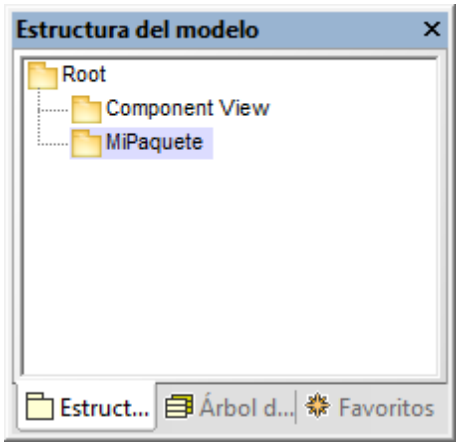
Aquí obtiene una vista previa de las opciones de estilo definidas para este estereotipo, pero no puede cambiar las opciones. Esto debe hacerse desde el diagrama de clases donde se definió el estereotipo.

9.2.7.4 Asignar iconos de estereotipo personalizados

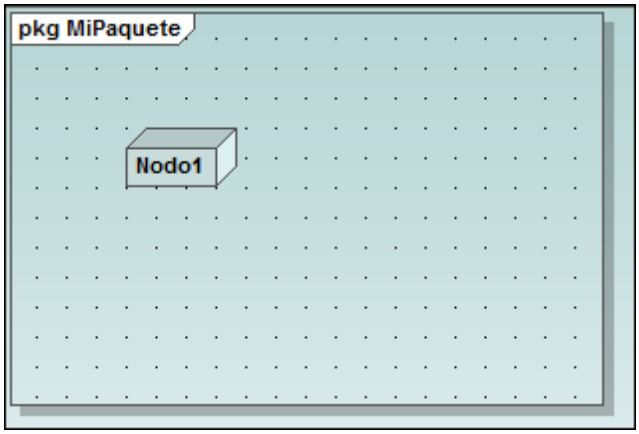
Para terminar, en UModel también puede crear iconos personalizados para sus estereotipos.

Para crear un icono de estereotipo personal:

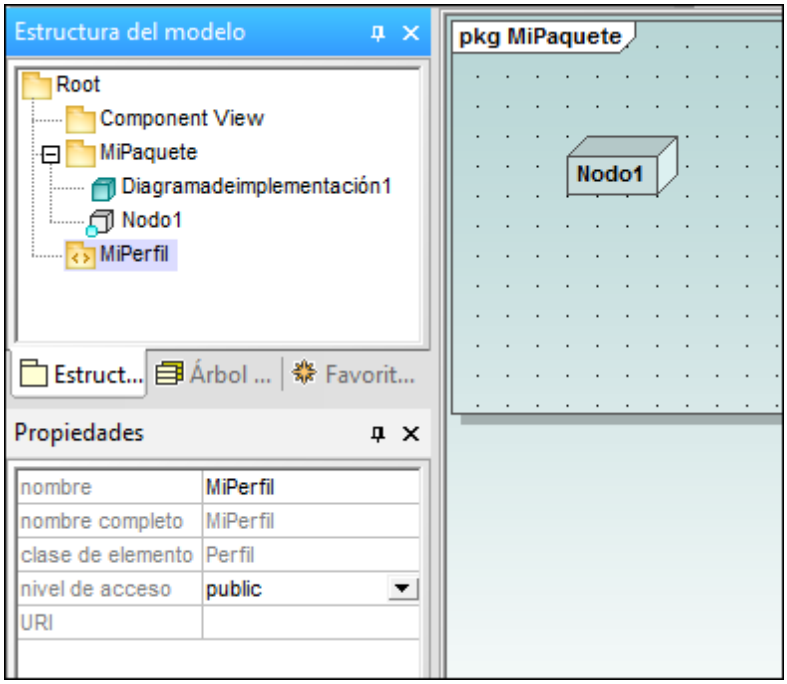
1. Haga clic con el botón derecho en el paquete `Root` y elija **Elemento nuevo | Paquete** en el menú contextual.
2. A este paquete le ponemos el nombre `MiPaquete`.



3. Haga clic con el botón derecho en **MiPaquete** y elija **Diagrama nuevo | Implementación** en el menú contextual.
   Esto crea un diagrama de implementación en **MiPaquete**.
4. Ahora haga clic con el botón derecho en **MiPaquete** y seleccione **Nuevo/a | Nodo**.



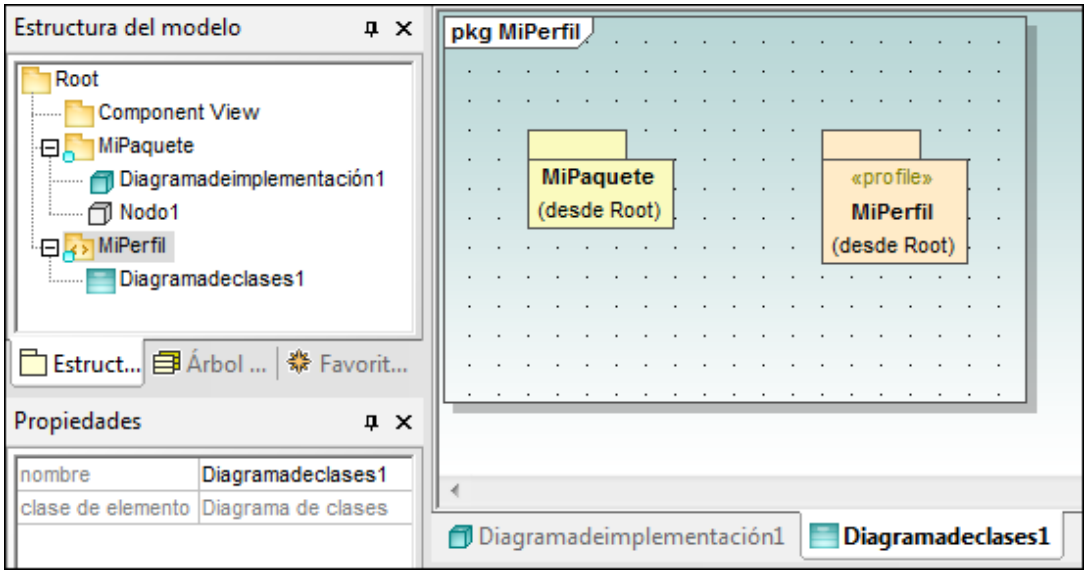
   Esto añade el objeto **Nodo1** al diagrama.
5. En la Estructura del modelo haga clic con el botón derecho en el paquete **Root** y después elija **Elemento nuevo | Perfil**.
6. Ponga el nombre **MiPerfil** a este nuevo perfil.



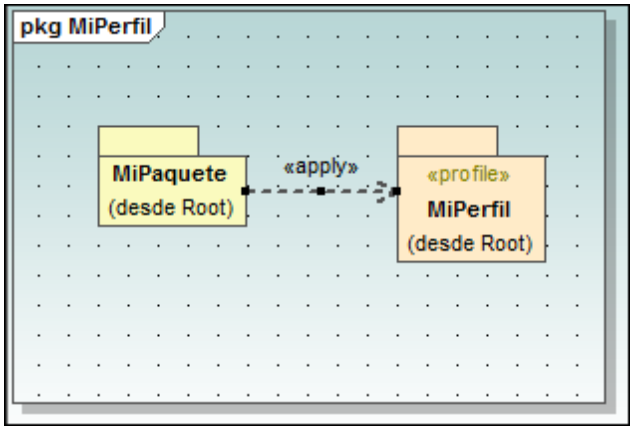
7. Haga clic con el botón derecho en **`MiPerfil`** y elija **Diagrama nuevo | Diagrama de clases**.

Para seleccionar el icono del estereotipo:

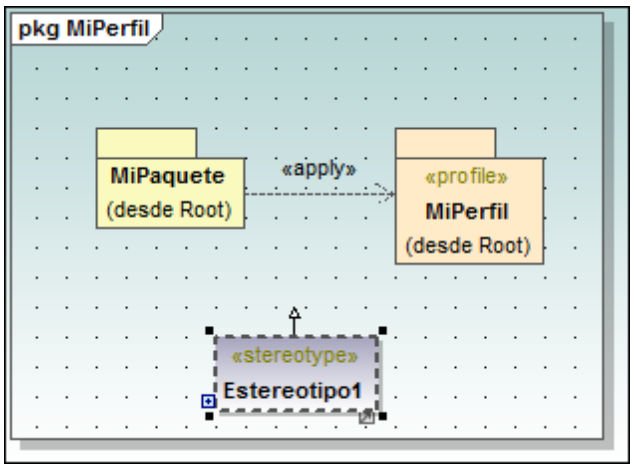
1. Arrastre los paquetes **`MiPaquete`** y **`MiPerfil`** desde la Estructura del modelo hasta el área de trabajo del diagrama de clases.



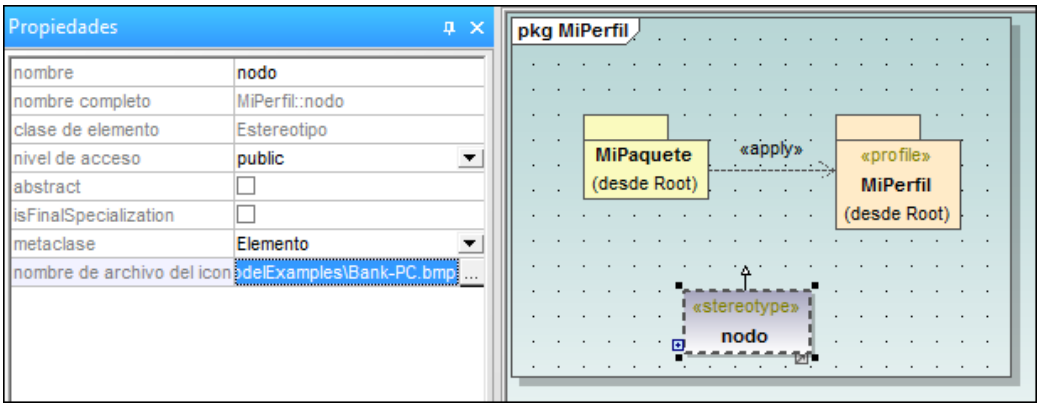
2. Haga clic en el icono **AplicaciónDePerfil**  de la barra de herramientas. Haga clic en **`MiPaquete`** y arrastre el puntero hasta **`MiPerfil`**.



3. Haga clic en el icono **Estereotipo** de la barra de herramientas y haga clic en el paquete `MiPerfil` para insertar el estereotipo.

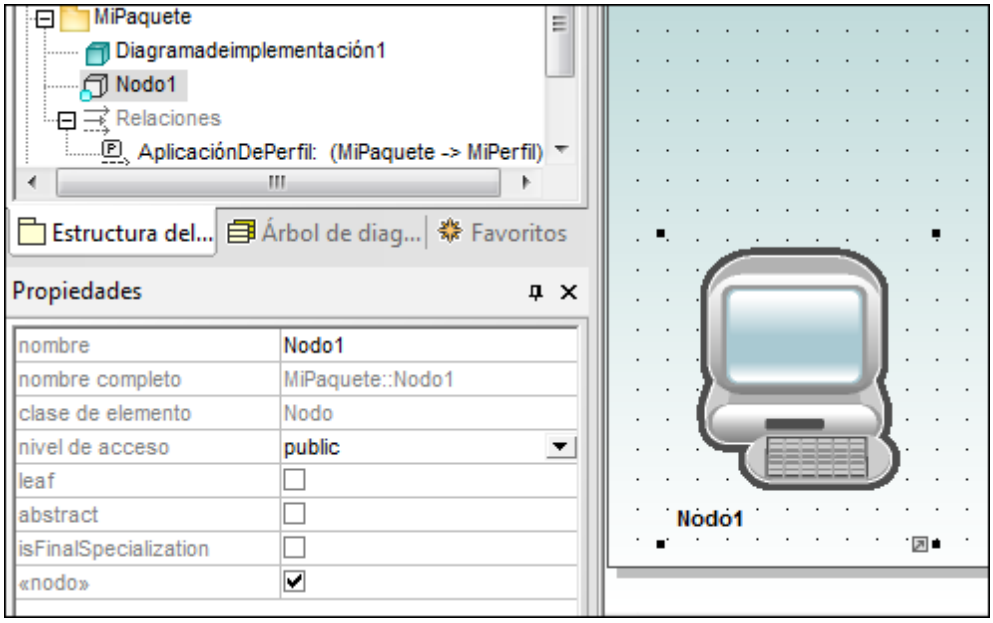


4. Haga clic en el estereotipo y ahora, en el panel Propiedades, cambie el nombre del elemento de `Estereotipo1` a `nodo`.
5. Cambie también el valor de la propiedad `metaclase` de `Elemento` a `Nodo`.
6. En el campo `nombre del archivo del icono` inserte la ruta de acceso de la imagen que desea usar como icono del estereotipo (o pulse el botón **Examinar** para seleccionar la ruta de acceso).



7. Vuelva al diagrama de implementación (haciendo clic en la pestaña *DiagramaDeImplementación1*).

8. Haga clic en el elemento **Nodo1** del diagrama.
9. En el panel Propiedades marque la casilla <<nodo>>.



La representación del icono del estereotipo ahora es la imagen seleccionada.

Nota: con los iconos personalizados se utiliza el color RGB 82,82,82 como color de fondo (es decir, transparente).

9.3 Otros diagramas

UModel ofrece funciones para importar y generar esquemas XML del W3C y puede generar código a partir de ellos mediante ingeniería directa e inversa.

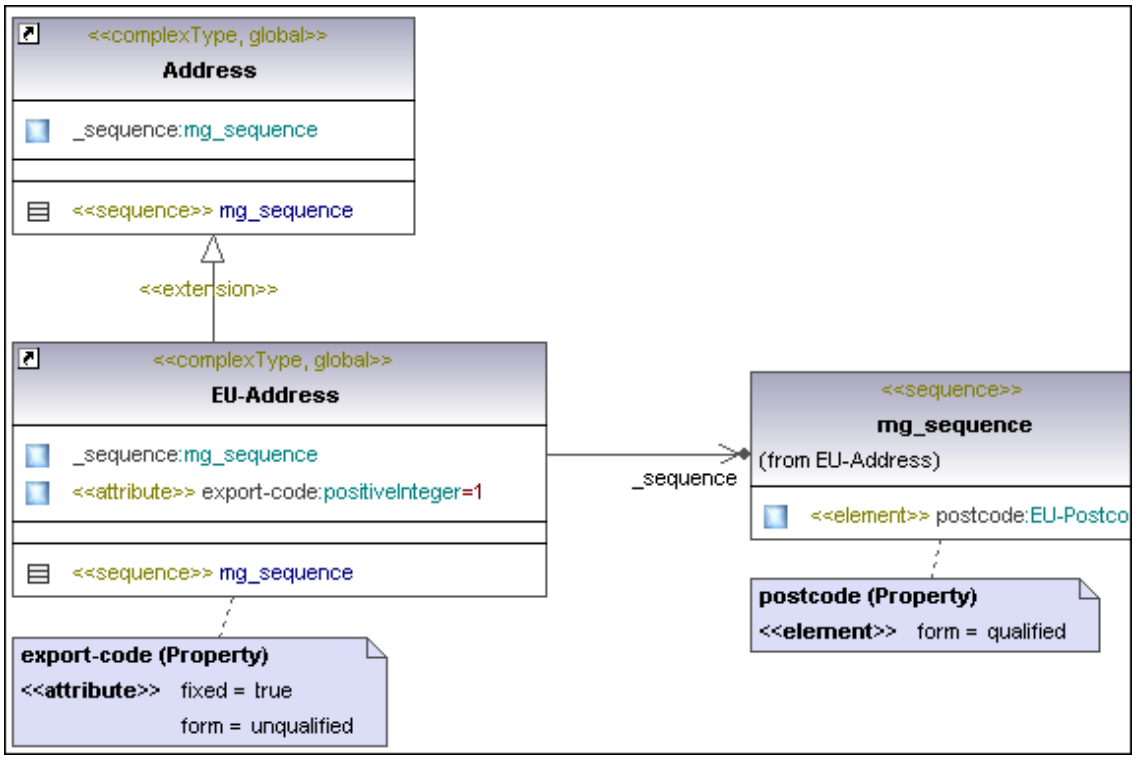
9.3.1 Diagramas de esquema XML

Sitio web de Altova: Esquemas XML en UML

Los diagramas de esquema XML dibujan los componentes de un esquema XML usando la notación UML. Los elementos globales (es decir, los elementos `element`, `simpleType`, `complexType`) se dibujan como clases o tipos de datos y los atributos se representan dentro del compartimento de atributos.

En el compartimento de operaciones no hay operaciones. El elemento de modelado *valor etiquetado* se utiliza para presentar los detalles del esquema.

Para ver las correspondencias entre los elementos UML y los elementos/atributos de XML Schema, consulte el apartado Correspondencia entre elementos de XML Schema y de UModel.

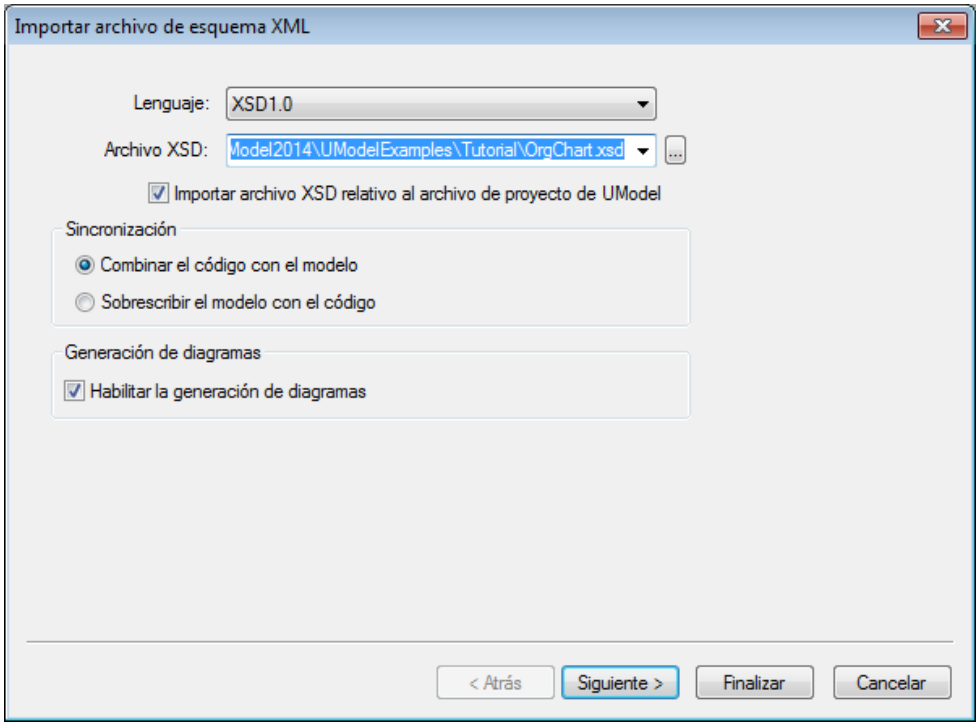


Nota: en UModel no se pueden importar esquemas XML no válidos. Los esquemas XML no se validan durante la importación ni durante su creación en UModel. Los esquemas XML tampoco se revisan durante el proceso de revisión sintáctica. Lo que sí se comprueba durante la importación es si el formato XML del esquema XML es correcto.

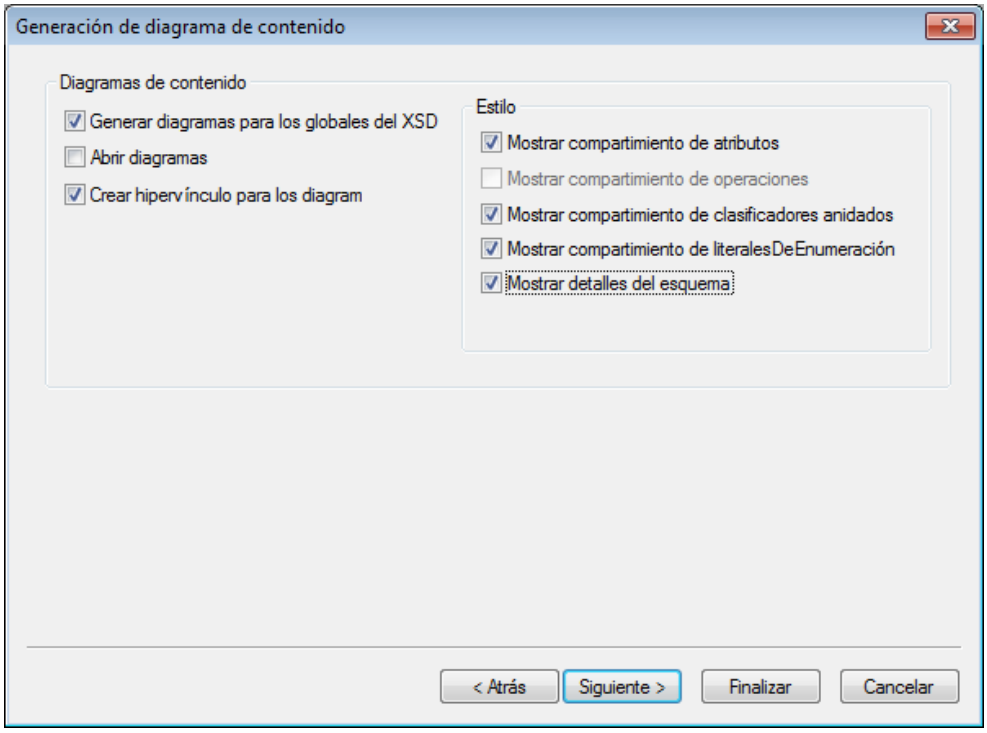
9.3.1.1 Importar esquemas XML

Para importar un solo esquema XML:

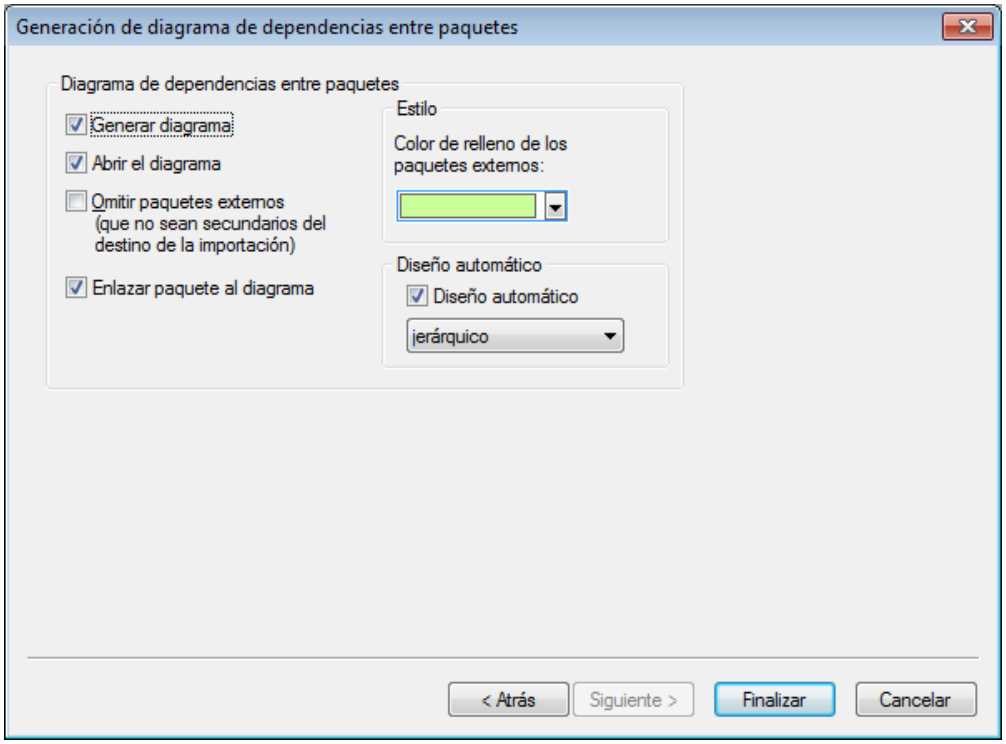
1. Seleccione el comando de menú **Proyecto | Importar archivo de esquema XML**.



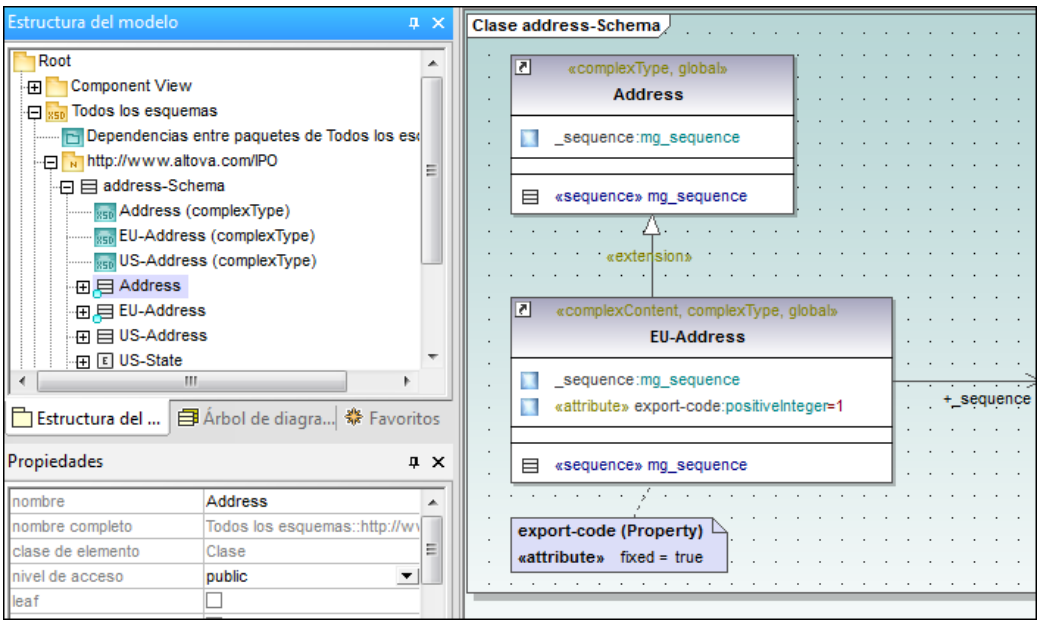
2. Asegúrese de que la casilla *Habilitar la generación de diagramas* esté activa. Haga clic en **Siguiente** para continuar.



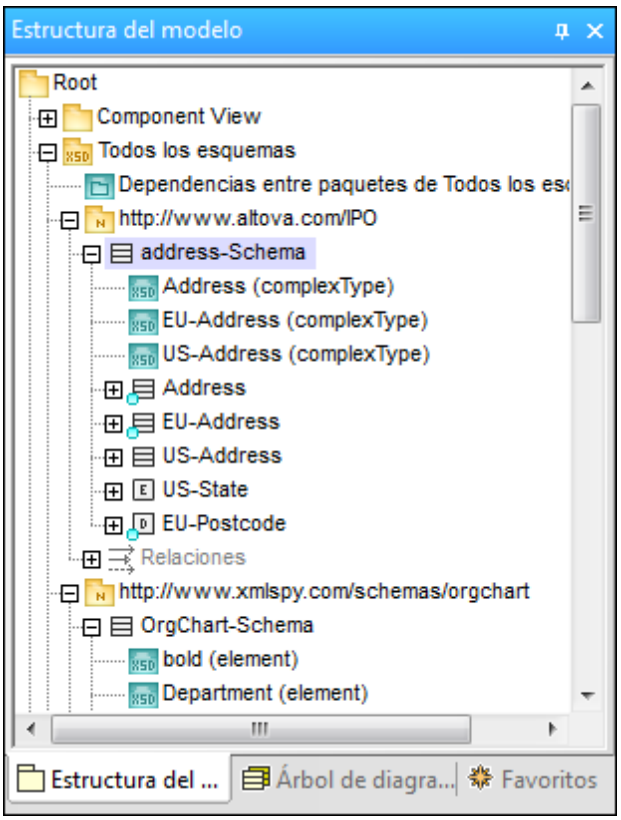
3. En la siguiente pantalla, elija las opciones en el grupo *Diagramas de contenido*. La primera opción crea un diagrama distinto por cada elemento global del esquema.
4. En el grupo de opciones *Estilo* seleccione qué compartimentos deben aparecer en los diagramas de clases. Si marca la casilla *Mostrar detalles del esquema como valores etiquetados*, los detalles del esquema aparecerán en el elemento de modelado `ValorEtiquetado`.
5. Haga clic en **Siguiente** para configurar la generación de diagramas de dependencias entre paquetes.



6. Haga clic en **Finalizar** para empezar a importar el esquema XML.
   El esquema se importa en UModel y los diagramas generados se abren en el área de trabajo. Por ejemplo, en la imagen siguiente puede ver el contenido del diagrama **`EU-Address`** (`complexType`).

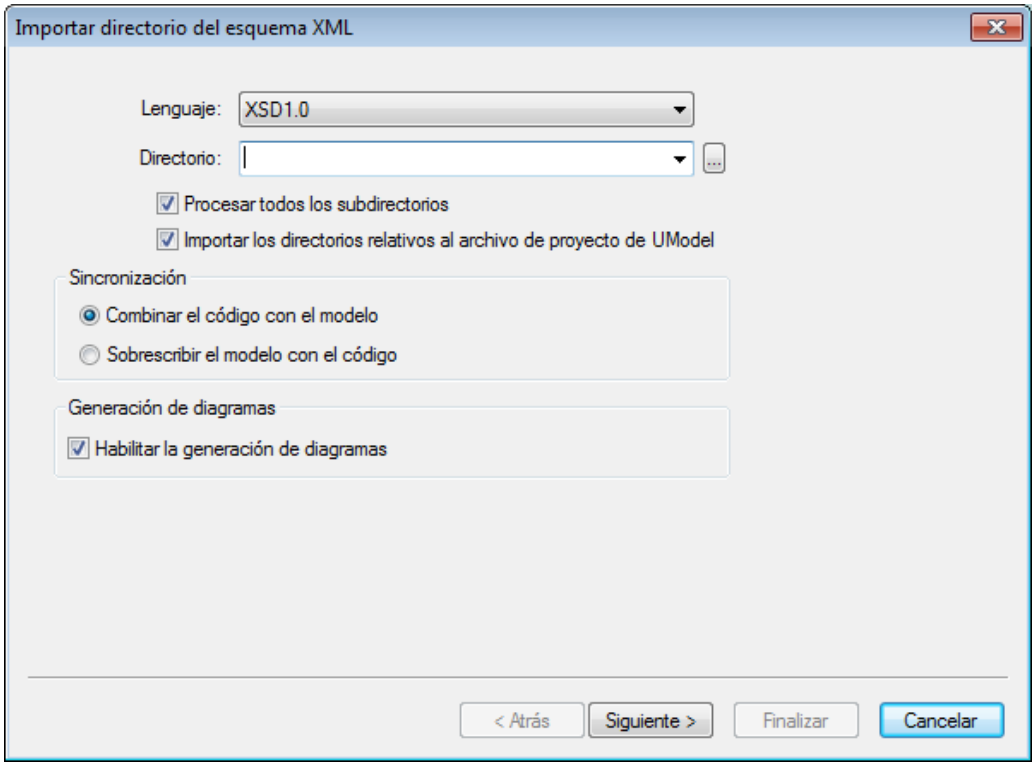


Nota: También se creó un paquete nuevo llamado **`Todos los esquemas`**, que se configuró como raíz del espacio de nombres XSD. Todos los elementos XSD globales generan un diagrama de esquema XML y los diagramas están bajo los paquetes de los correspondientes espacios de nombres.



Para importar varios esquemas XML:

1. Seleccione el comando de menú **Proyecto | Importar directorio del esquema XML**.

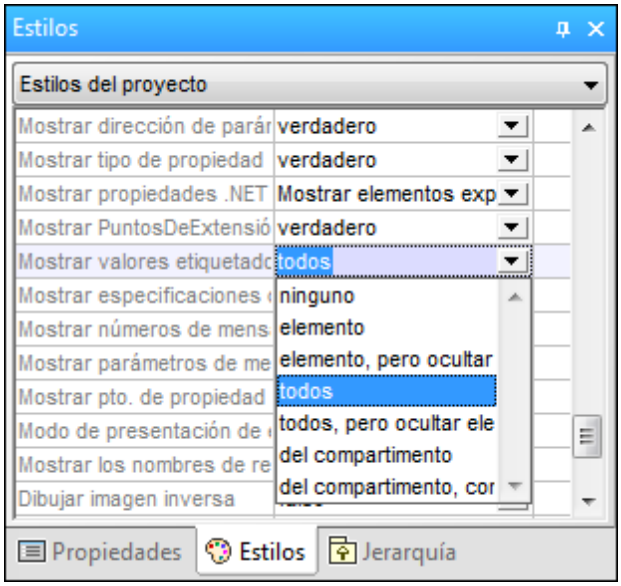


2. Marque la casilla *Procesar todos los subdirectorios* si quiere importar los esquemas de todos los subdirectorios.
   El resto del proceso de importación sigue el mismo orden que cuando se importa un solo esquema XML (ver más arriba).

   Nota: si un esquema XML incluye o importa otros esquemas, estos esquemas se importan automáticamente también.

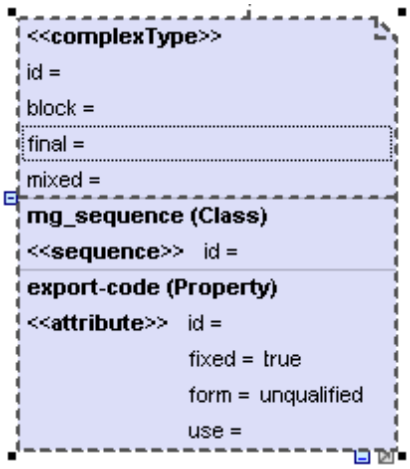
Ver valores etiquetados: detalles del esquema

En UModel tiene la opción de configurar los detalles del esquema que se presentan como valores etiquetados en el elemento `ValorEtiquetado`. Esto se hace en el campo `Mostrar valores etiquetados` del panel Estilos o haciendo clic en el icono **Activar o desactivar el modo compacto** (situado en la esquina inferior derecha del elemento Valor etiquetado). Con este icono puede alternar entre los diferentes estados del valor etiquetado (p. ej. `todos`, `todos, pero ocultar elementos ocultos`, etc.)

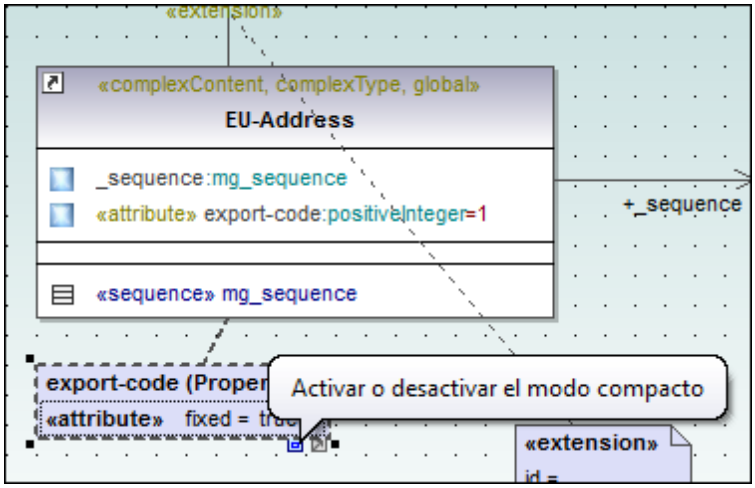


Nota: el estilo seleccionado para los valores etiquetados en la pestaña Estilos se utiliza para todo el proyecto. En cambio, si hace clic con el botón derecho en una clase y elige el comando **Valores etiquetados | Todos**, por ejemplo, esta opción solamente se aplica a la clase seleccionada.

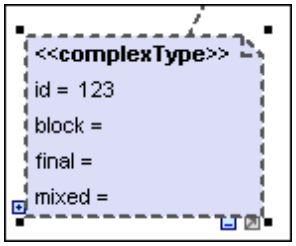
Mostrar valores etiquetados | todos
Esta opción muestra los valores etiquetados de la clase y de atributos propios y operaciones propias, etc.



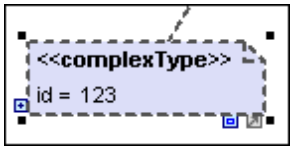
Mostrar valores etiquetados | Todos, pero ocultar elementos vacíos
Esta opción solamente muestra los valores etiquetados que tienen asignado un valor (p. ej. `fixed=true`).



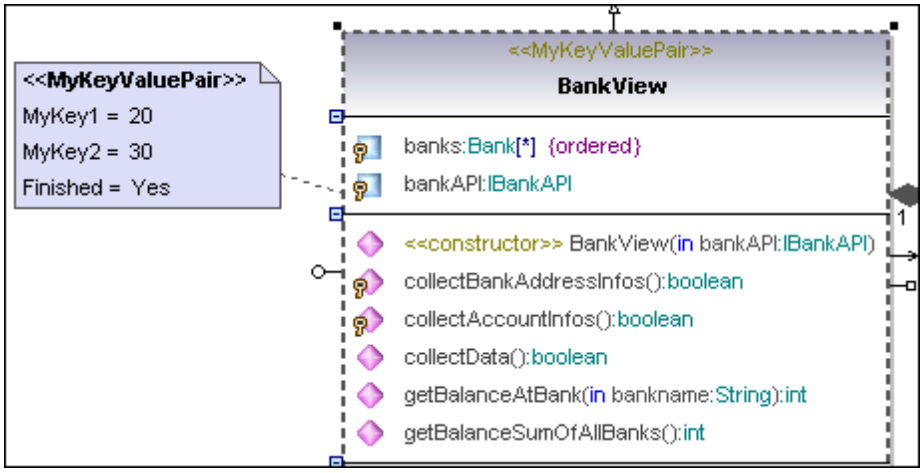
Mostrar valores etiquetados | Todos
Esta opción muestra los valores etiquetados de la clase pero **no** los de los atributos y operaciones propios, etc.



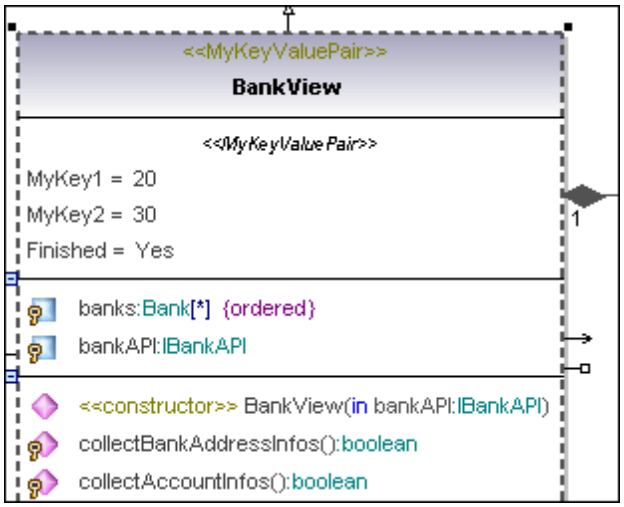
Mostrar valores etiquetados | Elemento, pero ocultar elementos vacíos
Esta opción solamente muestra los valores de elemento etiquetados de una clase, sin los atributos propios, que tengan un valor (p. ej. `id=123`).



Mostrar valores etiquetados | en compartimento



Esta opción muestra los valores etiquetados en un compartimento distinto de la clase.



Mostrar valores etiquetados | En compartimento, pero ocultar elementos vacíos
Esta opción solamente muestra los valores de elemento etiquetados de una clase, sin los atributos propios, que tengan un valor.

Anotación de XML Schema:
Cuando importe esquemas XML, tenga en cuenta que en la ventana Documentación solamente aparece la primera anotación de un elemento `complexType` o `simpleType`.

9.3.1.2 Insertar elementos

Usar iconos de la barra de herramientas:

1. Haga clic en un icono de la barra de herramientas Diagrama de esquema XML.
2. Haga clic en el área de trabajo del diagrama en el que desea insertar el elemento.

Para insertar varios elementos del tipo seleccionado, mantenga pulsada la tecla **Ctrl** mientras hace clic en el área de trabajo.

Arrastrar elementos ya disponibles a un diagrama de esquema XML:

La mayoría de los elementos de un diagrama de esquema XML se pueden insertar en otro.

1. En la Estructura del modelo busque el elemento que quiere insertar en el otro diagrama (puede usar el cuadro de búsqueda o pulsar **Ctrl+F** para buscar el elemento).
2. Arrastre el elemento hasta el diagrama de esquema XML.

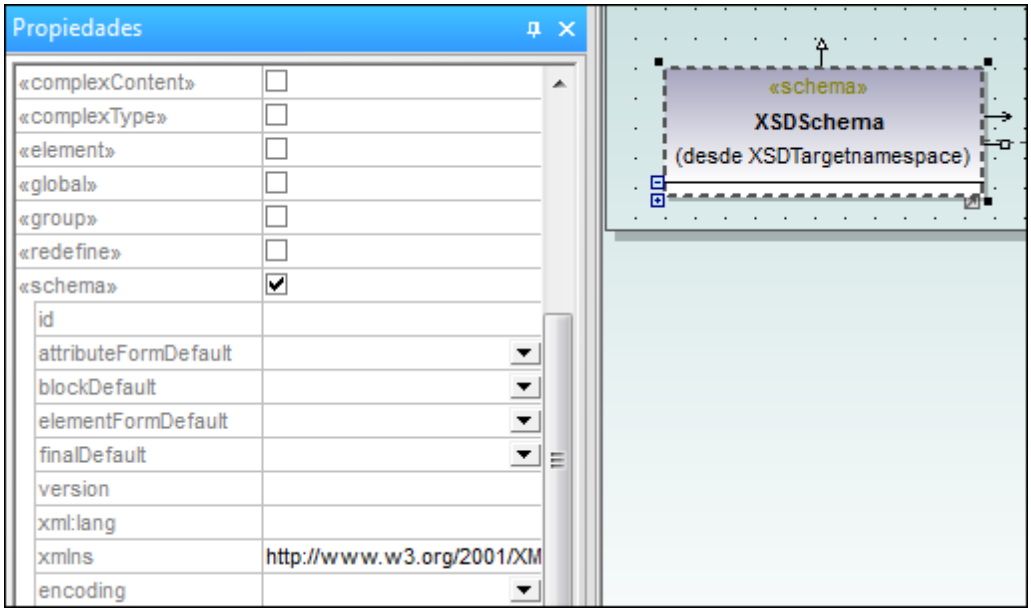
Nota: también puede usar los comandos **Copiar** y **Pegar sólo en el diagrama** para insertar elementos.

targetNamespace XSD

Inserta/define el espacio de nombres de destino para el esquema. Este espacio de nombres de destino debe pertenecer al paquete **`Raíz de espacio de nombres XSD`**.

schema XSD

Inserta/define un esquema XML. El esquema XML debe pertenecer a un paquete del espacio de nombres de destino XSD.



Element (global) XSD

Inserta un elemento global en el diagrama. Tenga en cuenta que también se genera automáticamente una propiedad en el compartimento de atributos.

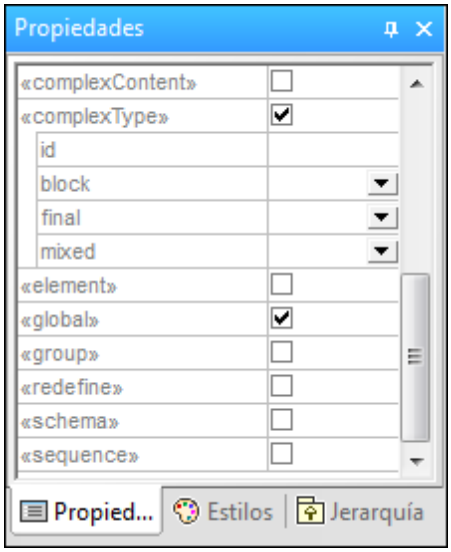
Para definir el tipo de datos de la propiedad:

1. Haga doble clic en la propiedad y ponga el cursor al final de la línea.
2. Inserte dos puntos (:) y seleccione el tipo de datos en el cuadro de diálogo emergente (p. ej. `string`).

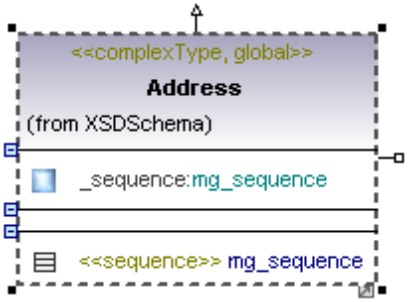
Para crear un modelo de contenido formado por un complexType con elementos obligatorios:

Esto supone insertar un elemento complexType, un elemento `sequence` y tres elementos.

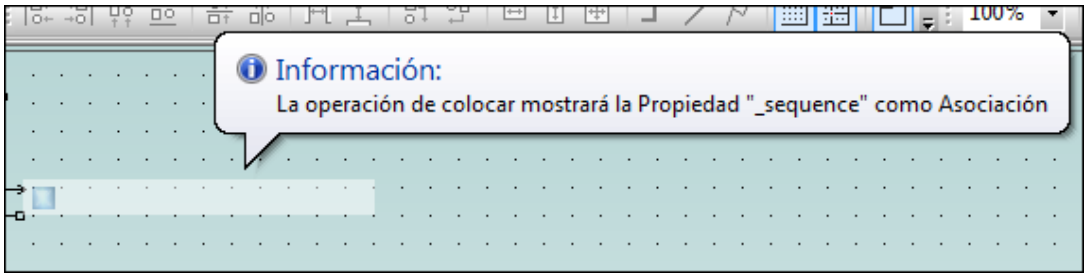
1. Haga clic en el icono **ComplexType XSD** y después haga clic en el diagrama para insertarlo.
2. Haga doble clic en el nombre y ponga el nombre nuevo **`Address`**.



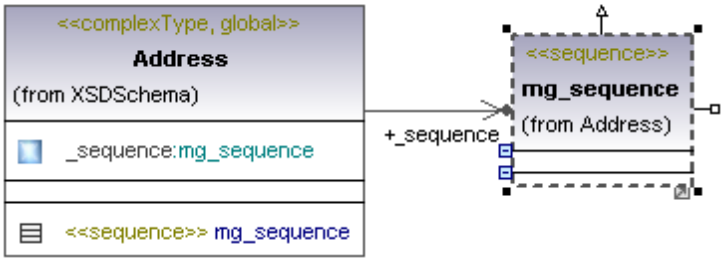
3. Haga clic con el botón derecho en **`Address`** y elija **Nuevo/a | XSD Sequence**.



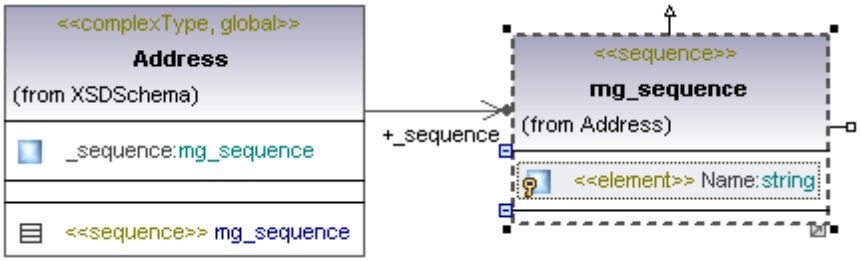
4. Haga clic en el atributo `_sequence:mg_sequence` del compartimento de atributos y arrástrelo fuera del compartimento.



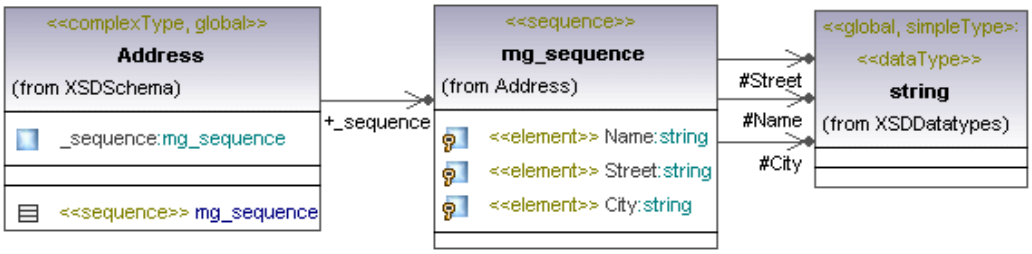
Esto crea una clase `sequence` o `compositor` al soltar el atributo.



5. Haga clic con el botón derecho en la clase **sequence** y elija **Nuevo/a | XSD Element (local)** en el menú contextual.
   Esto añade una propiedad nueva.
6. Haga doble clic en la propiedad, inserte el nombre del elemento (p. ej. **Name**), añada dos puntos (:) e inserte el tipo de datos `string`.



7. Haga lo mismo con otros dos elementos y llámelos `Street` y `City`.
8. Haga clic en la propiedad `Name` y arrástrela hasta el diagrama.

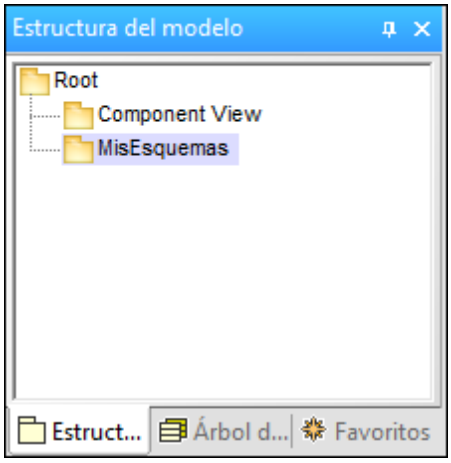


9.3.1.3 *Crear y generar un esquema XML*

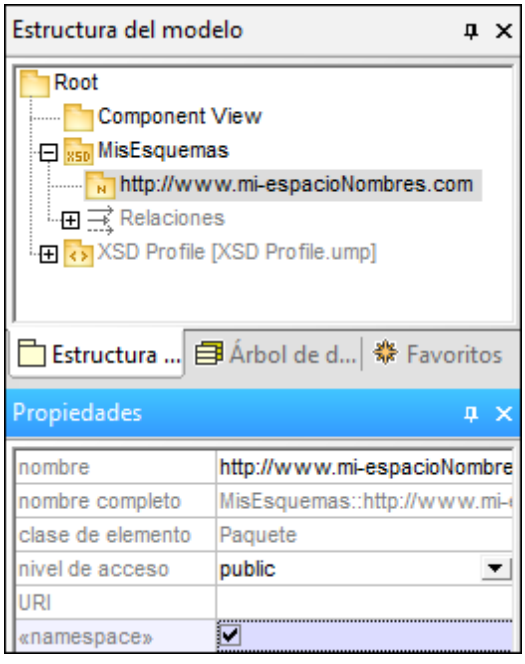
Por lo general, el esquema se importa y se edita en UModel y después se generan los cambios. Sin embargo, en UModel también puede generar un esquema desde cero. A continuación describimos el proceso de forma muy general.

Para crear un esquema nuevo en UModel:

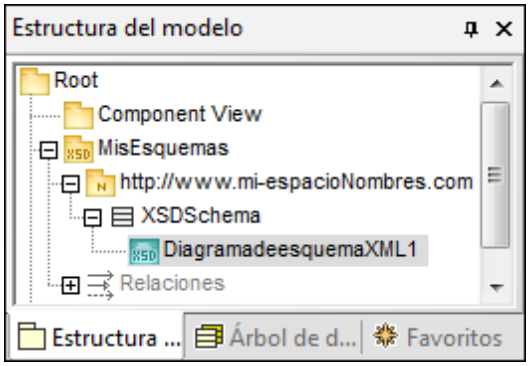
1. En la Estructura del modelo cree un paquete nuevo llamado **MisEsquemas**.



2. Haga clic con el botón derecho en el paquete nuevo y seleccione la opción **Ingeniería de código | Establecer como raíz de espacio de nombres XSD**. Aparece un aviso preguntando si desea asignar el perfil XSD si esta es la primera raíz de espacio de nombres XSD del proyecto.
3. Haga clic en **Aceptar** para asignar el perfil.
4. Haga clic con el botón derecho en el paquete nuevo y seleccione la opción **Elemento nuevo | Paquete**.
5. En el panel Propiedades haga doble clic en el campo `nombre` del paquete e inserte el espacio de nombres que desea usar (p. ej. `http://www.mi-espacioNombres.com`).
6. Marque la casilla `<<namespace>>` para definir este paquete como espacio de nombres de destino.



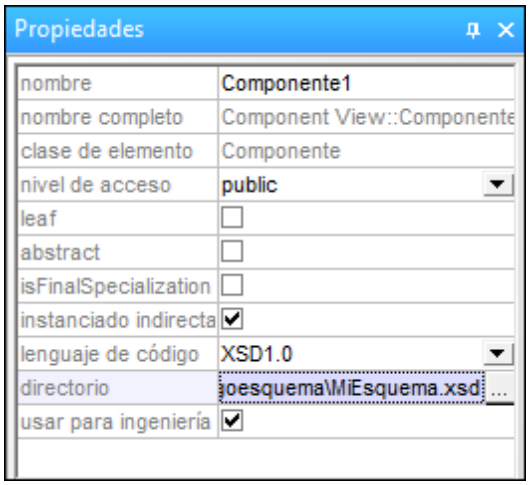
7. Haga clic con el botón derecho en el paquete de espacio de nombres y seleccione la opción **Diagrama nuevo | Diagrama de esquema XML**.
   Aparece un aviso preguntando si desea añadir el diagrama en un esquema XML nuevo.
8. Haga clic en **Sí**.



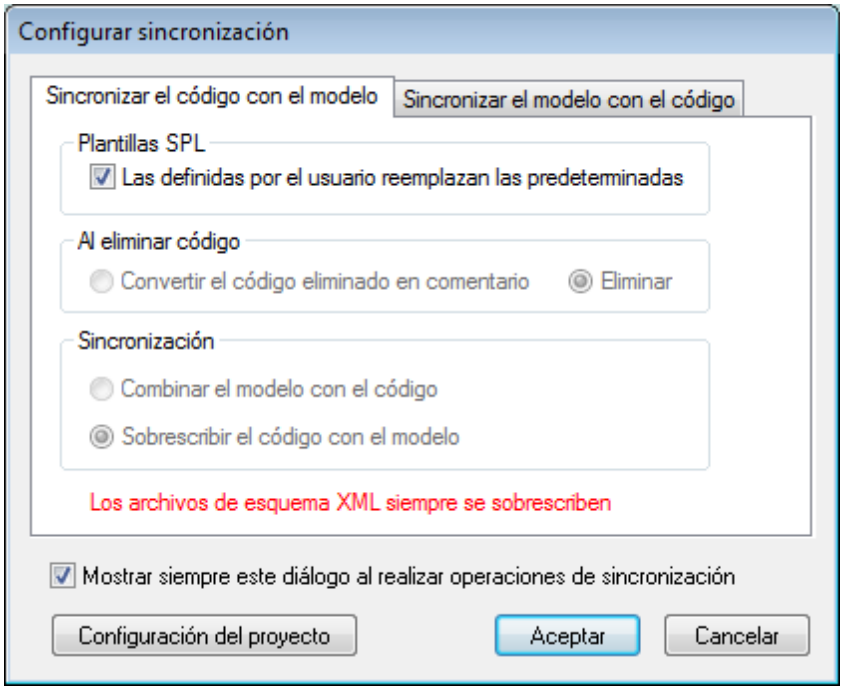
Ahora puede crear su esquema con ayuda de los iconos de la barra de herramientas Diagrama de esquema XML.

Para generar el esquema XML:

1. Arrastre el elemento `XSDSchema` hasta un componente para crear una `RealizaciónDeComponente`.
2. Defina el lenguaje de código XSD 1.0 para el componente y escriba la ruta de acceso donde se debe guardar el esquema generado (p. ej. `C:\códigoesquema\MiEsquema.xsd`).



3. Ahora seleccione el comando de menú **Proyecto | Sobrescribir el código de programa con el proyecto de UModel**. En el cuadro de diálogo haga clic en **Aceptar** para generar el esquema.



Chapter 10

XMI: intercambio de metadatos XML

10 XMI: intercambio de metadatos XML

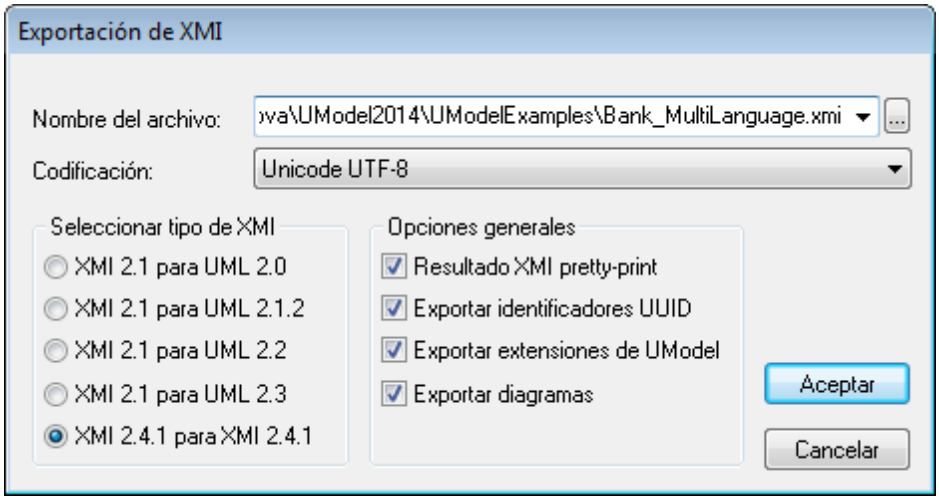
Sitio web de Altova: Intercambio de modelos UML mediante archivos XMI

UModel ofrece una función para exportar e importar datos XMI 2.4 para UML 2.0 / 2.1 / 2.1.1 y 2.1.2, 2.2, 2.3, 2.4. No utilice la función de exportación XMI para archivar proyectos de UModel. Archive los archivos de proyecto `*.ump`.

El comando **Archivo | Exportar a un archivo XMI** genera un archivo XMI a partir del proyecto de UModel. Por su parte, el comando **Archivo | Importar desde un archivo XMI** importa datos desde un archivo XMI generado con anterioridad.

El cuadro de diálogo "Exportación de XMI" sirve para seleccionar el formato XMI en el que se exporta el archivo (p. ej. XMI para UML 2.0/2.1.1). Durante el proceso de exportación también se exportan los archivos incluidos, incluso los que están *incluidos mediante referencia*.

Importante: si tiene pensado reimportar el código XMI al proyecto de UModel, asegúrese de marcar la casilla *Exportar extensiones de UModel*.



Resultado XMI pretty-print

Marque esta casilla para aplicar sangría a las etiquetas XML del archivo XMI de destino e aplicarle los retornos de carro/saltos de línea correspondientes.

Exportar identificadores UUID

El estándar XMI define tres tipos de identificadores de elementos: ID, UUID y etiquetas.

- Los **ID** son identificadores únicos en el documento XMI y son compatibles con la mayoría de las herramientas UML. UModel exporta este tipo de identificadores por defecto (es decir, no hace falta marcar ninguna casilla).
- Los **UUID** son identificadores universales únicos que sirven para asignar a cada elemento una identificación global única. Estos identificadores son únicos a nivel global (es decir, no solo en el documento XMI). Marque la casilla *exportar identificadores UUID* para

generar identificadores UUID.

- Los UUID se almacenan en el formato estándar UUID/GUID (p. ej. "6B29FC40-CA47-1067-B31D-00DD010662DA", "550e8400-e29b-41d4-a716-446655440000",...)
- UModel admite el uso de etiquetas para identificar elementos en XMI.

Nota: el proceso de importación XMI es compatible automáticamente con los identificadores ID y UUID.

Exportar extensiones de UModel

XMI define un mecanismo que permite a cada aplicación exportar sus propias extensiones a la especificación UML. Sin embargo, algunas herramientas UML solo son capaces de importar los datos UML estándar (pasando por alto las extensiones de UModel). Sin embargo, cuando importe datos a UModel estos datos de extensión estarán disponibles.

Algunos datos de UModel, como los nombres de archivo de clases o los colores de los elementos, no forman parte de la especificación UML y, por tanto, deben eliminarse en XMI o guardarse en *extensiones*. Si se exportan como extensiones y se vuelven a importar, estos nombres de archivo y colores se importan tal y como se definieron. Si no usa las extensiones para el proceso de exportación, estos datos de UModel se perderán.

Al importar un documento XMI, UModel detecta el formato y genera un modelo automáticamente.

Exportar diagramas

Marque esta casilla para exportar los diagramas de UModel como *extensiones* en el archivo XMI. Para poder guardar los diagramas como extensiones debe marcar también la casilla *Exportar extensiones de UModel.*

Chapter 11

Trabajo en equipo con UModel

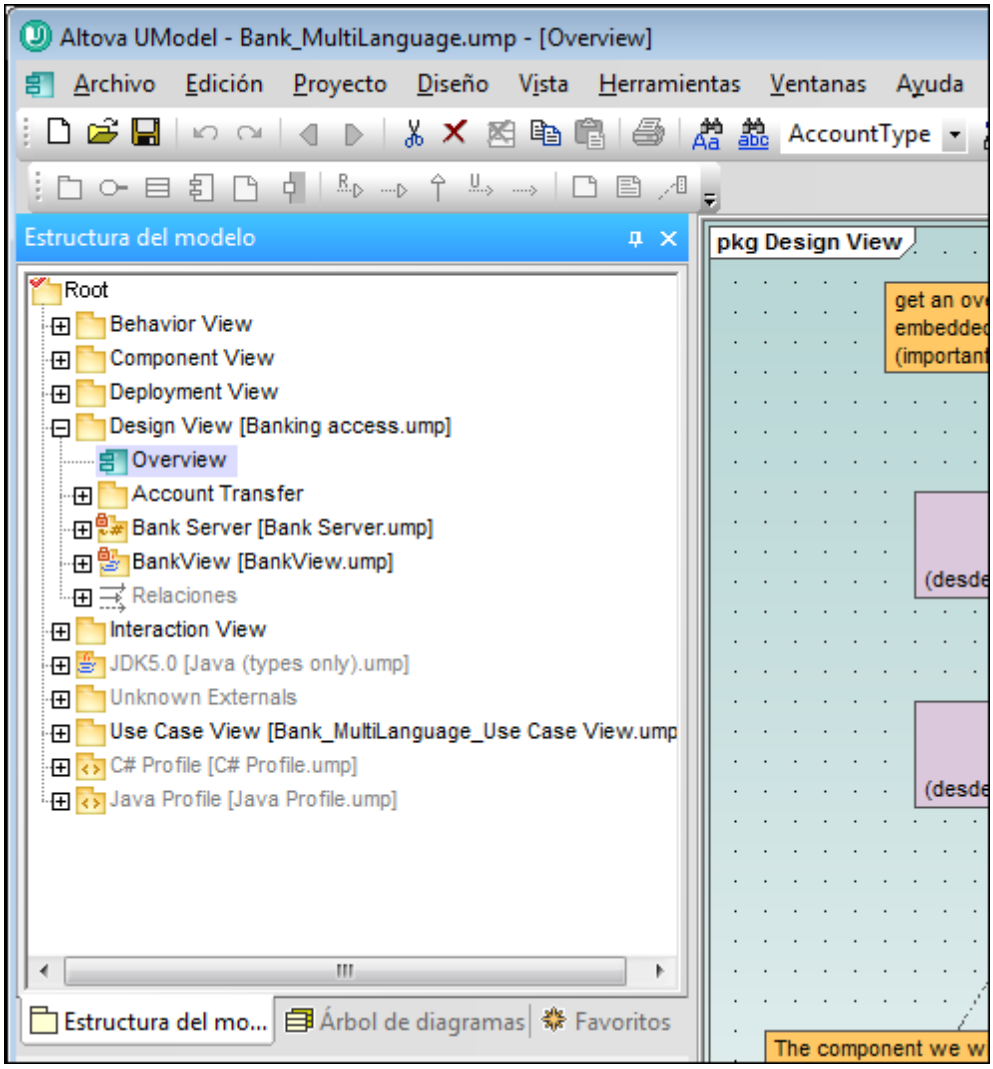
11 Trabajo en equipo con UModel

Los proyectos de UModel se pueden dividir en varios subproyectos para que varios programadores puedan editar partes diferentes del proyecto por separado. Los subproyectos pueden añadirse al sistema de control de versiones utilizado por el equipo.

El proyecto de nivel superior (es decir, el proyecto en el que se pueden incluir los subproyectos) recibe el nombre de *proyecto principal*. Los subproyectos se crean al nivel de los paquetes como archivos de proyecto UModel independientes y tienen la extensión de archivo `*.ump`.

Puede crear e incluir subproyectos de dos maneras distintas:

- un subproyecto puede ser **editable** dentro del proyecto **principal** y a nivel de subproyecto.
- un subproyecto puede ser de **sólo lectura** dentro del proyecto **principal** y editable a nivel de subproyecto.



Los subproyectos pueden estar organizados de forma jerárquica, con una estructura plana o de ambas formas. Esto permite, en teoría, dividir cada paquete de un proyecto principal en varios

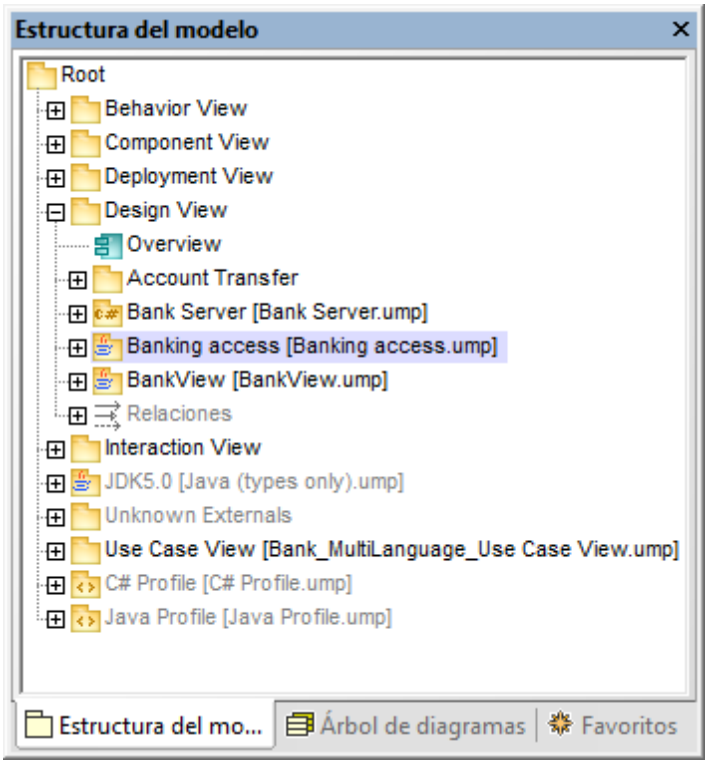
subproyectos editables o de sólo lectura.

Durante el proceso de ingeniería de código todos los componentes subordinados de un subproyecto se tienen en cuenta. Es decir, para la ingeniería de código no hay diferencia alguna entre un archivo de proyecto sencillo y uno formado por varios subproyectos editables.

Esto también afecta a los diagramas UML, que se pueden editar a nivel de proyecto principal o de subproyecto.

11.1 Crear y editar subproyectos

A continuación usamos un ejemplo para explicar cómo se divide un proyecto en varios subproyectos. El ejemplo se sirve del archivo de muestra `Bank_MultiLanguage.ump`.

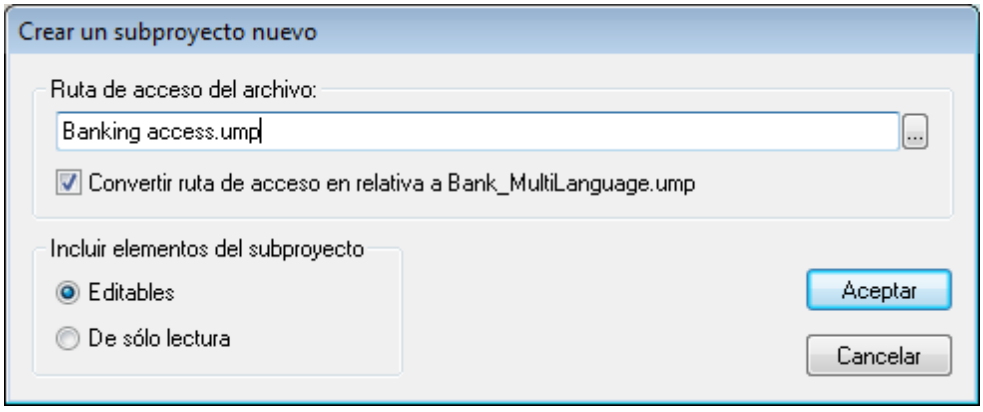


Nota: en UModel puede compartir paquetes y sus diagramas UML con varios proyectos. Los paquetes se pueden incluir en otros proyectos por medio de referencias o como copias. Para más información consulte el apartado Compartir paquetes y diagramas.

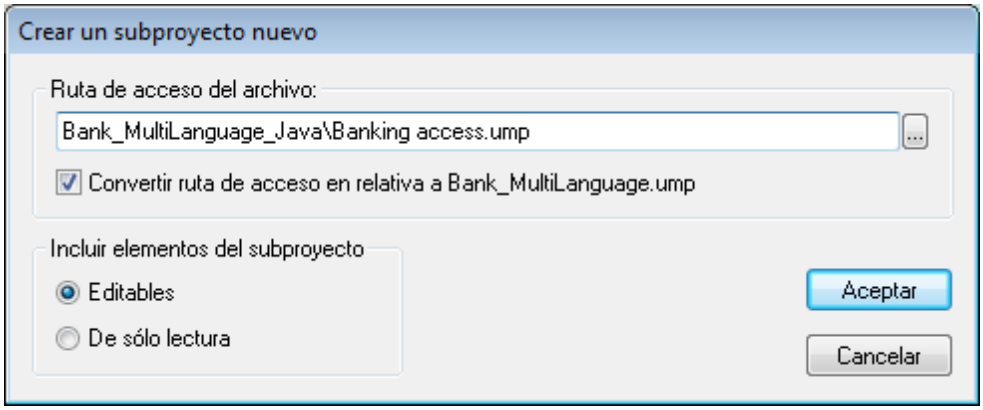
Para crear un archivo de subproyecto:

Puede crear subproyectos a partir de un proyecto principal o de otro subproyecto.

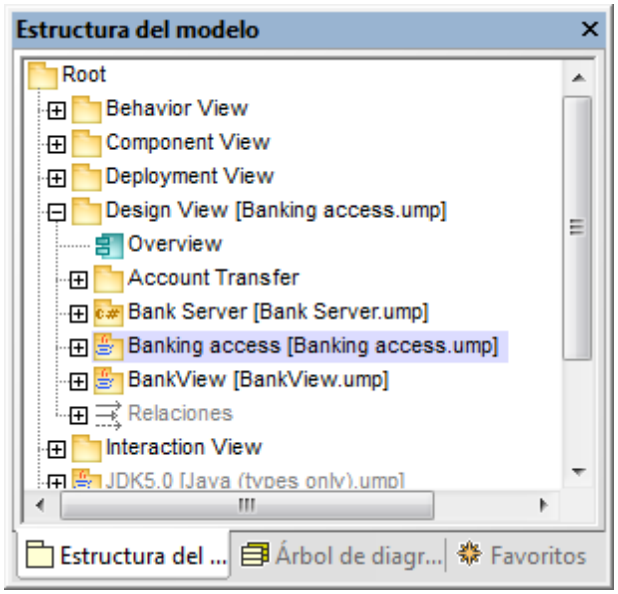
1. Haga clic con el botón derecho en un paquete (p. ej. **`Banking access`**) y seleccione **Subproyecto | Crear subproyecto nuevo**.



2. Haga clic en el botón **Examinar** y seleccione el subdirectorio **`\Bank_MultiLanguage_Java`**.

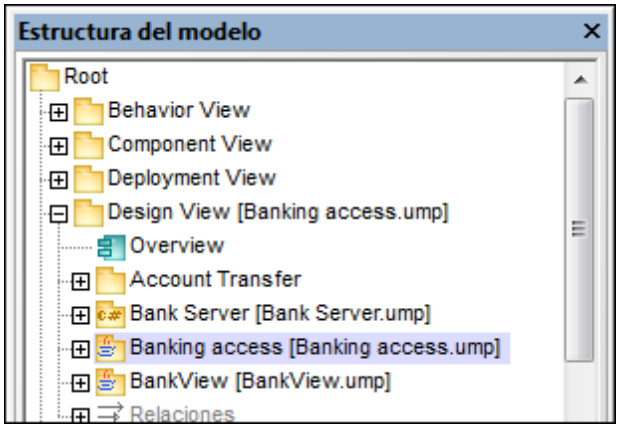


3. Seleccione el botón de opción *Editables* para poder editar el subproyecto desde el proyecto principal. (Si selecciona la opción *De sólo lectura*, el subproyecto no se podrá editar en el proyecto principal). Ahora haga clic en **Aceptar**.



El nombre del subproyecto aparece entre corchetes junto al nombre de paquete y el archivo `Banking access.ump` se guarda en la carpeta `UModelExamples \Bank_MultiLanguage_Java`.

Utilice el mismo método para crear un subproyecto de la carpeta **`BankView`**. El archivo `BankView.ump` se guarda en la carpeta `...\UModelExamples \Bank_MultiLanguage_Java\`.

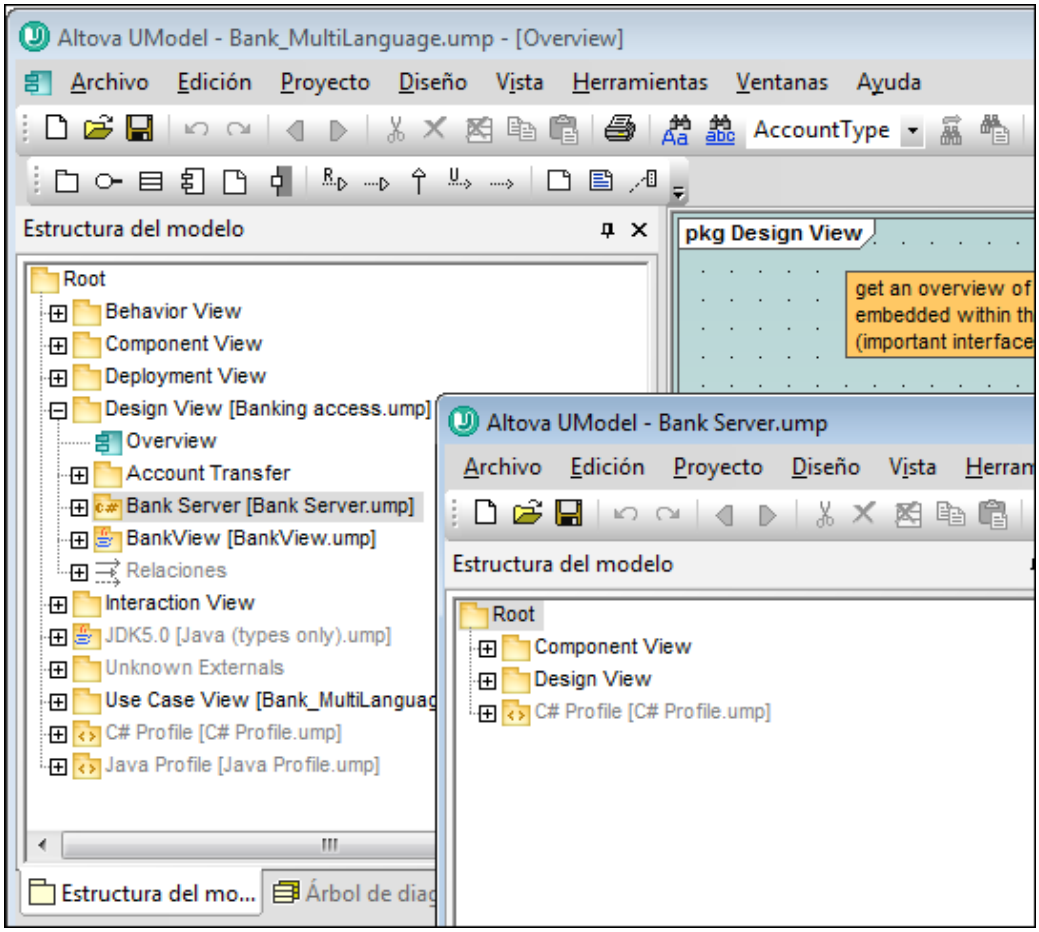


Nota: para cambiar la ruta de acceso del archivo de subproyecto haga clic con el botón derecho en el subproyecto y seleccione **Subproyecto | Editar la ruta de acceso del archivo**.

Para abrir y editar subproyectos:

Una vez creado el subproyecto, el archivo `*.ump` resultante se puede abrir y editar como si fuera un archivo de proyecto principal. Para que esto funcione no puede haber referencias sin resolver a otros elementos. UModel revisa automáticamente el archivo de subproyecto tanto cuando se crea como cuando se guarda.

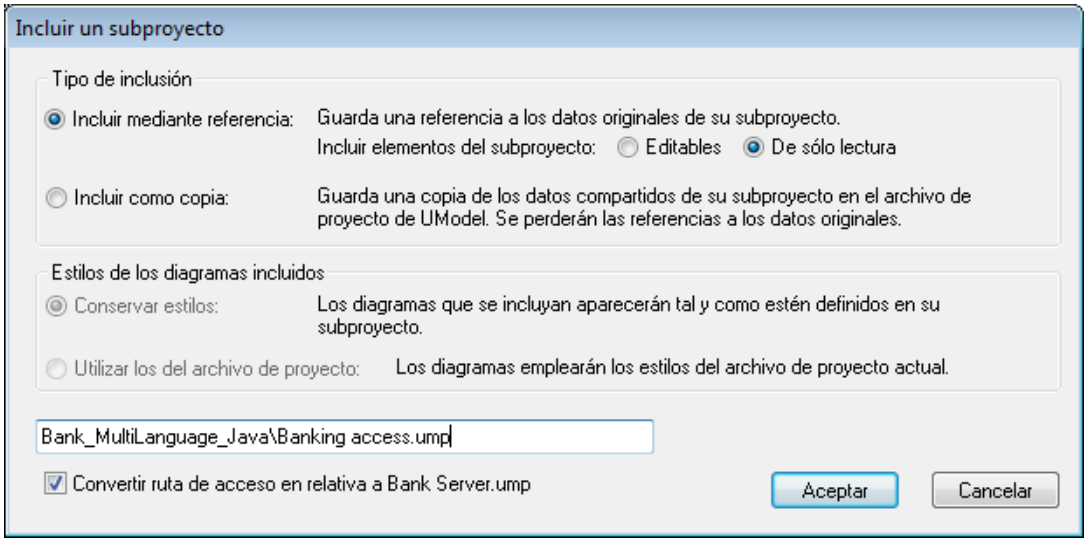
1. Haga clic con el botón derecho en el paquete de subproyecto (p. ej. `Bank Server.ump`) del proyecto principal y seleccione la opción **Subproyecto | Abrir en forma de proyecto**. Esto inicia otra instancia de UModel y abre el subproyecto como si fuera el proyecto principal. Las referencias no resueltas aparecen en la ventana *Mensajes*.



Para reutilizar subproyectos en un otros proyectos:

Los subproyectos en los que se dividió el proyecto principal se pueden usar en otros proyectos principales.

1. Abra el proyecto y después seleccione el comando de menú **Proyecto | Incluir un subproyecto**.
2. Haga clic en el botón **Examinar** y seleccione el archivo `*.ump` que desea incluir (p. ej. `Banking access.ump`).



3. Elija cómo se añade el archivo de subproyecto (mediante referencia o como copia).

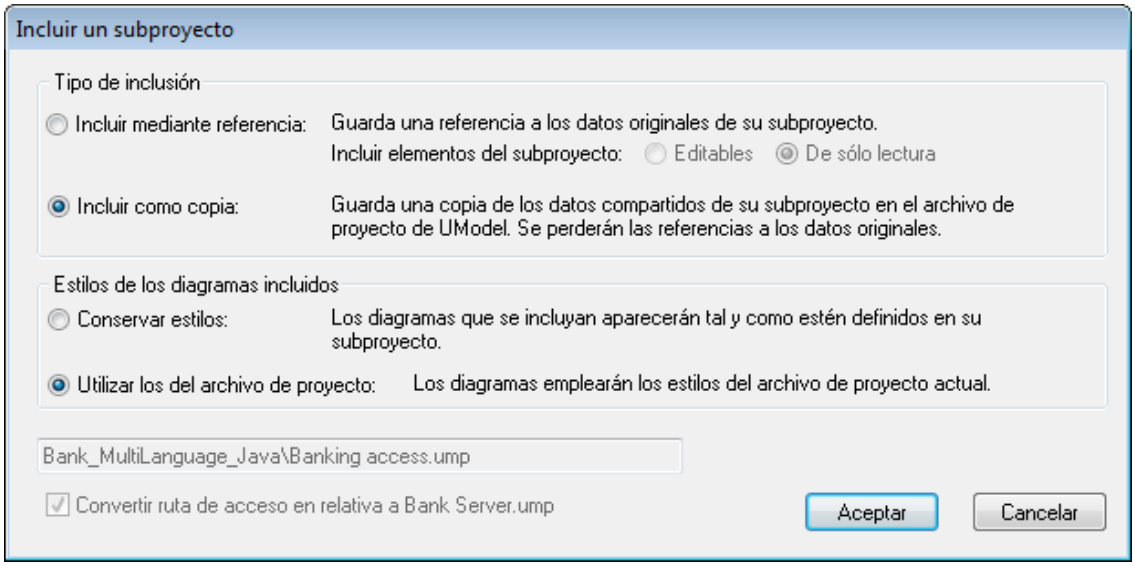
Guardar proyectos

Cuando se guarda el archivo de proyecto principal, también se guardan todos los subproyectos editables (es decir, se guardan todos los datos de los paquetes compartidos de los subproyectos).

Por tanto, no debería crear/agregar datos (componentes) fuera de la estructura compartida/del subproyecto, si el subproyecto se definió como editable en un proyecto principal. Si existen datos fuera de la estructura del subproyecto, UModel emite una advertencia en la ventana *Mensajes*.

Guardar subproyectos

Cuando se guardan subproyectos, se guardan también todas las referencias a subproyectos del mismo nivel y subproyectos secundarios. Por ejemplo, si tenemos los subproyectos del mismo nivel `sub1` y `sub2` y `sub1` utiliza elementos de `sub2`, entonces al guardar `sub1` se guardan automáticamente las referencias a `sub2`.



Si abrimos `sub1` como proyecto principal, se entiende como proyecto autónomo y se puede editar sin referencias al verdadero proyecto principal.

Para volver a integrar los subproyectos en el proyecto principal:

Los subproyectos se pueden volver a copiar dentro del proyecto principal. Si el subproyecto no contiene diagramas, la reintegración es inmediata. Por el contrario, si existen diagramas, UModel muestra un cuadro de diálogo.

1. Haga clic con el botón derecho en el subproyecto y seleccione la opción **Subproyecto | Incluir como copia**.
   Esto abre el cuadro de diálogo "Incluir un subproyecto", donde puede definir qué estilo de diagramas se deben usar al incluir el subproyecto.
2. Por último seleccione la opción de estilo y haga clic en **Aceptar**.

Chapter 12

Control de código fuente

12 Control de código fuente

La función de control de código fuente de UModel funciona con la API del complemento Microsoft Source Control (antes conocido como MSSCCI), versiones 1.1, 1.2 y 1.3. Gracias a esta API podrá ejecutar comandos de control de código fuente como **Proteger** y **Desproteger** desde UModel directamente con prácticamente cualquier control de código fuente que permita la conexión a clientes nativos o de terceros a través de la API del complemento Microsoft Source Control.

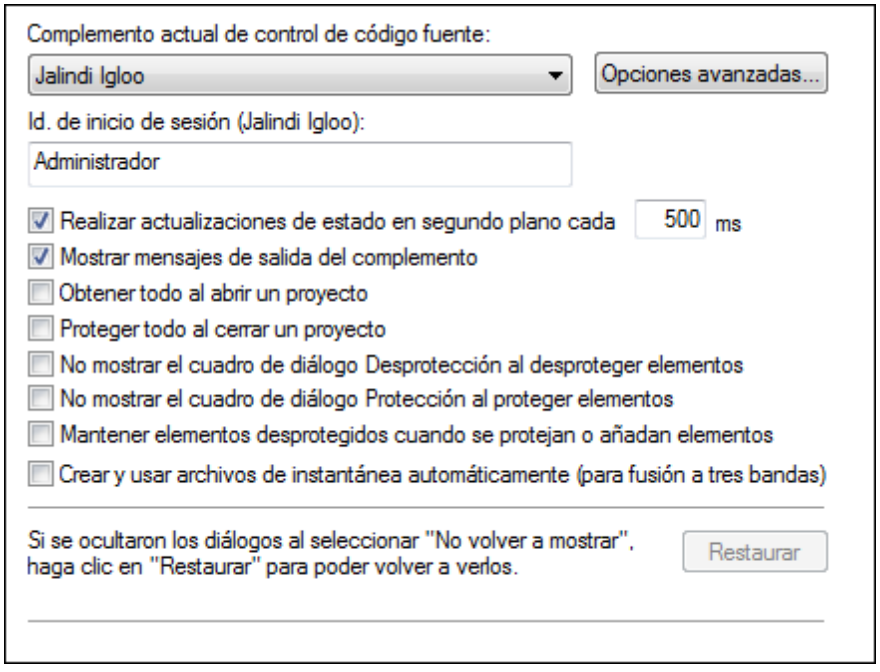
Puede usar cualquier complemento comercial o libre que sea compatible con la API del complemento Microsoft Source Control y puede conectarse a todos los sistemas de control de versiones compatibles (*ver lista de sistemas de control de código fuente compatibles*).

Instalar y configurar el proveedor de control de código fuente

Para ver los proveedores de control de código fuente que están disponibles en el sistema:

1. Haga clic en **Opciones** en el menú **Herramientas**.
2. Haga clic en la pestaña *Control de código fuente*.

Los complementos de control de código fuente que sean compatibles con la API del complemento Microsoft Source Control aparecerán en la lista desplegable *Complemento actual de control de código fuente*.



Si no se encuentra ningún complemento compatible en el sistema, aparece este mensaje:

```
"No se Registration of installed source control providers could not be found or
is incomplete."
```

Algunos sistemas de control de código fuente no instalan el complemento de control de código fuente automáticamente. En este caso deberá instalar el complemento por separado. UModel espera que los complementos compatibles con la API del complemento Microsoft Source Code Control estén instalados bajo esta entrada del registro del sistema operativo:

```
HKEY_LOCAL_MACHINE\SOFTWARE\SourceCodeControlProvider\InstalledSCCProviders
```

Tras la instalación, el complemento aparecerá automáticamente en la lista de complementos disponibles de UModel.

Acceso a los comandos de control de código fuente

Los comandos para trabajar con el control de código fuente están en el menú **Proyecto | Control de código fuente**.

Problemas de rendimiento y recursos

Algunas bases de datos de control de código fuente de gran tamaño pueden crear problemas con recursos y de rendimiento cuando realicen actualizaciones automáticas de estado en segundo plano.

Para aumentar la velocidad del sistema puede deshabilitar (o aumentar el intervalo de) la opción *Realizar actualizaciones de estado en segundo plano cada....segundos* de la pestaña *Control de código fuente* del cuadro de diálogo "Opciones" (**Herramientas | Opciones**).

Nota: la versión de 64 bits de UModel es compatible automáticamente con todos los programas de control de código fuente de 32 bits que se enumeran en esta documentación. Cuando usa una versión de 64 bits de UModel con un programa de control de código fuente de 32 bits, la opción *Realizar actualizaciones de estado en segundo plano cada....segundos* se deshabilita automáticamente y no se puede seleccionar.

Comparación de datos con Altova DiffDog

Muchos sistemas de control de código fuente (como Git y TortoiseSVN) se pueden configurar para usar la herramienta de comparación Altova DiffDog. Para más información consulte la documentación de Altova DiffDog y el sitio web de Altova.

12.1 Sistemas de control de código fuente compatibles

A continuación puede ver una lista con todos los servidores de control de código fuente compatibles con UModel, junto con sus correspondientes clientes de control de código fuente. La lista está ordenada alfabéticamente.

Notas:

- Altova ha implementado la API del complemento Microsoft Source Control (versiones 1.1, 1.2 y 1.3) en UModel y ha probado la compatibilidad con los controladores y sistemas de control de versiones de la lista que aparece a continuación. Altova seguirá ofreciendo compatibilidad con estos productos cuando se actualicen.
- Los clientes de control de código fuente que no aparecen en la lista pero que implementan la API del complemento Microsoft Source Control también deberían funcionar con UModel.

Sistema de control de código fuente	Clientes de control de código fuente
AccuRev 4.7.0 Windows	AccuBridge for Microsoft SCC 2008.2
Bazaar 1.9 Windows	Aigenta Unified SCC 1.0.6
Borland StarTeam 2008	Borland StarTeam Cross-Platform Client 2008 R2
Codice Software Plastic SCM Professional 2.7.127.10 (Server)	Codice Software Plastic SCM Professional 2.7.127.10 (SCC Plugin)
Collabnet Subversion 1.5.4	• Aigenta Unified SCC 1.0.6 • PushOK SVN SCC 1.5.1.1 • PushOK SVN SCC x64 versión 1.6.3.1 • TamTam SVN SCC 1.2.24
ComponentSoftware CS-RCS (PRO) 5.1	ComponentSoftware CS-RCS (PRO) 5.1
Dynamsoft SourceAnywhere for VSS 5.3.2 Standard/Professional Server	Dynamsoft SourceAnywhere for VSS 5.3.2 Client
Dynamsoft SourceAnywhere Hosted	Dynamsoft SourceAnywhere Hosted Client (22252)
Dynamsoft SourceAnywhere Standalone 2.2 Server	Dynamsoft SourceAnywhere Standalone 2.2 Client
Git	PushOK GIT SCC plug-in (consulte Control de código fuente con Git)
IBM Rational ClearCase 7.0.1 (LT)	IBM Rational ClearCase 7.0.1 (LT)
March-Hare CVSNT 2.5 (2.5.03.2382)	Aigenta Unified SCC 1.0.6
March-Hare CVS Suite 2008	• Jalindi Igloo 1.0.3 • March-Hare CVS Suite Client 2008 (3321) • PushOK CVS SCC NT 2.1.2.5 • PushOK CVS SCC x64 versión 2.2.0.4 • TamTam CVS SCC 1.2.40

Sistema de control de código fuente	Clientes de control de código fuente
Mercurial 1.0.2 for Windows	Sergey Antonov HgSCC 1.0.1
Microsoft SourceSafe 2005 with CTP	Microsoft SourceSafe 2005 with CTP
Microsoft Visual Studio Team System 2008/2010 Team Foundation Server	Microsoft Team Foundation Server 2008/2010 MSSCCI Provider
Perforce 2008 P4S 2008.1	Perforce P4V 2008.1
PureCM Server 2008/3a	PureCM Client 2008/3a
QSC Team Coherence Server 7.2.1.35	QSC Team Coherence Client 7.2.1.35
Reliable Software Code Co-Op 5.1a	Reliable Software Code Co-Op 5.1a
Seapine Surround SCM Client/Server for Windows 2009.0.0	Seapine Surround SCM Client 2009.0.0
Serena Dimensions Express/CM 10.1.3 for Win32 Server	Serena Dimensions 10.1.3 for Win32 Client
Softimage Alienbrain Server 8.1.0.7300	Softimage Alienbrain Essentials/Advanced Client 8.1.0.7300
SourceGear Fortress 1.1.4 Server	SourceGear Fortress 1.1.4 Client
SourceGear SourceOffsite Server 4.2.0	SourceGear SourceOffsite Client 4.2.0 (Windows)
SourceGear Vault 4.1.4 Server	SourceGear Vault 4.1.4 Client
VisualSVN Server 1.6	• Aigenta Unified SCC 1.0.6 • PushOK SVN SCC 1.5.1.1 • PushOK SVN SCC x64 versión 1.6.3.1 • TamTam SVN SCC 1.2.24

12.2 Comandos de control de código fuente

En los apartados siguientes utilizamos Visual SourceSafe para explicar las características de control de código fuente de UModel. En todos los ejemplos utilizamos el proyecto de UModel `Bank_CSharp.ump` y sus archivos de código asociados, que están disponibles en la carpeta C:\Documents and Settings\<usuario>\Mis Documentos\Altova\UModel2017\UModelExamples\. No se debe confundir el proyecto de control de código fuente con el proyecto de UModel. Los proyectos de control de código fuente dependen de directorios, mientras que los proyectos de UModel son construcciones lógicas sin dependencias de directorio directas.

Hay varias maneras de acceder a los comandos de control de código fuente:

- Desde el comando de menú **Proyecto | Control de código fuente**.
- Desde el **menú contextual** que aparece al hacer clic con el botón derecho en elementos del panel Estructura del modelo.
- Con los botones de la barra de herramientas Control de código fuente. Para activar esta barra de herramientas haga clic en **Herramientas | Personalizar | Barras de herramientas**.

Estos comandos son los de la edición independiente de UModel. En el complemento de UModel para Visual Studio y Eclipse están disponibles las funciones y comandos de control de código fuente de dichos entornos de desarrollo.

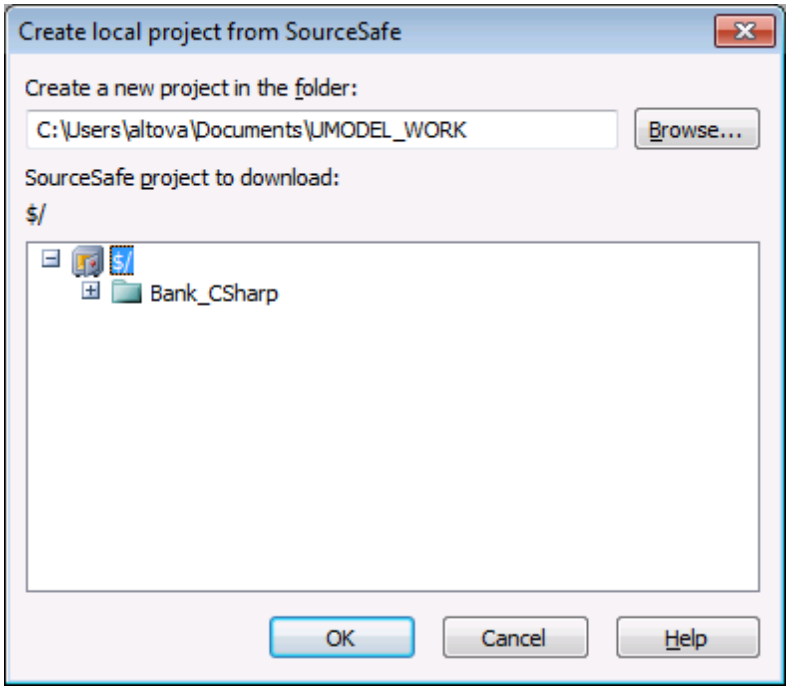
Abrir desde el control de código fuente
Habilitar control de código fuente
Obtener la versión más reciente
Obtener
Obtener carpetas
Desproteger
Proteger
Anular desprotección
Agregar al control de código fuente
Quitar del control de código fuente
Compartir desde el control de código fuente
Mostrar historial
Mostrar diferencias
Mostrar propiedades
Actualizar estado
Administrador del control de código fuente
Cambiar control de código fuente

12.2.1 Abrir desde el control de código fuente

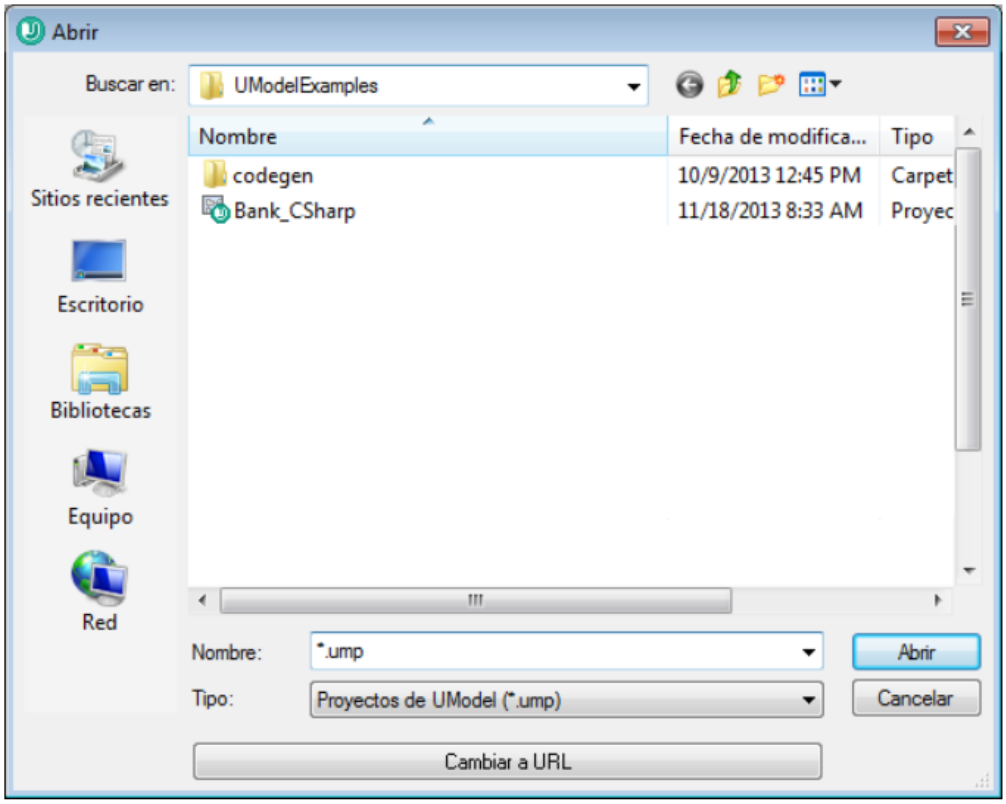
Este comando crea un proyecto local a partir de una BD de control de código fuente ya disponible y lo pone bajo control de código fuente. Para los ejemplos siguientes usamos SourceSafe.

1. Haga clic en **Proyecto | Control de código fuente | Abrir desde el control de código fuente**.
   Se abre el cuadro de diálogo de inicio de sesión. Inserte sus datos para continuar.
   Aparece el cuadro de diálogo "Create local project from SourceSafe".

2. Defina el directorio que debe contener el proyecto local nuevo (p. ej. `c:\temp\ssc`), que en adelante será el directorio de trabajo (o carpeta de desprotección).

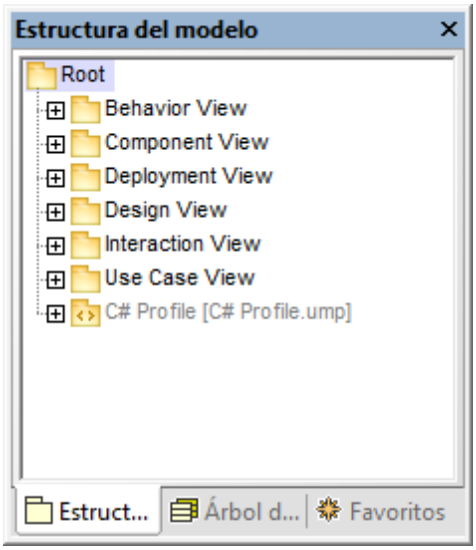


3. Seleccione el proyecto de SourceSafe que desea descargar (p. ej. `Bank_CSharp`). Si la carpeta que define aquí no existe en la ubicación, deberá crearla.
4. Haga clic en **Sí** para crear el directorio nuevo. Ahora aparece el cuadro de diálogo "Abrir".



5. Seleccione el archivo de proyecto de UModel `Bank_CSharp.ump` y haga clic en **Abrir**.

   `Bank_CSharp.ump` se abre en UModel y el archivo se pone bajo control de código fuente. Observe que junto a la carpeta **`Root`** (en la Estructura del modelo) aparece un icono en forma de candado. La carpeta **`Root`** representa tanto el archivo de proyecto como el directorio de trabajo para las operaciones de control de código fuente.



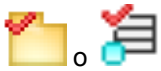
   El directorio `BankCSharp` se creó localmente y ahora puede trabajar con estos archivos de forma totalmente normal.

   Nota: para poner bajo control de código fuente los archivos de código generados cuando se sincronizó el código, consulte el apartado Agregar al control de código fuente.

Iconos del control de código fuente:


o

El icono en forma de candado indica que el archivo / la carpeta está bajo control de código fuente pero no está desprotegido.


o

La marca de verificación roja indica que el archivo / la carpeta se desprotegió para poder editarlo. Si además en la barra de título de la aplicación aparece un asterisco, significa que se realizaron cambios en el archivo y al cerrar la aplicación aparece un aviso para que guarde los cambios.


o

El icono en forma de flecha indica que otro usuario de la red desprotegió el archivo o que se desprotegió en otro directorio de trabajo distinto.

12.2.2 Habilitar control de código fuente

Este comando sirve para habilitar/deshabilitar el control de código fuente para proyectos de UModel y está disponible en el menú **Proyecto | Control de código fuente**. Si se ejecuta desde un archivo/una carpeta, la acción se lleva a cabo en todo el proyecto de UModel.

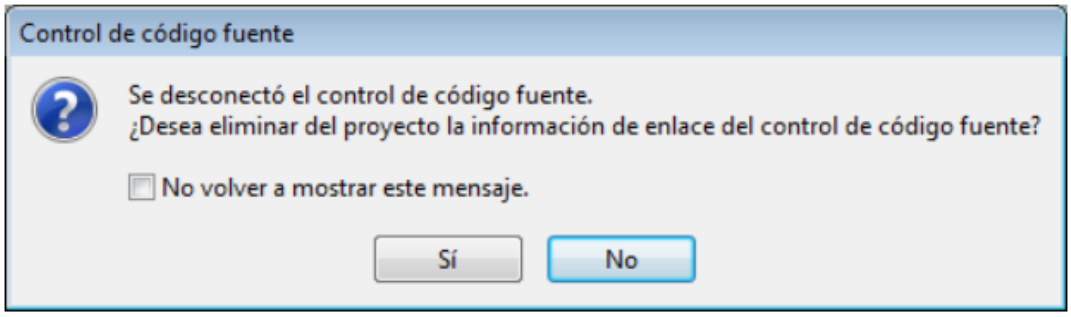
Para habilitar el control de código fuente para un proyecto:

1. Haga clic en **Proyecto | Control de código fuente** y active el comando **Habilitar control de código fuente**.

   La aplicación recupera el estado previo de protección/desprotección de los archivos, lo cual se refleja en la Estructura del modelo.

Para deshabilitar el control de código fuente para un proyecto:

1. Haga clic en el comando **Proyecto | Control de código fuente** y desactive el comando **Habilitar control de código fuente**.



La aplicación pregunta si quiere quitar la información de enlace del proyecto.

Haga clic en **No** para deshabilitar el control de código fuente en el proyecto **de forma provisional**.

Haga clic en **Sí** para deshabilitarlo **permanentemente**.

12.2.3 Obtener la versión más reciente

Este comando **recupera** la versión más reciente del archivo seleccionado del control de código fuente y la **coloca** en el directorio de trabajo. Los archivos se recuperan para solo lectura y no se desprotegen.

Si al ejecutar el comando los archivos están desprotegidos, pueden pasar tres cosas dependiendo del proveedor de control de código fuente: (i) no ocurre nada, (ii) los datos nuevos se combinan con el archivo local (iii) o los cambios se sobrescriben.

Este comando funciona igual que el comando **Obtener**, con la diferencia de que no abre el

cuadro de diálogo "Control de código fuente - Obtener". Esto significa, por tanto, que con este comando no se pueden definir opciones avanzadas.

Si se ejecuta en una carpeta, este comando obtiene la versión más reciente recursivamente, es decir, abarca todos los archivos situados bajo el actual.

Para obtener la versión más reciente de un archivo:

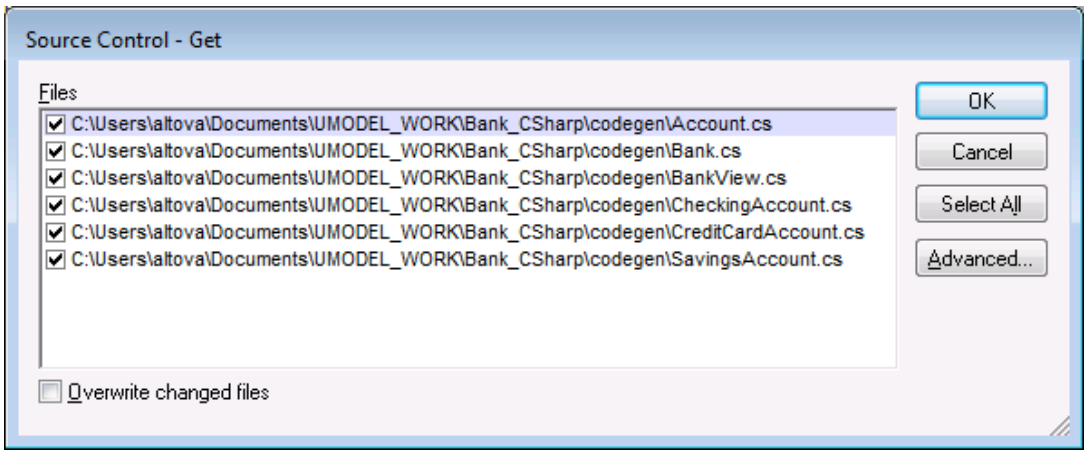
1. En la Estructura del modelo seleccione los archivos cuya versión más reciente desea obtener.
2. Haga clic **Proyecto | Control de código fuente | Obtener la versión más reciente**.

12.2.4 Obtener

Este comando recupera una copia de solo lectura de los archivos seleccionados y los coloca en la carpeta de trabajo. Los archivos no se desprotegen.

Para recuperar una copia de los archivos seleccionados:

1. Seleccione los archivos en la Estructura del modelo.
2. Haga clic en **Proyecto | Control de código fuente | Obtener**.
   Aparece este cuadro de diálogo, cuyas opciones se describen más abajo:

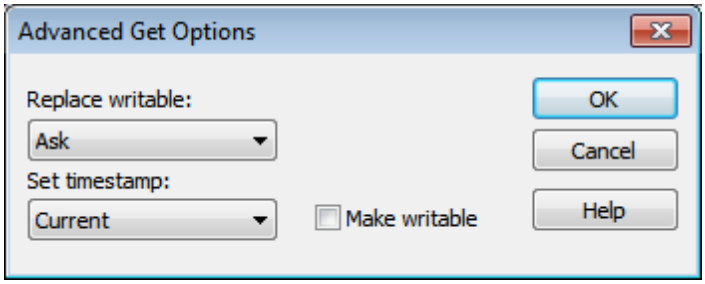


Sobrescribir archivos modificados
Marque esta casilla si quiere sobrescribir los archivos que se modificaron localmente con los archivos de la BD del control de código fuente.

Seleccionar todo
Haga clic en este botón para seleccionar todos los archivos que aparecen en la lista del cuadro de diálogo.

Opciones avanzadas

El cuadro de diálogo "Opciones avanzadas" (*imagen siguiente*) se abre con el botón **Opciones avanzadas** del cuadro de diálogo "Obtener" (*primera imagen de este apartado*).



Aquí puede seleccionar (i) si se reemplazan los archivos que se pueden escribir y que están desprotegidos, (ii) la marca de tiempo y (iii) si la propiedad de solo lectura del archivo recuperado se cambia para que el archivo se pueda escribir.

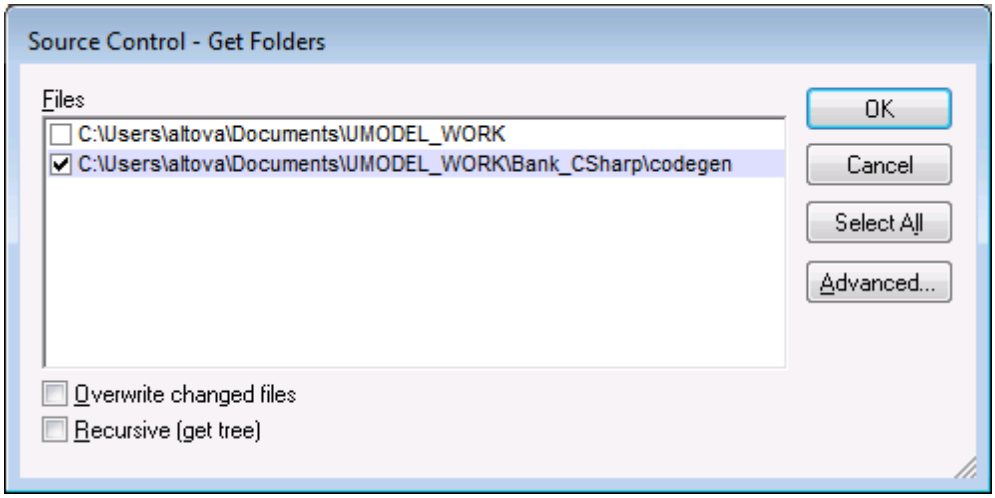
La casilla *Make writable* quita el atributo de solo lectura de los archivos recuperados.

12.2.5 Obtener carpetas

Este comando recupera una copia de solo lectura de los archivos de las carpetas seleccionadas y las coloca en la carpeta de trabajo. Los archivos no se desprotegen.

Para obtener una copia de los archivos de las carpetas seleccionadas:

1. En la Estructura del modelo seleccione qué carpetas desea obtener.
2. Haga clic en **Proyecto | Control de código fuente | Obtener**.
   Aparece este cuadro de diálogo, cuyas opciones se describen más abajo:



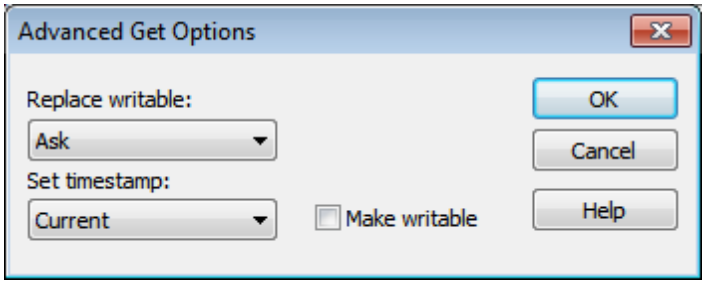
Sobrescribir archivos modificados

Marque esta casilla para sobrescribir los archivos que se modificaron localmente con los archivos de la BD del control de código fuente.

Jerarquía recursiva (obtener árbol)
Marque esta casilla para recuperar todos los archivos de la estructura de carpetas situada bajo la carpeta seleccionada.

Opciones avanzadas

El cuadro de diálogo "Opciones avanzadas" (*imagen siguiente*) se abre con el botón **Opciones avanzadas** del cuadro de diálogo "Obtener" (*primera imagen de este apartado*).



Aquí puede seleccionar (i) si se reemplazan los archivos que se pueden escribir y que están desprotegidos, (ii) la marca de tiempo y (iii) si la propiedad de solo lectura del archivo recuperado se cambia para que el archivo se pueda escribir.

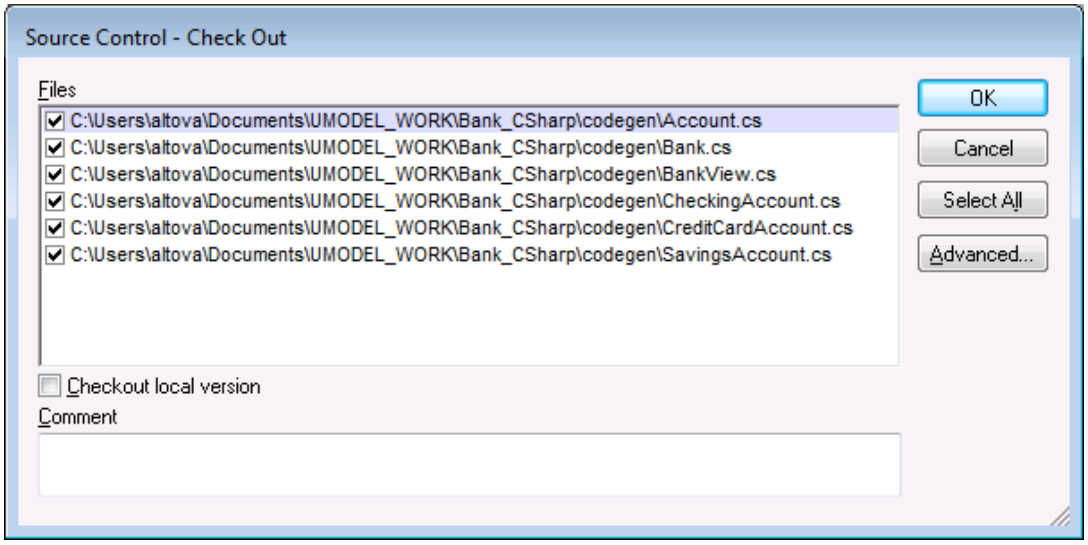
La casilla *Make writable* quita el atributo de solo lectura de los archivos recuperados.

12.2.6 Desproteger

Este comando desprotege la versión más reciente de los archivos seleccionados y coloca una copia editable en el directorio de trabajo. Los otros usuarios ven un icono de "desprotegido" en los archivos desprotegidos.

Para desproteger archivos:

1. En la Estructura del modelo seleccione el archivo o la carpeta que desea desproteger.
2. Haga clic en **Proyecto | Control de código fuente | Desproteger**.
   Aparece un cuadro de diálogo (*imagen siguiente*) donde puede seleccionar qué archivos se desprotegen finalmente (marcando sus casillas). Más abajo se describen las opciones de este cuadro de diálogo.



Desproteger versión local
Marque esta casilla para desproteger solamente las versiones locales de los archivos y no las versiones que están en la BD del control de código fuente.

Estos son los elementos que se pueden desproteger:

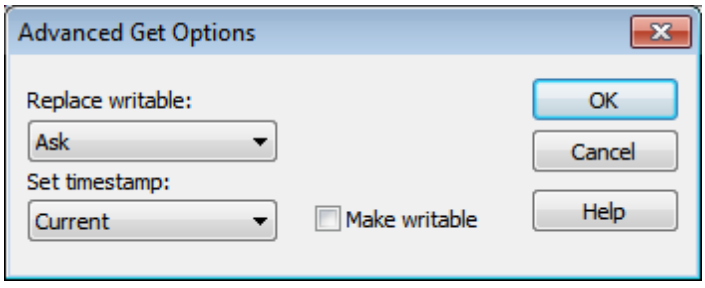
- Archivos (para seleccionar varios, pulse la tecla **Ctrl** mientras hace clic en los archivos en la Estructura del modelo).
- Carpetas (para seleccionar varias, pulse la tecla **Ctrl** mientras hace clic en las carpetas en la Estructura del modelo).

 o

La marca de verificación **roja** indica que el archivo / la carpeta **se desprotegió** para poder editarlo.

Opciones avanzadas

El cuadro de diálogo "Opciones avanzadas" (*imagen siguiente*) se abre con el botón **Opciones avanzadas** del cuadro de diálogo "Obtener" (*primera imagen de este apartado*).



Aquí puede seleccionar (i) si se reemplazan los archivos que se pueden escribir y que están desprotegidos, (ii) la marca de tiempo y (iii) si la propiedad de solo lectura del archivo recuperado se cambia para que el archivo se pueda escribir.

La casilla *Make writable* quita el atributo de solo lectura de los archivos recuperados.

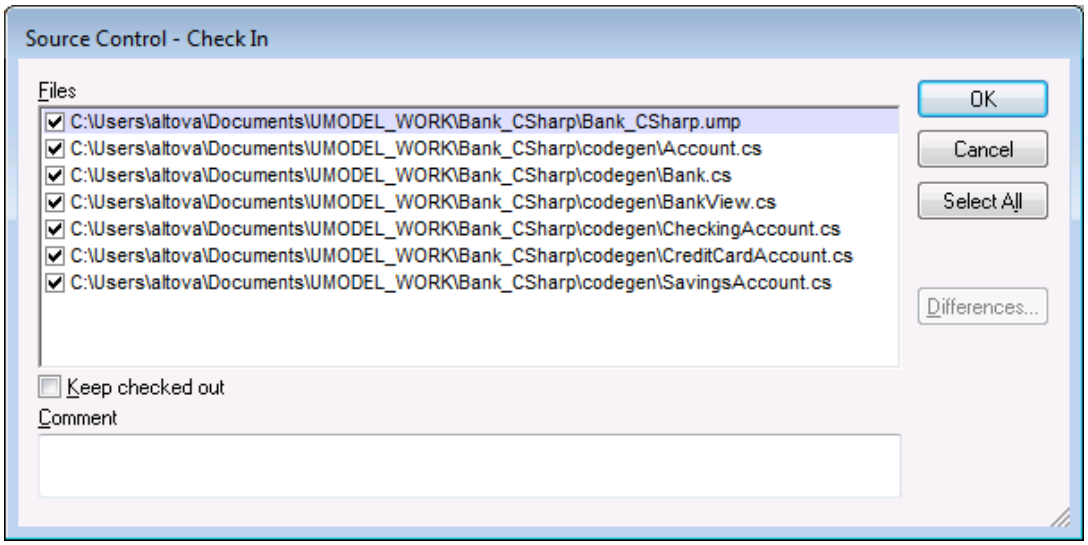
12.2.7 Proteger

Este comando protege los archivos que estén desprotegidos (es decir, los archivos que se modificaron localmente) y los pone en la BD del control de código fuente.

Para proteger archivos:

1. En la Estructura del modelo seleccione los archivos que quiere proteger.
2. Haga clic en **Proyecto | Control de código fuente | Proteger**.

   Nota: también puede hacer clic con el botón derecho en los archivos que quiere proteger y seleccionar **Proteger** en el menú contextual.



Estos son los elementos que se pueden proteger:

- Archivos (para seleccionar varios, pulse la tecla **Ctrl** mientras hace clic en los archivos en la Estructura del modelo).
- Carpetas (para seleccionar varias, pulse la tecla **Ctrl** mientras hace clic en las carpetas en la Estructura del modelo).

 o

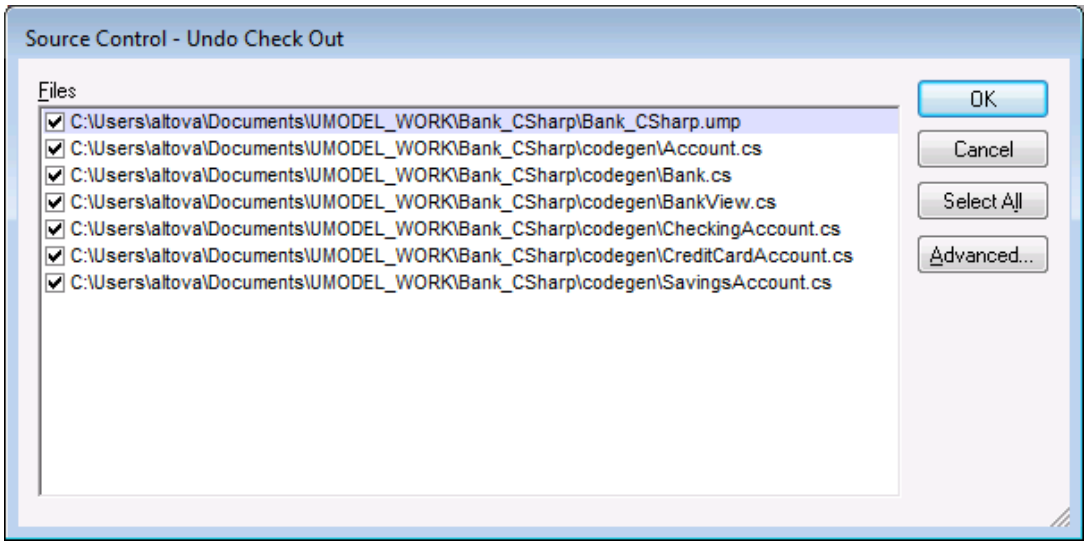
El icono en forma de candado indica que el archivo / la carpeta está **bajo control de código fuente** pero no está desprotegido.

12.2.8 Anular desprotección

Este comando descarta los cambios realizados en archivos desprotegidos (es decir, los archivos que se modificaron localmente) y mantiene la versión previa de la BD del control de código fuente.

Para anular la desprotección:

1. Seleccione los archivos correspondientes en la Estructura del modelo.
2. Haga clic en **Proyecto | Control de código fuente | Anular desprotección**. Aparece un cuadro de diálogo (*imagen siguiente*) donde puede seleccionar para qué archivos se anula la desprotección finalmente (marcando sus casillas).

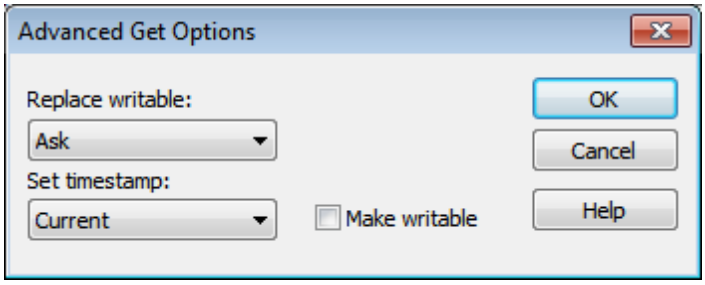


Estos son los elementos cuya desprotección se puede anular:

- Archivos (para seleccionar varios, pulse la tecla **Ctrl** mientras hace clic en los archivos en la Estructura del modelo).
- Carpetas (para seleccionar varias, pulse la tecla **Ctrl** mientras hace clic en las carpetas en la Estructura del modelo).

Opciones avanzadas

El cuadro de diálogo "Opciones avanzadas" (*imagen siguiente*) se abre con el botón **Opciones avanzadas** del cuadro de diálogo "Obtener" (*primera imagen de este apartado*).



Aquí puede seleccionar (i) si se reemplazan los archivos que se pueden escribir y que están desprotegidos, (ii) la marca de tiempo y (iii) si la propiedad de solo lectura del archivo recuperado se cambia para que el archivo se pueda escribir.

La casilla *Make writable* quita el atributo de solo lectura de los archivos recuperados.

12.2.9 Agregar al control de código fuente

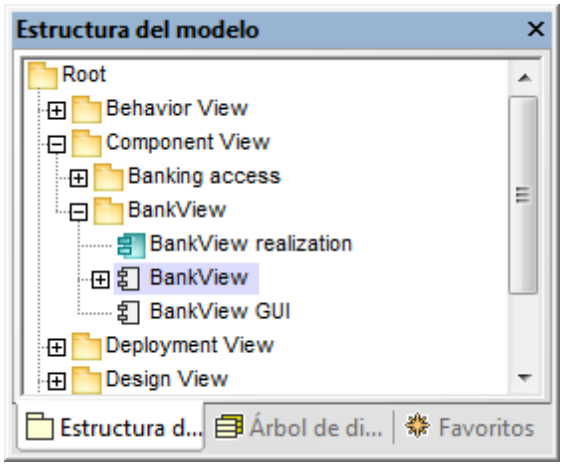
Este comando añade los archivos o las carpetas seleccionados a la BD del control de código fuente y los pone bajo su control. Si añade un proyecto de UModel nuevo, deberá indicar la carpeta del espacio de trabajo y la ubicación donde se debe almacenar el proyecto.

Tras poner el proyecto de UModel (`*.ump`) bajo control de código fuente, puede añadir al control de código fuente los **archivos de código** (creados durante el proceso de generación de código).

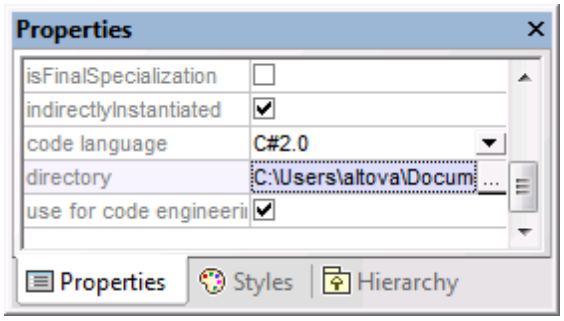
Es importante tener en cuenta que para que esto funcione los archivos de código generados y el proyecto de UModel deben ponerse dentro/bajo el mismo **directorio de trabajo** de SourceSafe. El directorio de trabajo que utilizamos para este ejemplo es `C:\Users\Altova\Documents\UMODEL_WORK\`.

Para agregar archivos de código generados con UModel al control de código fuente:

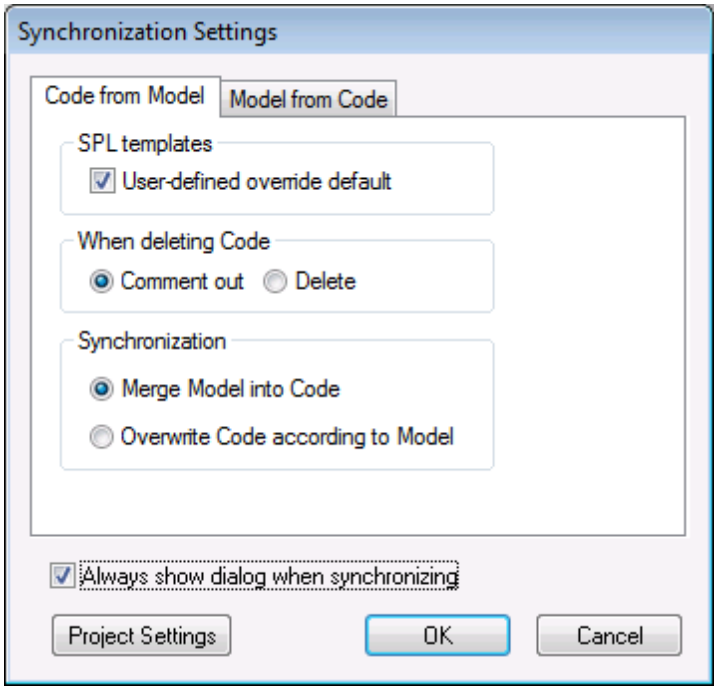
1. En la Estructura del modelo expanda la carpeta **`Component View`** y navegue hasta el componente **`BankView`**.



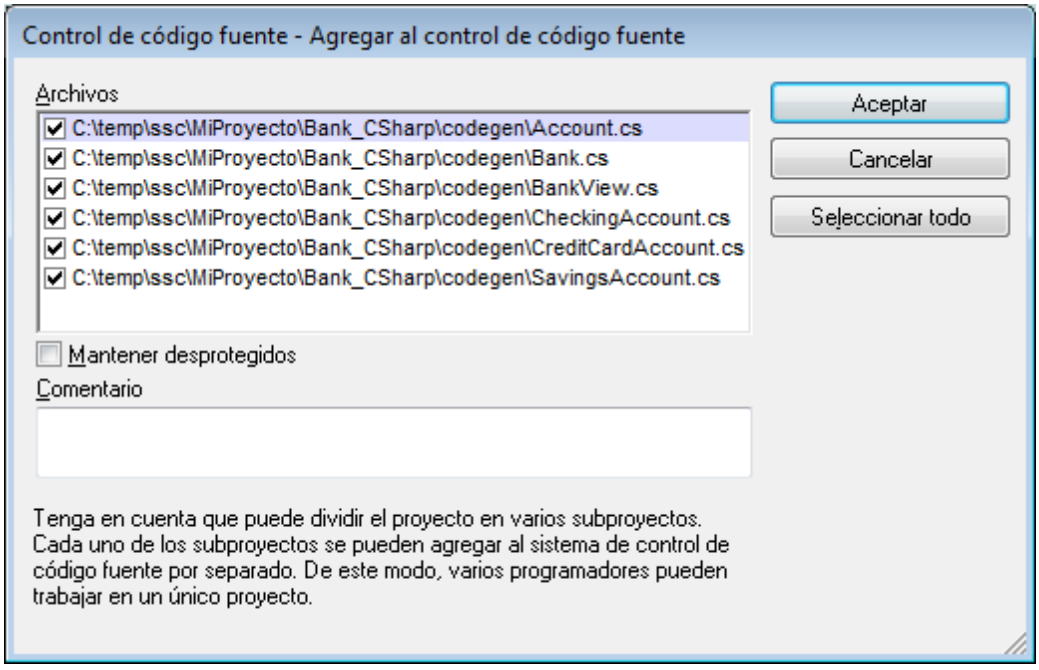
2. Haga clic en el componente **`BankView`**. En la ventana Propiedades haga clic en el icono **Examinar** del campo `directorio`.



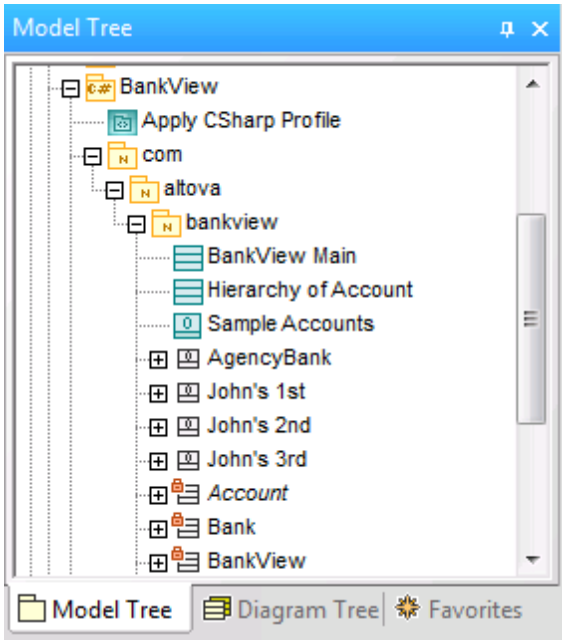
3. Cambie el directorio de ingeniería de código a `C:\Users\Altova\Documents\UMODEL_WORK\codegen`.
4. Ahora haga clic en **Proyecto | Combinar el código de programa con el proyecto de UModel**.
5. Si es necesario, cambie las opciones de configuración y haga clic en **Aceptar**.
   La ventana Mensajes muestra cómo se desarrolla el proceso.
   Aparece un cuadro de mensajes que pregunta si desea poner los archivos recién creados bajo control de código fuente.



6. Haga clic en **Sí**.
7. Aparece el cuadro de diálogo "Agregar al control de código fuente" donde puede seleccionar qué archivos se ponen bajo control de código fuente.



8. Seleccione los archivos y haga clic en **Aceptar**.
   Observe que junto a las clases que ahora están bajo control de código fuente aparece un icono en forma de candado.



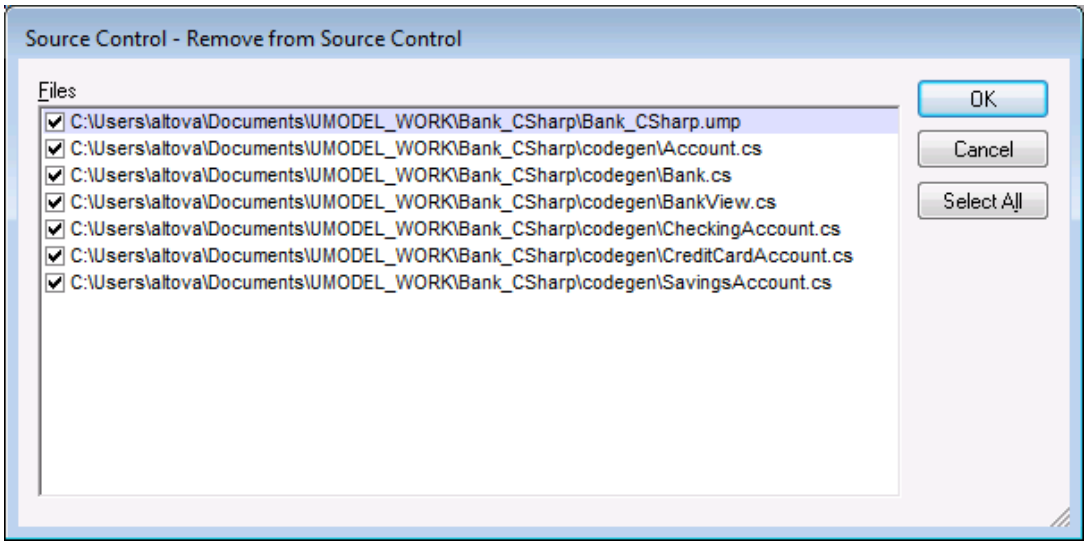
12.2.10 Quitar del control de código fuente

Este comando quita de la BD del control de código fuente los archivos añadidos previamente a la BD. Este tipo de archivos siguen estando visibles en la Estructura del modelo pero no se pueden proteger ni desproteger. Para ponerlos otra vez bajo control de código fuente, utilice el comando

Agregar al control de código fuente.

Para quitar archivos del control de código fuente:

1. Seleccione los archivos en la Estructura del modelo.
2. Haga clic en **Proyecto | Control de código fuente | Quitar del control de código fuente**.
   Aparece un cuadro de diálogo (*imagen siguiente*) donde puede seleccionar qué archivos se quitan del control de código fuente finalmente (marcando sus casillas).



Estos son los elementos que se pueden quitar del control de código fuente:

- Archivos (para seleccionar varios, pulse la tecla **Ctrl** mientras hace clic en los archivos en la Estructura del modelo).
- Carpetas (para seleccionar varias, pulse la tecla **Ctrl** mientras hace clic en las carpetas en la Estructura del modelo).

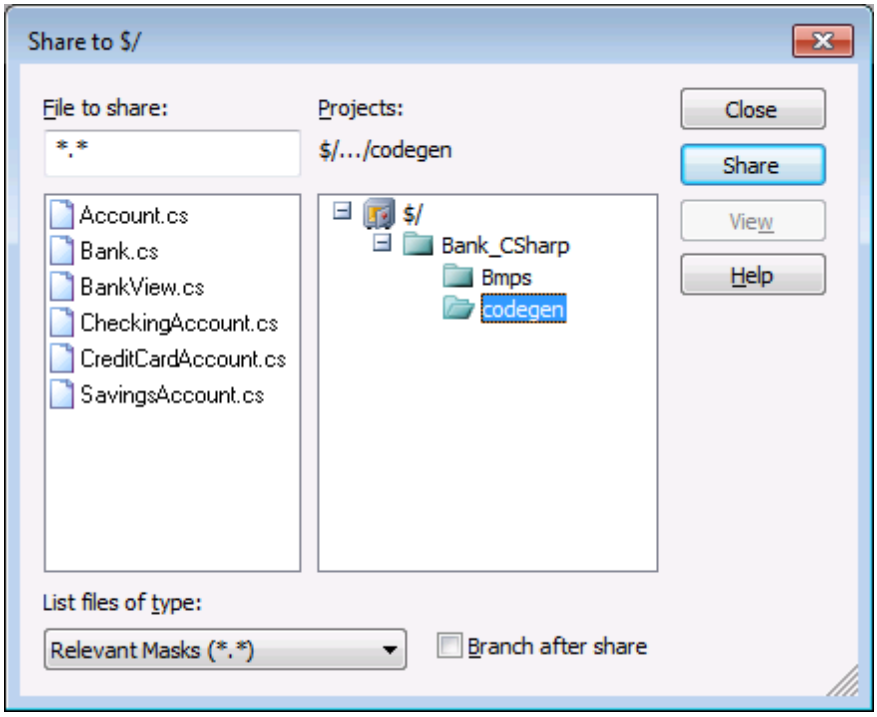
12.2.11 Compartir desde el control de código fuente

Este comando comparte/ramifica archivos de otros proyectos/carpetas del repositorio de control de código fuente con la carpeta seleccionada. Para usar este comando es necesario tener privilegios para proteger/desproteger datos en el proyecto desde el que se comparten los archivos.

Para compartir un archivo desde el control de código fuente:

1. En la Estructura del modelo seleccione la carpeta con la que quiere compartir archivos (p. ej. `BankView Component` de la carpeta `Component View`).
2. Haga clic en **Proyecto | Control de código fuente | Compartir desde el control de código fuente**.
   Aparece un cuadro de diálogo (*imagen siguiente*) donde puede seleccionar qué carpeta

de proyecto contiene el archivo que desea compartir.



3. Seleccione el archivo que desea compartir y haga clic en el botón **Share**.
   El archivo se elimina de la lista *File to share*.
4. Haga clic en el botón **Close** para continuar.

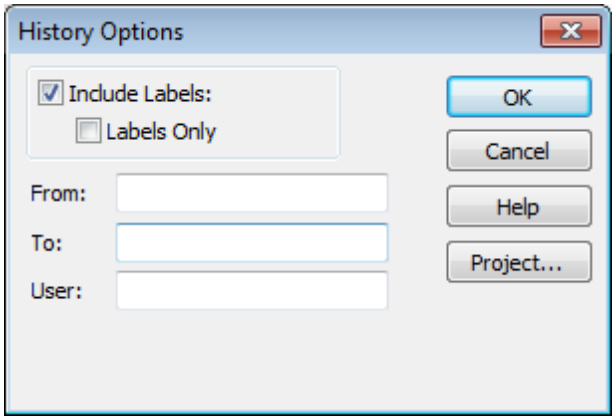
   Nota: marque la casilla *Branch after share* para compartir el archivo y crear una rama nueva para crear una nueva versión.

12.2.12 Mostrar historial

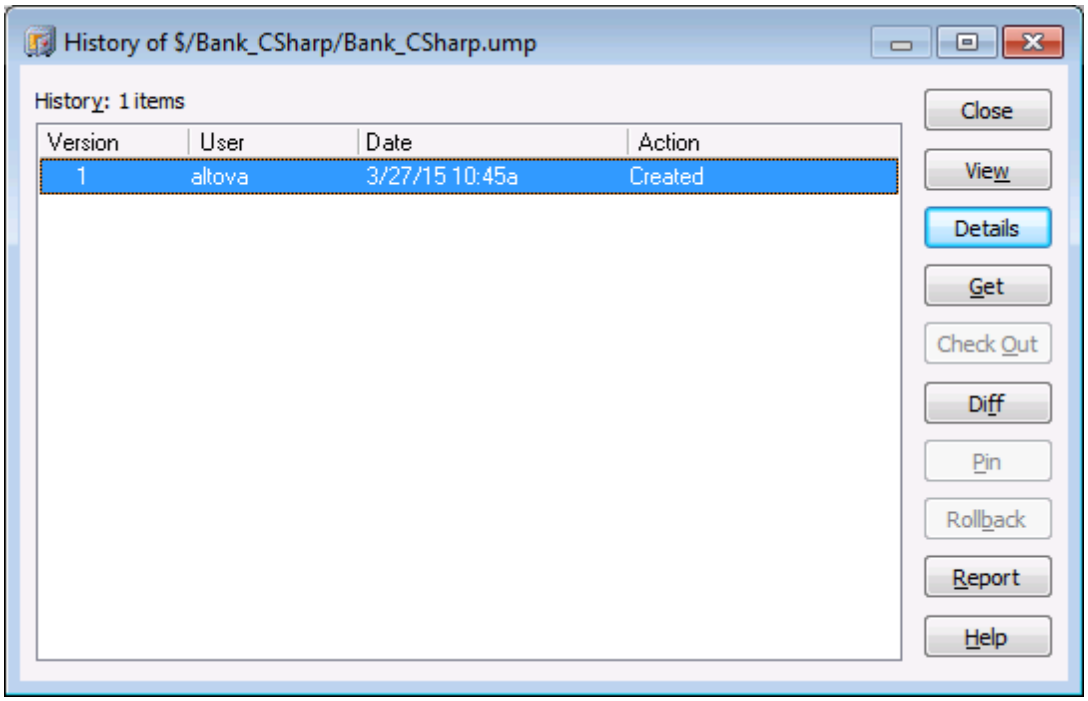
Este comando **muestra el historial** de un archivo que está bajo control de código fuente. El historial permite ver la historia de cambios por la que ha pasado el archivo, ver diferencias entre sus diferentes versiones y recuperar versiones previas.

Para ver el historial de un archivo:

1. Seleccione el archivo en la Estructura del modelo.
2. Haga clic en el comando **Proyecto | Control de código fuente | Mostrar historial**.
   Aparece un cuadro de diálogo pidiendo más información.



3. Seleccione las entradas correspondientes y haga clic en **OK** para confirmar.



En este cuadro de diálogo puede comparar versiones del archivo y obtener versiones previas.

Para ver el historial detallado haga doble clic en una entrada de la lista.

Estos son los botones del cuadro de diálogo del historial:

Close
Cierra el cuadro de diálogo.

View
Abre otro cuadro de diálogo donde puede seleccionar en qué aplicación desea ver el archivo.

Details
Abre un cuadro de diálogo que muestra las propiedades del archivo seleccionado.

Get
Recupera una de las versiones previas del archivo y la coloca en el directorio de trabajo.

Check Out
Desprotege la versión **más reciente** del archivo.

Diff
Abre el cuadro de diálogo "Difference options" donde puede configurar la vista de las diferencias detectadas entre las dos versiones del archivo.
Para marcar dos versiones de un archivo pulse la tecla **Ctrl** mientras hace clic en las entradas. Después pulse el botón **Diff** para ver las diferencias.

Pin
Ancla/desancla una versión del archivo para que pueda definir la versión que se debe usar en la comparación de dos archivos.

Rollback
Revierte a la versión seleccionada del archivo.

Report
Genera un historial que puede imprimirse, guardarse en un archivo o copiarse en el portapapeles.

Help
Abre la ayuda en pantalla del cliente de control de código fuente.

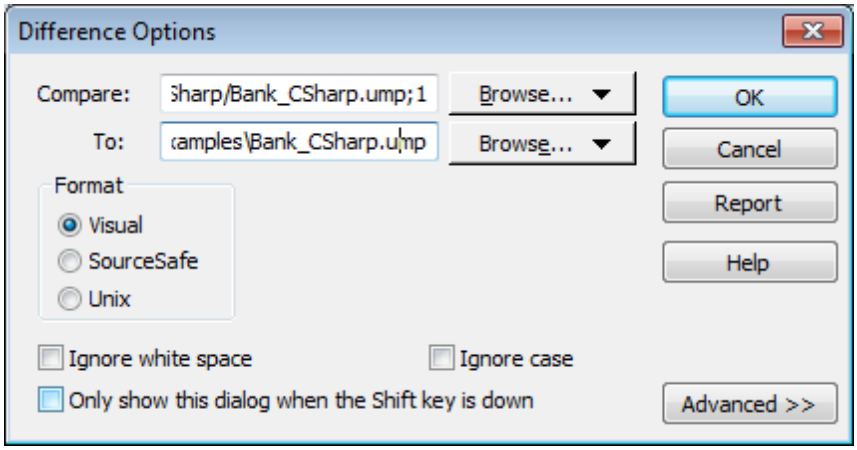
12.2.13 Mostrar diferencias

Este comando muestra las diferencias que existen entre el archivo que está en el repositorio del control de código fuente y el mismo archivo protegido/desprotegido del directorio de trabajo.

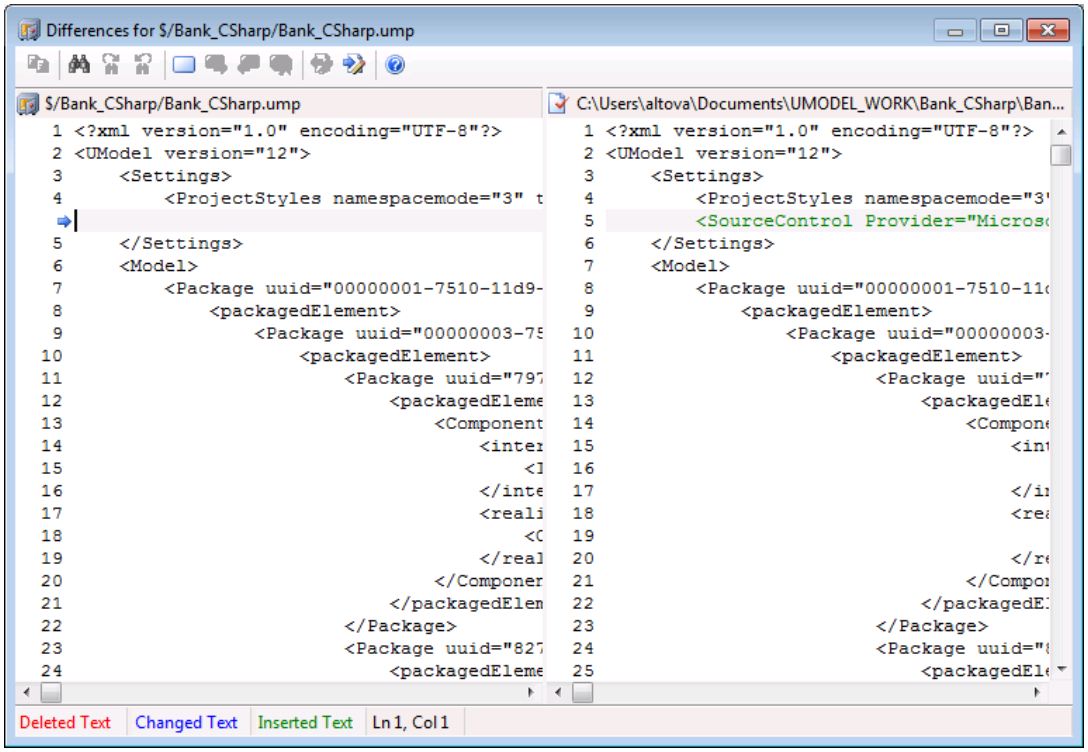
Si ancló uno de los archivos en el cuadro de diálogo del historial, el archivo anclado se inserta automáticamente en el campo de texto *Compare*. Con los botones **Browse** puede buscar los archivos que desea comparar.

Para ver las diferencias que hay entre dos archivos:

1. En la Estructura del modelo seleccione el archivo que desea comparar.
2. Haga clic en **Proyecto | Control de código fuente | Mostrar diferencias**.
   Aparece un cuadro de diálogo que le pide más información.



3. Seleccione las entradas correspondientes y haga clic en OK para **confirmar**.



Las diferencias detectadas aparecen resaltadas. Por ejemplo, la imagen anterior muestra los resultados de la comparación en MS SourceSafe.

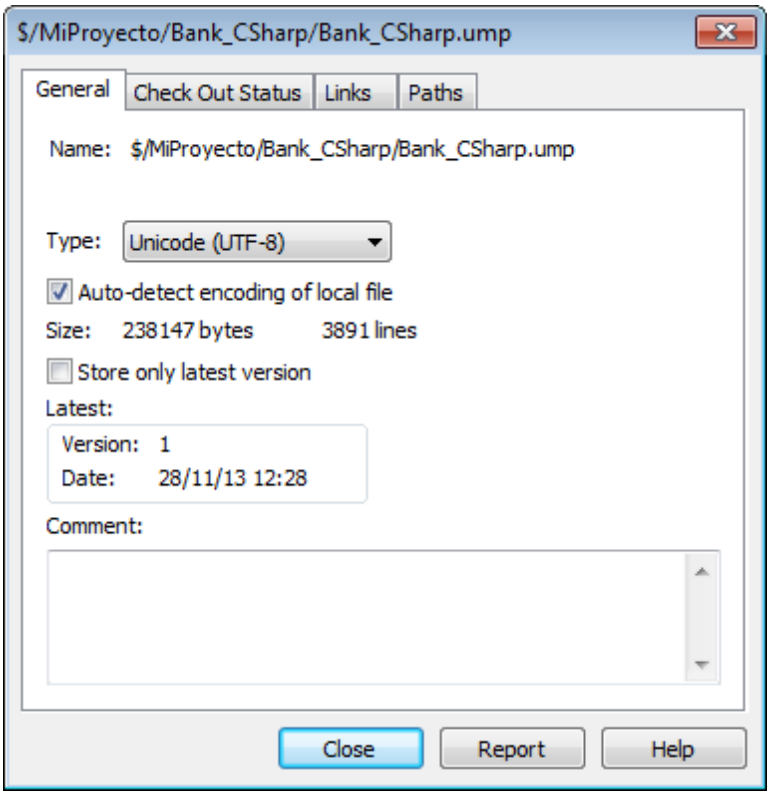
12.2.14 Mostrar propiedades

Este comando muestra las propiedades del archivo seleccionado y varía de un proveedor de control de código fuente a otro.

Para ver las propiedades del archivo seleccionado haga clic en **Proyecto | Control de código**

fuente | Mostrar propiedades.

Tenga en cuenta que este comando no se puede ejecutar en varios archivos a la vez.



12.2.15 Actualizar estado

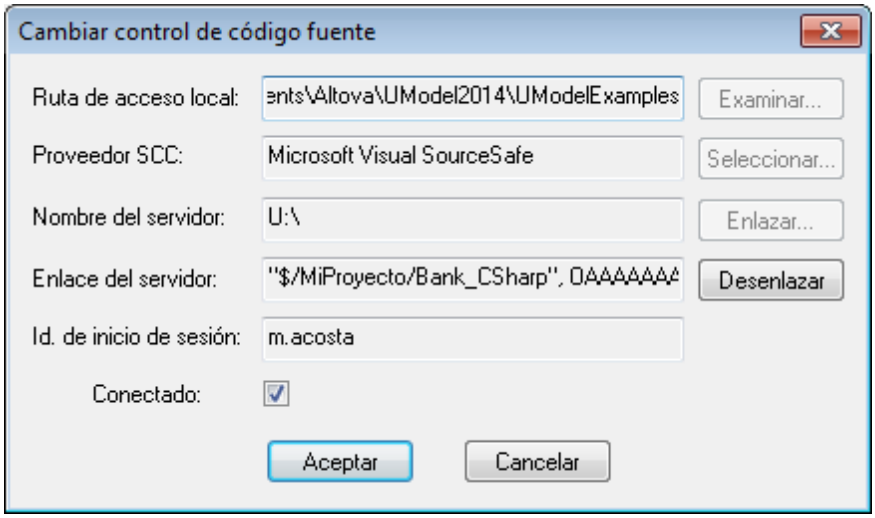
Este comando **actualiza** el estado de todos los archivos de proyecto, independientemente de cuál sea su estado actual.

12.2.16 Administrador del control de código fuente

Este comando **inicia** el software de control de código fuente, con su interfaz de usuario nativa.

12.2.17 Cambiar control de código fuente

Este cuadro de diálogo sirve para cambiar el enlace de control de código fuente activo. Haga clic en el botón **Desenlazar** y después (si quiere) haga clic en el botón **Seleccionar** para seleccionar un proveedor nuevo. Para terminar haga clic en el botón **Enlazar** para enlazar el proyecto con una ubicación nueva del repositorio.

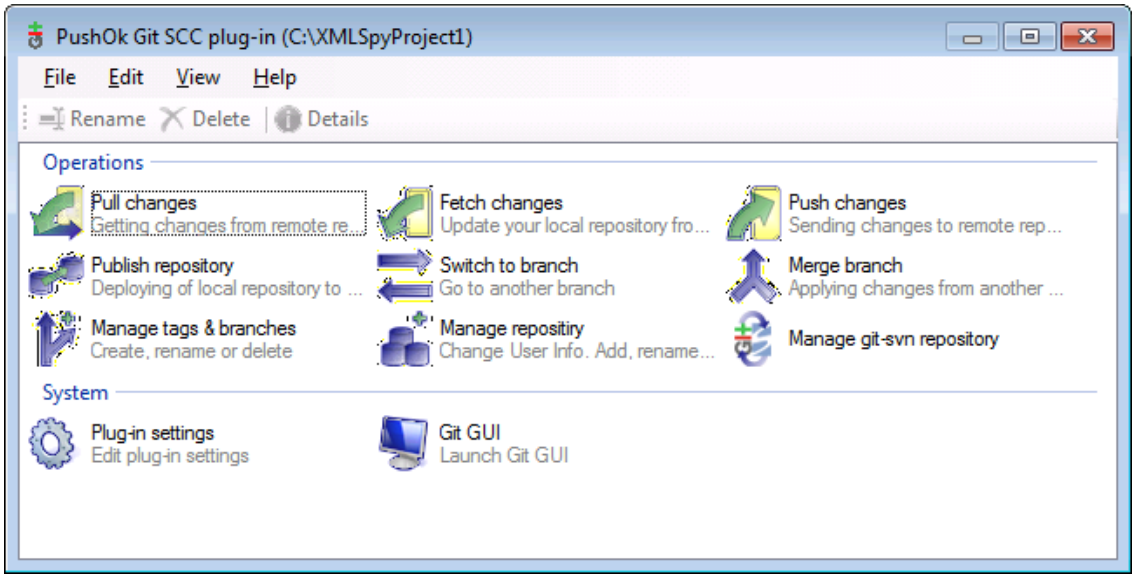

Cambiar control de código fuente
Ruta de acceso local:
ents\Altova\UModel2014\UModelExamples
Examinar...
Proveedor SCC:
Microsoft Visual SourceSafe
Seleccionar...
Nombre del servidor:
U:\
Enlazar...
Enlace del servidor:
"$/MiProyecto/Bank_CSharp", 0AAAAAAA
Desenlazar
Id. de inicio de sesión:
m.acosta
Conectado:
Aceptar
Cancelar

12.3 Control de código fuente con Git

UModel es compatible con el sistema de control de versiones Git por medio de un complemento externo llamado **GIT SCC plug-in** (http://www.pushok.com/software/git.html).

> Cuando se redactó esta documentación, la versión del complemento **GIT SCC plug-in** era una versión experimental. Para usar el complemento es necesario registrarse con el autor del complemento.

El complemento GIT SCC permite trabajar con repositorios Git utilizando los comandos del menú **Proyecto | Control de código fuente** de UModel. Recuerde que los comandos de este menú vienen de la API del complemento Microsoft Source Control, cuyo diseño es diferente al de Git. Como consecuencia, el complemento hace de intermediario entre las funciones tipo Visual Source Safe y las funciones de Git. Esto significa, por un lado, que algunos comandos como **Obtener la versión más reciente** no estarán habilitados cuando trabaje con Git. Por otro lado, hay acciones nuevas propias de Git que están disponibles en el cuadro de diálogo de administración del código fuente (**Proyecto | Control de código fuente | Administrador del control de código fuente** en UModel).



En el menú **Proyecto | Control de código fuente** también encontrará los comandos más frecuentes de Git.

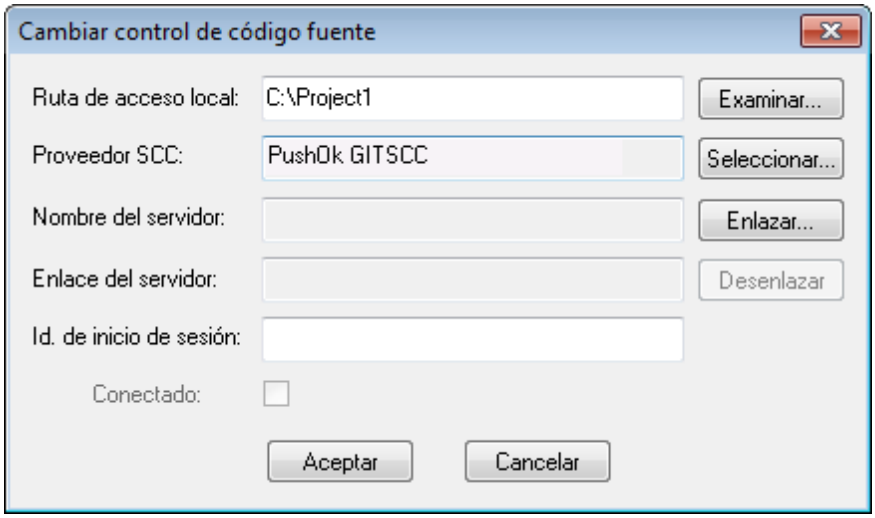
Los diferentes apartados de esta sección describen la configuración inicial del complemento y el flujo de trabajo básico:

- Habilitar Git con el complemento GIT SCC
- Agregar un proyecto al control de código fuente de Git
- Clonar un proyecto desde el control de código fuente de Git

12.3.1 Habilitar Git con el complemento de control de código fuente

Para habilitar el control de código fuente de Git en UModel es necesario tener instalado el complemento externo **PushOK GIT SCC plug-in**, registrarse y seleccionarlo en la lista de proveedores de control de código fuente:

1. Descargue el archivo de instalación del complemento desde el sitio web del autor (http://www.pushok.com), ejecútelo y siga las instrucciones que aparecen en pantalla.
2. En el menú **Proyecto** de UModel, haga clic en **Proyecto | Control de código fuente | Cambiar de control de código fuente** y seleccione **PushOk GITSCC**. Si **Push Ok GITSCC** no aparece en la lista de proveedores, es probable que la instalación del complemento no finalizara correctamente. Consulte la documentación del autor para resolver este problema.



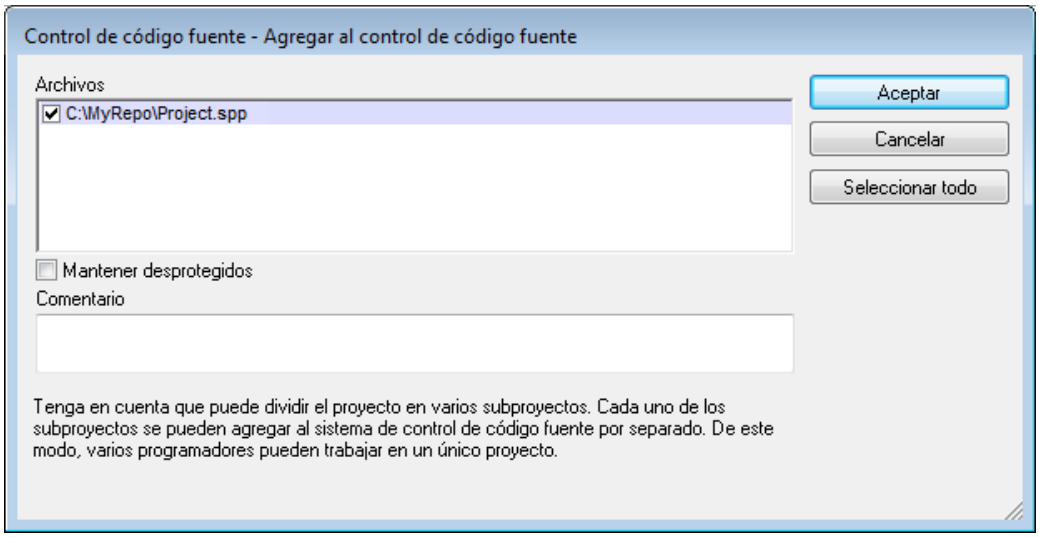
3. Para terminar debe registrar el complemento haciendo clic en **Registration**. Siga los pasos del asistente para terminar de registrar el complemento.

12.3.2 Agregar un proyecto al control de código fuente de Git

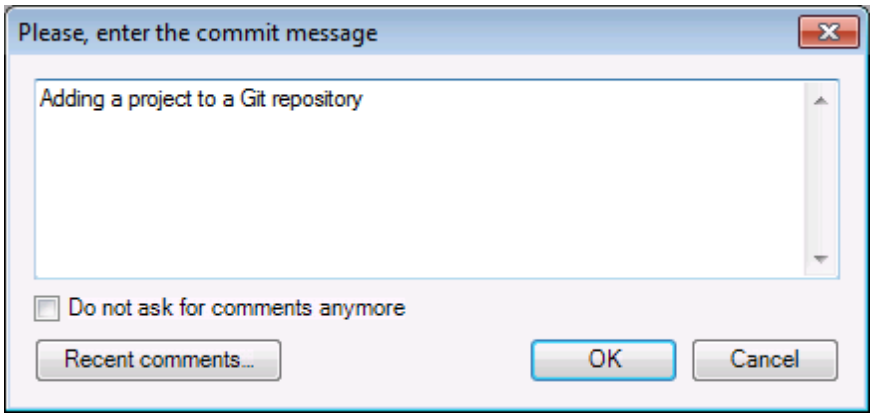
Puede guardar proyectos de UModel como repositorios de Git. La estructura de los archivos o carpetas que añada al proyecto se corresponderán con la estructura del repositorio Git.

Para agregar un proyecto al control de código fuente de Git:

1. Compruebe que el proveedor de control de código fuente seleccionado es **PushOK GIT SCC Plug-in** (*ver el apartado anterior*).
2. Cree un proyecto nuevo vacío y compruebe que no hay errores de validación (es decir, que no se detectan errores ni advertencias tras ejecutar el comando **Proyecto | Revisar la sintaxis del proyecto**).
3. Guarde el proyecto en una carpeta local (p. ej. `C:\MyRepo\Project.ump`).
4. En el panel Estructura del modelo haga clic en el nodo `Root`.
5. Ahora haga clic en **Proyecto | Control de código fuente | Agregar al control de código fuente**.



6. Haga clic en **Aceptar**.



7. Escriba el texto del mensaje de confirmación y haga clic en **OK** para agregar el proyecto al control de código fuente.

Ahora ya puede añadir elementos de modelado (diagramas, clases, paquetes, etc.) al proyecto. Recuerde que todos los archivos y carpetas del proyecto deben estar bajo la carpeta raíz del proyecto. Por ejemplo, si creó el proyecto en la carpetea `C:\MyRepo`, entonces solamente podrá añadir al proyecto los archivos que estén bajo `C:\MyRepo`. Si intenta añadir archivos de proyecto que estén fuera de la carpeta raíz del proyecto, aparecerá este mensaje de advertencia:

Sólo se pueden agregar archivos a una ubicación bajo la raíz de enlace del proyecto (C:\MyRepo).

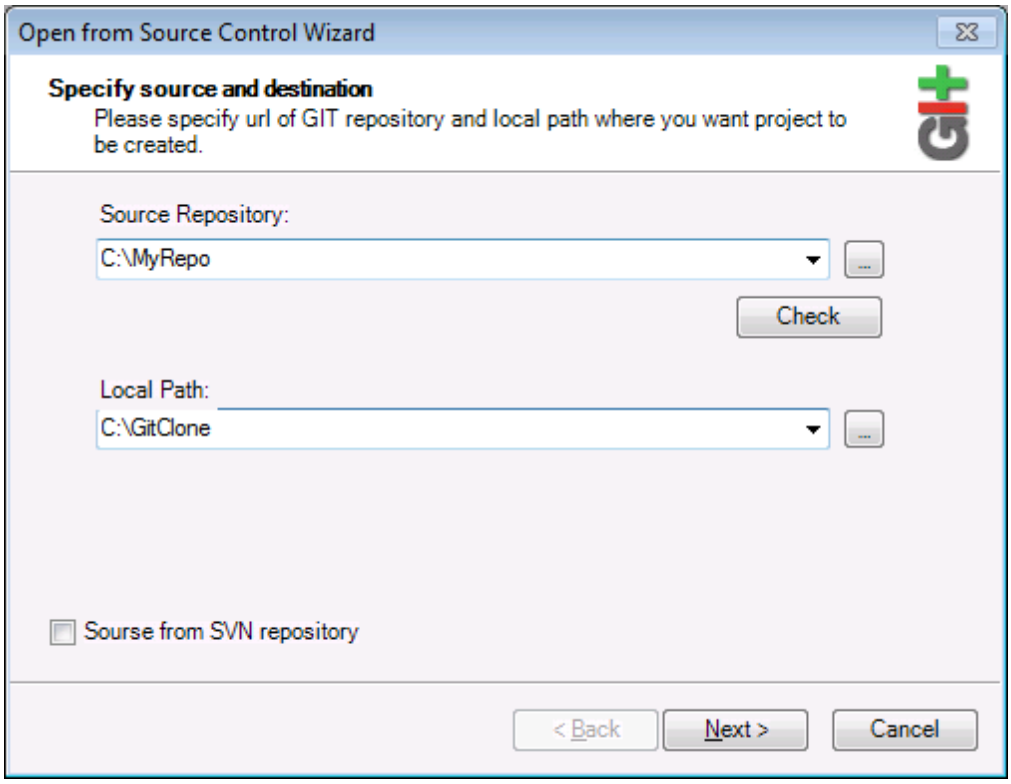
12.3.3 Clonar un proyecto desde el control de código fuente de Git

Los proyectos que ya estén en el control de código fuente de Git (*ver el apartado anterior*) se pueden abrir desde el repositorio Git:

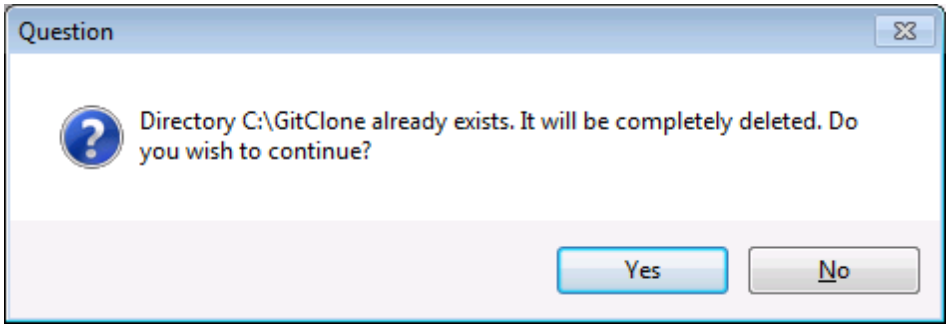
1. Compruebe que el proveedor de control de código fuente seleccionado es **PushOK GIT**

SCC Plug-in (*ver el apartado habilitar Git con complemento de control de código fuente GIT SCC*).

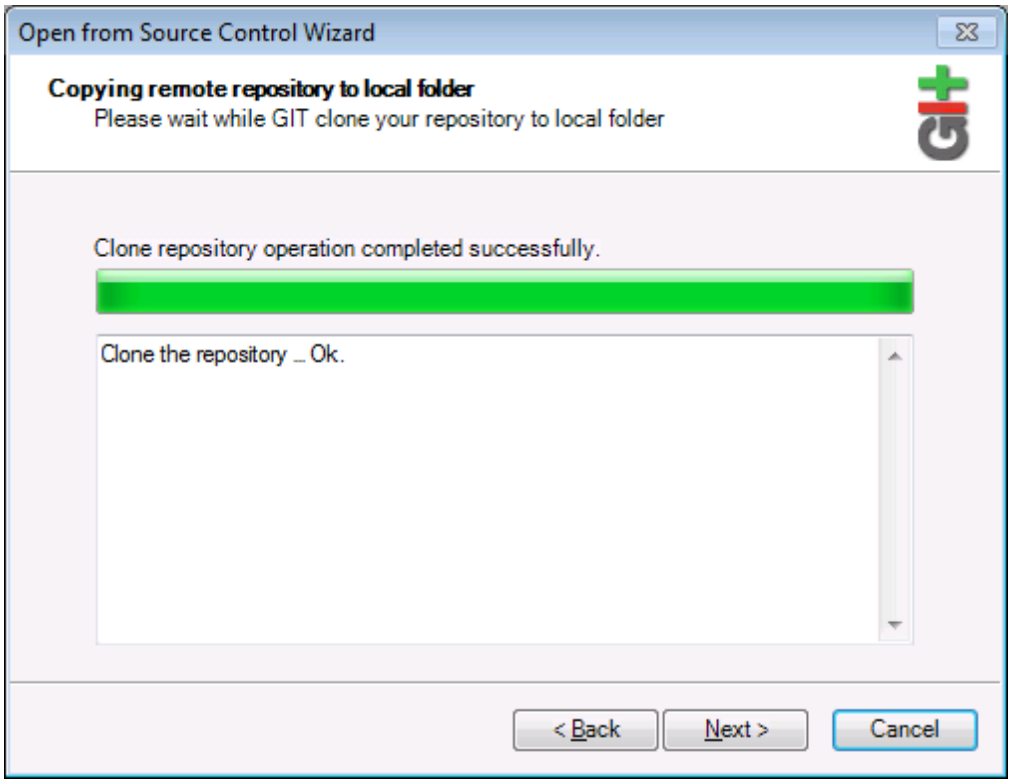
2. Haga clic en **Proyecto | Control de código fuente | Abrir desde el control de código fuente**.
3. Escriba la ruta de acceso o la URL del repositorio fuente. Haga clic en el botón **Check** para verificar la ruta de acceso o la dirección URL.



4. En el campo *Local Path* escriba la ruta de acceso de la carpeta local donde desea crear el proyecto y haga clic en **Next** para continuar. Si la carpeta local ya existe (aunque esté vacía), aparece este cuadro de diálogo preguntando si desea borrar totalmente la carpeta:



5. Haga clic en **Yes** para confirmar y después en **Next** para continuar.



6. Siga los pasos del asistente hasta el final.
7. Al final aparece un cuadro de diálogo "Explorar" donde puede abrir el proyecto de UModel (archivo *.ump). Seleccione el archivo de proyecto para cargar el contenido del proyecto en UModel.

Chapter 13

Iconos en los diagramas de UModel

13 Iconos en los diagramas de UModel

En UModel cada tipo de diagrama tiene una barra de herramientas distinta. Cada barra de herramientas ofrece iconos para los elementos compatibles con el tipo de diagrama correspondiente.

Estos iconos pueden ser de dos tipos:

- **Agregar:** en este grupo están los iconos de todos los elementos que se pueden agregar en el diagrama.
- **Relación:** en este grupo están todos los iconos de los tipos de relación que se pueden crear entre los elementos del diagrama.

13.1 Diagramas de actividades

Agregar:
Acción (AcciónLlamadaDeComportamiento)
Acción (AcciónOperaciónDeLlamada)
AcciónAceptarEvento
AcciónAceptarEvento (EventoDeTiempo)
AcciónEnviarSeñal

NodoDeDecisión (rama)
NodoDeCombinación
NodoInicial
NodoFinalDeActividad
NodoFinalDeFlujo
NodoDeBifurcación (vertical)
NodoDeBifurcación (horizontal)
NodoDeReunión
NodoDeReunión (horizontal)

PinDeEntrada
PinDeSalida
PinDeValor

NodoDeObjeto
NodoDeBúferCentral
NodoAlmacénDeDatos
ParticiónDeActividades (horizontal)
ParticiónDeActividades (vertical)
ParticiónDeActividades (2D)

FlujoDeControl
FlujoDeObjeto
ControladorDeExcepción

Actividad
NodoParámetroDeActividad
NodoDeActividadEstructurada
RegiónDeExpansión
NodoDeExpansión
RegiónDeActividadInterrumpible

Nota
Enlace de nota

13.2 Diagramas de clases

Relaciones:
Asociación
Agregación
Composición
ClaseDeAsociación
Dependencia
Utilización
RealizaciónDeInterfaz
Generalización

Agregar:
Paquete
Clase
Interfaz
Enumeración
TipoDeDatos
TipoPrimitivo
Perfil
Estereotipo
AplicaciónDePerfil
EspecificaciónDeInstancia

Nota
Enlace de nota

13.3 Diagramas de comunicación

Agregar:
LíneaDeVida
Mensaje (Llamada)
Mensaje (Respuesta)
Mensaje (Creación)
Mensaje (Destrucción)

Nota
Enlace de nota

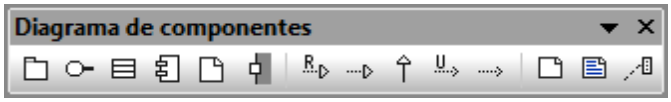
13.4 Diagramas de estructura de un compuesto

Agregar:
Colaboración
UsoDeColaboración
Parte (Propiedad)
Clase
Interfaz
Puerto

Relaciones:
Conector
Dependencia (Enlace de roles)
RealizaciónDeInterfaz
Utilización

Nota
Enlace de nota

13.5 Diagramas de componentes



Agregar:
Paquete
Interfaz
Clase
Componente
Artefacto

Relaciones:
Realización
RealizaciónDeInterfaz
Utilización
Dependencia

Nota
Enlace de nota

13.6 Diagramas de implementación

Agregar:
Paquete
Componente
Artefacto
Nodo
Dispositivo
EntornoDeEjecución

Relaciones:
Manifestación
Implementación
Asociación
Generalización
Dependencia

Nota
Enlace de nota

13.7 Diagramas global de interacción

Agregar:
AcciónLlamadaDeComportamiento (Interacción)
AcciónLlamadaDeComportamiento (UsoDeInteracción)
NodoDeDecisión
NodoDeCombinación
NodoInicial
NodoFinalDeActividad
NodoDeBifurcación
NodoDeBifurcación (Horizontal)
NodoDeReunión
NodoDeReunión (Horizontal)
RestricciónDeDuración

Relaciones:
FlujoDeControl

Nota
Enlace de nota

13.8 Diagramas de objetos

Agregar:
Paquete
Clase
Interfaz
Enumeración
TipoDeDatos
TipoPrimitivo
EspecificaciónDeInstancia

Relaciones:
Asociación
ClaseDeAsociación
Dependencia
Utilización
RealizaciónDeInterfaz
Generalización

Nota
Enlace de nota

13.9 Diagramas de paquetes

Agregar:
Paquete
Perfil

Relaciones:
Dependencia
ImportaciónDePaquete
CombinaciónDePaquete
AplicaciónDePerfil

Nota
Enlace de nota

13.10 Diagramas de perfil

Agregar:
Perfil
Estereotipo

Relaciones:
Generalización
AplicaciónDePerfil
ImportaciónDePaquete
ImportaciónDeElemento

Nota
Enlace de nota

13.11 Diagramas de máquina de estados de protocolos

Agregar:
Estado
Estado compuesto
Estado ortogonal
Estado de submáquina

EstadoFinal
EstadoInicial

PuntoDeEntrada
PuntoDeSalida
Elección
Unión
Terminar
Bifurcación
Bifurcación (horizontal)
Reunión
Reunión (horizontal)
ReferenciaDePuntoDeConexión

Relaciones:
TransiciónDeProtocolo

Nota
Enlace de nota

13.12 Diagramas de secuencia

Agregar
LíneaDeVida
FragmentoCombinado
FragmentoCombinado (Alternativos)
FragmentoCombinado (Bucle)
UsoDeInteracción
Puerta
InvarianteDeEstado
RestricciónDeDuración
RestricciónDeTiempo

Mensaje (Llamada)
Mensaje (Respuesta)
Mensaje (Creación)
Mensaje (Destrucción)

Mensaje asíncrono (Llamada)
Mensaje asíncrono (Respuesta)
Mensaje asíncrono (Destrucción)

Mensajes sin numeración
Mensajes con numeración sencilla
Mensajes con notación decimal anidada

Nota
Enlace de nota

Activar/desactivar el movimiento de mensajes dependientes
Activar/desactivar la creación automática de respuestas para mensajes (Llamada)
Activar/desactivar la creación automática de operaciones en el destino al escribir el nombre de la operación

13.13 Diagramas de máquina de estados

Agregar:
Estado
Estado compuesto
Estado ortogonal
Estado de submáquina

EstadoFinal
EstadoInicial

PuntoDeEntrada
PuntoDeSalida
Elección
Unión
Terminar
Bifurcación
Bifurcación (horizontal)
Reunión
Reunión (horizontal)
HistorialDetallado
HistorialSuperficial
ReferenciaDePuntoDeConexión

Relaciones:
Transición

Nota
Enlace de nota

Activar/desactivar la creación automática de operaciones en el destino al escribir el nombre de la operación

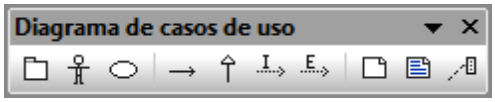
13.14 Diagramas de ciclo de vida

Agregar
LíneaDeVida (Estado o Condición)
LíneaDeVida (Valor general)
MarcaDeGraduación
Evento/estímulo
RestricciónDeDuración
RestricciónDeTiempo

Mensaje (Llamada)
Mensaje (Respuesta)
Mensaje asíncrono (Llamada)

Nota
Enlace de nota

13.15 Diagramas de casos de uso



Agregar:
Paquete
Actor
CasoDeUso

Relaciones:
Asociación
Generalización
Inclusión
Extensión

Nota

13.16 Diagramas de esquema XML

Agregar:
targetNamespace XSD
schema XSD
element (global) XSD
group XSD
complexType XSD
complexType (simpleContent) XSD
simpleType XSD
list XSD
union XSD
enumeration XSD
attribute XSD
attributeGroup XSD
notation XSD
import XSD

Relaciones:
include XSD
redefine XSD
restriction XSD
extension XSD
substitution XSD

Nota
Enlace de nota

Chapter 14

Referencia del usuario

14 Referencia del usuario

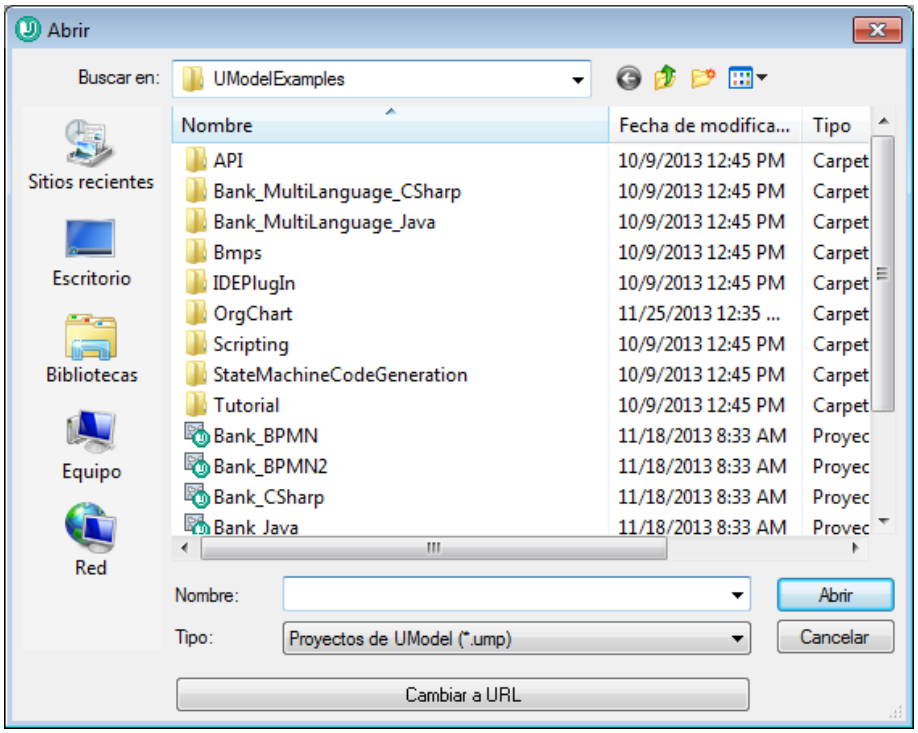
Esta sección repasa uno a uno todos los menús y comandos de menú de UModel, dando una breve descripción de ellos.

14.1 Menú Archivo

Nuevo
Si tiene abierto un proyecto, este comando lo cierra y crea un proyecto nuevo.

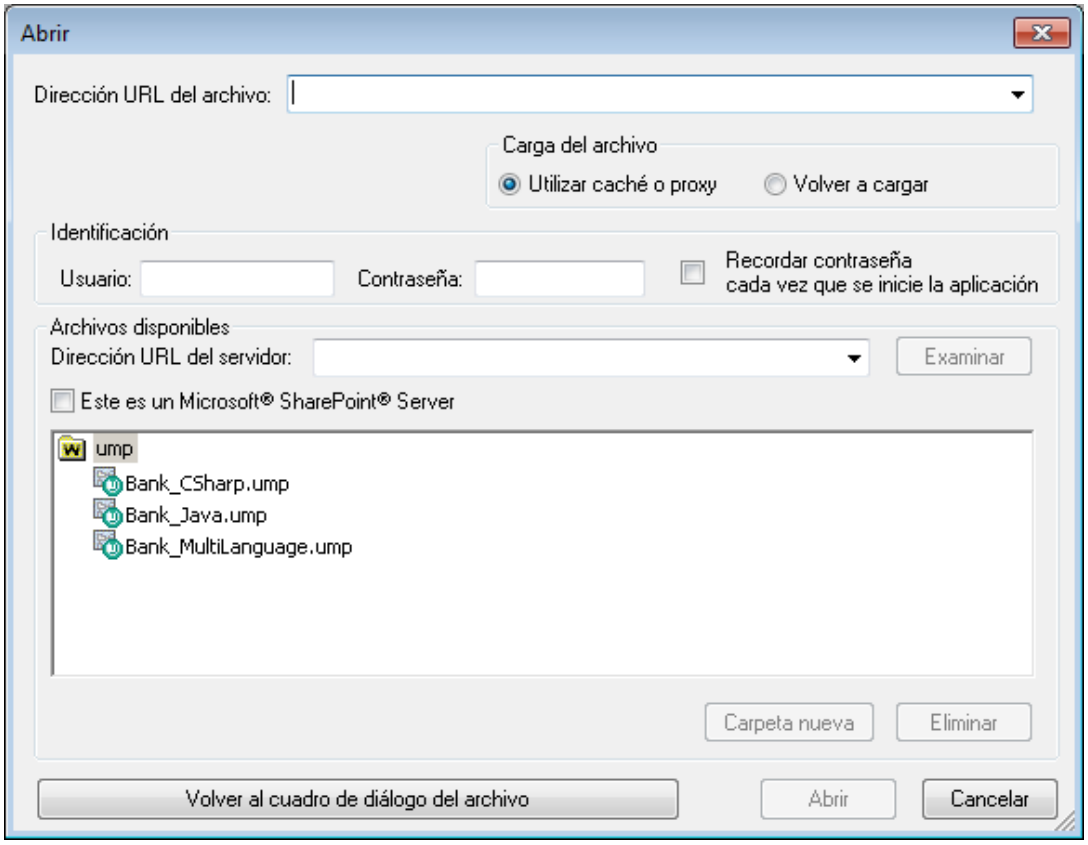
Abrir
Este comando abre un proyecto de modelado ya existente. Seleccione el archivo de proyecto `*.ump` en el cuadro de diálogo "Abrir" (*imagen siguiente*).

Observe que en la parte inferior del diálogo aparece el botón **Cambiar a URL**.



El botón **Cambiar a URL** cambia el cuadro de diálogo al modo URL, que sirve para abrir archivos de proyecto de UModel desde una dirección URL.

1. Escriba la dirección URL a la que desea acceder (en el campo *Dirección URL del servidor*).
2. Escriba el nombre de usuario y la contraseña (si el servidor está protegido con contraseña).
3. Haga clic en el archivo que desea cargar en UModel. Si el servidor es Microsoft® SharePoint®, marque la casilla *Este es un Microsoft® SharePoint® Server*. Para más información consulte las notas que aparecen más abajo.



La URL del archivo aparece en el campo *Dirección URL del archivo* (en la parte superior del cuadro de diálogo).
El botón **Abrir** se habilita en este momento.

4. Haga clic en el botón **Abrir** para cargar el archivo. El archivo aparece en la ventana principal de la aplicación.

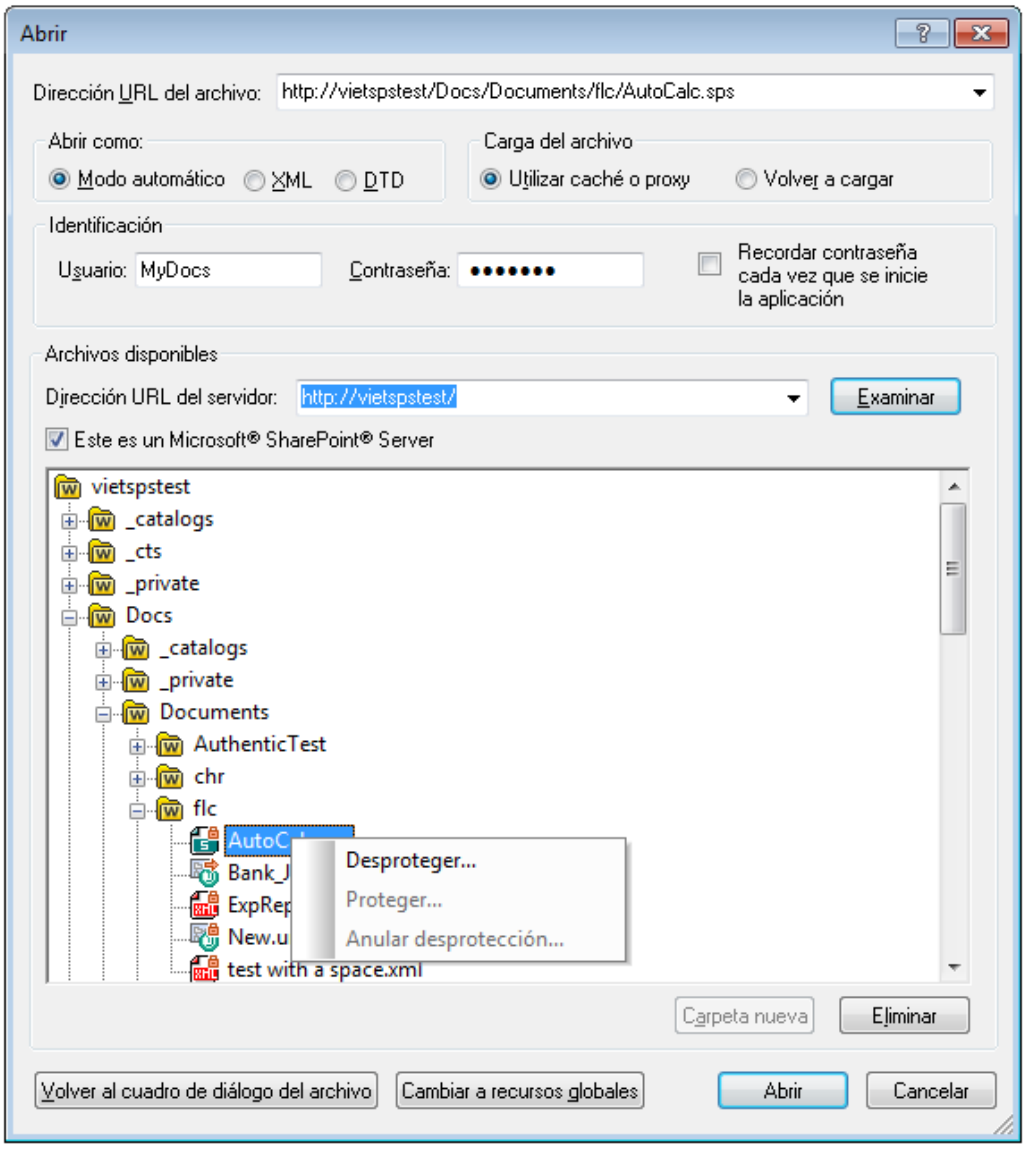
   Nota importante: el botón **Examinar** solamente se puede usar con servidores compatibles con WebDAV y con servidores Microsoft SharePoint. Los protocolos compatibles son FTP, HTTP y HTTPS.

   Además puede cargar el archivo por la memoria caché local o por un servidor proxy (lo cual acelera el proceso considerablemente si ya el archivo con anterioridad). Por último, si está trabajando en un sistema de publicación electrónica, puede volver a cargar el archivo seleccionando la opción *Volver a cargar*.

Notas sobre Microsoft® SharePoint® Server

Es necesario tener en cuenta algunas características de los archivos residentes en servidores Microsoft® SharePoint® Server:

- En la estructura de directorios que aparece en el panel *Archivos disponibles* los iconos de archivo tienen símbolos que indican el estado de protección de los archivos.



Al hacer clic con el botón derecho en un archivo aparece un menú contextual (*imagen anterior*).

- Estos son los iconos de archivo:

	Protegido. Se puede desproteger.
	Desprotegido por otro usuario. No se puede desproteger.
	Desprotegido localmente. Se puede editar y después proteger.

- Tras desproteger el archivo podrá editarlo en la aplicación de Altova y guardarlo con el comando **Archivo | Guardar (Ctrl+S)**.
- El archivo editado se puede proteger con el menú contextual del cuadro de diálogo "Abrir URL" (*imagen anterior*).
- Si otro usuario desprotegió un archivo, dicho archivo no se puede desproteger.
- Si un archivo está desprotegido localmente (por usted), puede deshacer la desprotección

con el comando **Deshacer desprotección** del menú contextual. Como resultado se devuelve el archivo al servidor sin ningún cambio.

- Si desprotege un archivo en una aplicación de Altova, no puede desprotegerlo en otra aplicación de Altova. En ese caso los comandos disponibles en la aplicación de Altova son **Proteger** y **Deshacer desprotección**

Volver a cargar
Este comando sirve para volver a cargar el proyecto y guardar (o descartar) los cambios realizados hasta ese momento.

Guardar
Este comando guarda el proyecto de modelado activo con el nombre de archivo actual.

Guardar como
Este comando guarda el proyecto de modelado activo con otro nombre. También ofrece la opción de darle al proyecto un nombre si es la primera vez que se guarda.

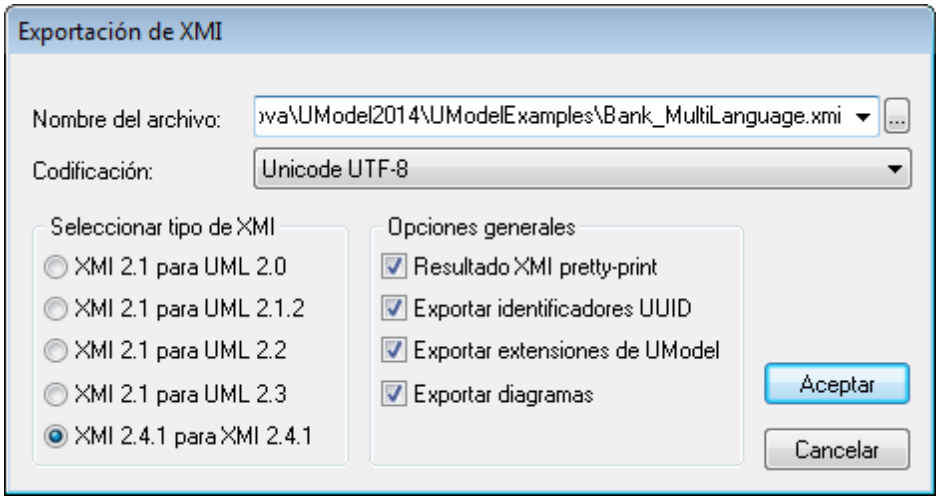
Guardar copia como
Este comando guarda una copia del proyecto de modelado activo con otro nombre de archivo.

Guardar el diagrama como imagen
Este comando abre el cuadro de diálogo "Guardar como" y permite guardar el diagrama activo como archivo .PNG o .EMF (metarchivo mejorado). También puede guardar archivos PNG de gran tamaño (1GB o más).

Guardar todos los diagramas como imagen
Este comando guarda todos los diagramas del proyecto activo como archivos .PNG o .EMF (metarchivo mejorado).

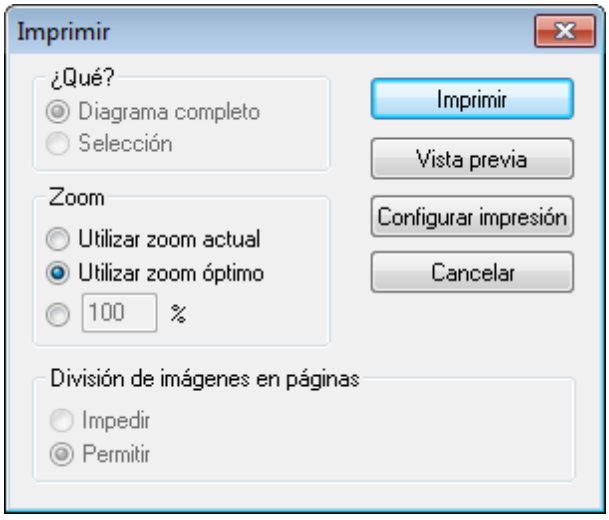
Importar desde un archivo XMI
Este comando importa un archivo XMI exportado con anterioridad. Si el archivo se generó con UModel, toda las extensiones y demás elementos se conservarán.

Exportar a un archivo XMI
Este comando exporta el modelo como archivo XMI (*imagen siguiente*). Puede seleccionar la versión de UML y los identificadores que desea exportar. Para más información consulte el apartado XMI: intercambio de metadatos XML.



Enviar por correo electrónico
Este comando abre la aplicación predeterminada de correo electrónico e inserta el proyecto de UModel actual como archivo adjunto.

Imprimir
Este comando abre el cuadro de diálogo "Imprimir" (*imagen siguiente*), desde donde puede imprimir una copia en papel del proyecto de modelado.



- *Utilizar zoom actual:* seleccione esta opción si quiere usar el factor de zoom actual del proyecto de modelado. Si selecciona esta opción, se habilita el grupo de opciones *División de imágenes en páginas.*
- *Impedir:* seleccione esta opción para que los elementos de modelado no se dividan en dos páginas y se mantengan como una unidad.
- *Utilizar zoom óptimo:* elija esta opción para ajustar el tamaño del proyecto de modelado al tamaño de la página. También puede especificar el factor de zoom numéricamente.

Imprimir todos los diagramas

Este comando abre el cuadro de diálogo "Imprimir" e imprime todos los diagramas UML que contiene el archivo de proyecto actual.

Vista previa de impresión
Este comando abre el cuadro de diálogo "Imprimir".

Configurar impresión
Este comando abre el cuadro de diálogo "Configurar impresión", donde puede definir la impresora que desea usar y seleccionar la configuración del papel.

14.2 Menú Edición

Deshacer 
UModel permite eliminar todos los cambios realizados y devolver el archivo a versiones anteriores. Todos los cambios se pueden deshacer uno por uno y no hay un límite de operaciones deshacer.

Rehacer 
Permite rehacer las acciones que deshizo con el comando **Deshacer**. Esto significa que puede ir adelante y atrás en el historial de acciones con los comandos **Deshacer** y **Rehacer**.

Cortar/Copiar/Eliminar
Estos comandos de edición estándar de Windows sirven para cortar, copiar y eliminar elementos de modelado. Para más información consulte el apartado Cortar, copiar y pegar en los diagramas de UModel.

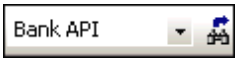
Pegar
Este comando (menú **Edición**, menú contextual o **Ctrl+V**) siempre añade un elemento de modelado nuevo al diagrama y a la Estructura del modelo. Para más información consulte el apartado Cortar, copiar y pegar en los diagramas de UModel.

Pegar sólo en el diagrama
Si usa este comando desde el menú contextual (tras hacer clic con el botón derecho en el fondo del diagrama), se añade un vínculo/una vista del elemento existente al diagrama actual y no a la Estructura del modelo. Para más información consulte el apartado Cortar, copiar y pegar en los diagramas de UModel.

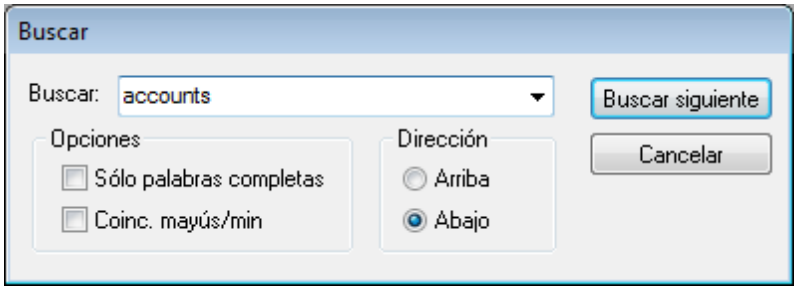
Eliminar sólo en el diagrama
Este comando elimina los elementos seleccionados del diagrama activo. Sin embargo, los elementos no se eliminan del proyecto de modelado y siguen estando disponibles en la Estructura del modelo. Recuerde que este comando no sirve para eliminar propiedades ni operaciones de clase. Las propiedades y operaciones se pueden seleccionar y eliminar en la clase directamente.

Seleccionar todo
Este comando selecciona todos los elementos de modelado del diagrama activo. Equivale a utilizar **Ctrl+A**.

Buscar
En UModel puede buscar elementos de modelado de varias formas:

- Desde el cuadro de texto de la **barra de título** principal 

- Con el comando de menú **Edición | Buscar**.
- Con la tecla de acceso rápido **Ctrl+F** (que abre el cuadro de diálogo "Buscar").



En el cuadro de diálogo "Buscar" puede buscar texto:

- En los paneles Estructura del modelo, Árbol de diagramas y Favoritos.
- En el panel Documentación.
- En cualquier diagrama activo.
- En el panel Mensajes.

Buscar siguiente  **(F3)**
Este comando repite la última búsqueda realizada con el comando **Buscar** y busca la siguiente instancia del término de búsqueda en el diagrama o en la pestaña activos.

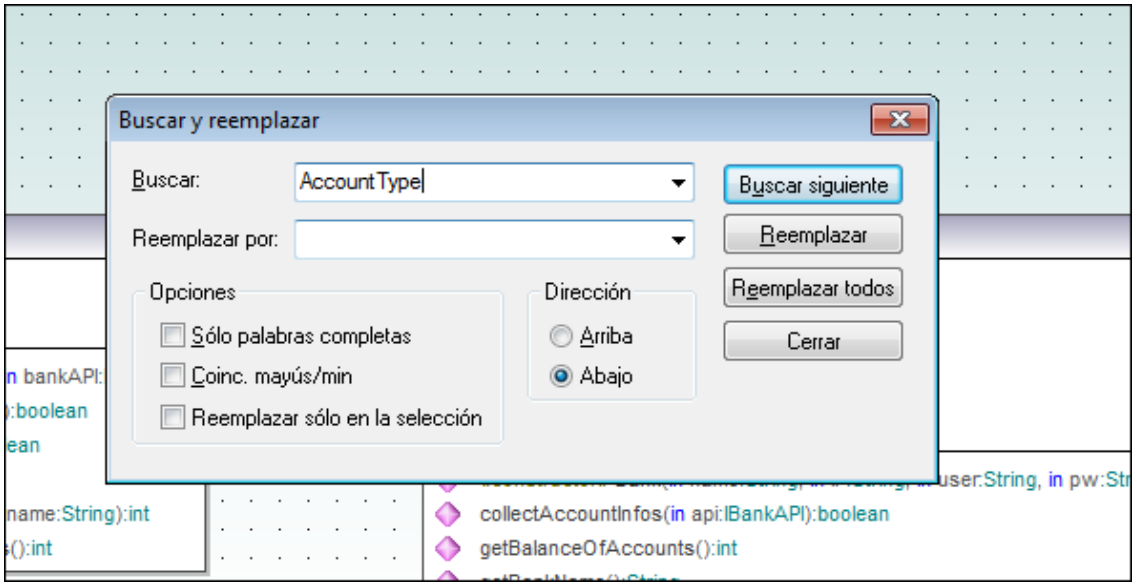
Buscar anterior (Mayús+F3)
Este comando repite la última búsqueda realizada con el comando **Buscar** y busca la instancia anterior del término de búsqueda en el diagrama o en la pestaña activos.

Reemplazar

Este comando busca y reemplaza elementos de modelado en el proyecto. Una vez encontrado, el elemento se resalta en el diagrama y en la Estructura del modelo.

Puede usar la función de búsqueda y reemplazo:

- En todos los diagramas.
- En los paneles Estructura del modelo, Árbol de diagramas y Favoritos.
- En el panel Documentación.



Copiar como mapa de bits
Este comando copia el diagrama activo en el portapapeles. Después podrá pegar el diagrama en cualquier aplicación.

Nota: los diagramas se copian en el portapapeles del sistema. Para verlos o acceder a ellos debe insertarlos en otra aplicación.

Copiar la selección como mapa de bits
Este comando copia los elementos de diagrama **seleccionados** en el portapapeles. Después podrá pegarlos en cualquier aplicación.

14.3 Menú Proyecto

Revisar la sintaxis del proyecto...
Este comando sirve para revisar sintaxis del proyecto de UModel. Las tablas que aparecen a continuación detallan qué aspectos del archivo de proyecto se revisan:

	Se comprueba si...	**Mensaje...**
A nivel de proyecto	existe una raíz de espacio de nombres Java como mínimo	Error
A nivel de componente	se estableció el archivo / directorio de proyecto	Error
	existe Realización	Error
	está desactivada la casilla *Usar para ingeniería de código*. Si es así, el componente no se revisa y se deshabilita la revisión de sintaxis en el componente.	Ninguno

Nivel	**Se comprueba si...**	**Mensaje...**
`Clase`	se estableció el nombre de archivo.	**Error** si la opción local *Generar los nombres de archivo de código que falten* no está activada. **Advertencia** si esa opción está activada.
	la clase está anidada (si lo está, no se realiza la revisión de sintaxis).	
	está en un espacio de nombres de lenguaje de código	Error
	se estableció el tipo de parámetro para las operaciones	Error
	se estableció el tipo para las propiedades	Error
	se estableció el tipo devuelto de las operaciones	Error
	hay operaciones duplicadas (nombres y tipos de parámetro)	Error
	alguna clase participa en Realización (solo si la clase no está anidada)	Advertencia
`Interfaz`	se estableció el nombre de archivo.	**Error** si no está activada la opción local *Generar los nombres de archivo de código que falten*. **Advertencia** si esa opción está activada.
	la interfaz está en un espacio de nombres de lenguaje de código	Error
	se estableció el tipo para las propiedades	Error
	se estableció el tipo de parámetro para las	Error

	operaciones	
	se estableció el tipo devuelto de las operaciones	Error
	hay operaciones duplicadas (nombres y tipos de parámetro)	Error
	alguna interfaz participa en una RealizaciónDeComponente	Advertencia
Enumeración	pertenece a la raíz de espacio de nombres Java.	Advertencia (no se generará código)
	no pertenece a la raíz de espacio de nombres Java. Entonces se deshabilita la revisión de sintaxis en la enumeración. El paquete no se revisa.	Ninguno

Revisión de sintaxis en los elementos UML que intervienen en la generación de código

clase	se comprueba si el nombre es un nombre Java válido (no contiene caracteres prohibidos y el nombre no es una palabra clave)	Error
propiedad de clase	se comprueba si el nombre es un nombre Java válido (no contiene caracteres prohibidos y el nombre no es una palabra clave)	Error
operación de clase	se comprueba si el nombre es un nombre Java válido (no contiene caracteres prohibidos y el nombre no es una palabra clave). Se busca si existe un parámetro de devolución	Error
parámetro de operación de clase	se comprueba si el nombre es un nombre Java válido (no contiene caracteres prohibidos y el nombre no es una palabra clave). Se comprueba si el tipo tiene un nombre de tipo Java válido	Error
interfaz	se comprueba si el nombre es un nombre Java válido (no contiene caracteres prohibidos y el nombre no es una palabra clave)	Error
operación de interfaz	se comprueba si el nombre es un nombre Java válido (no contiene caracteres prohibidos y el nombre no es una palabra clave)	Error
parámetro de operación de interfaz	se comprueba si el nombre es un nombre Java válido (no contiene caracteres prohibidos y el nombre no es una palabra clave)	Error
propiedades de la interfaz	se comprueba si el nombre es un nombre Java válido (no contiene caracteres prohibidos y el nombre no es una palabra clave)	Error
paquete con espacio de nombres de estereotipo	se comprueba si el nombre es un nombre Java válido (no contiene caracteres prohibidos y el nombre no es una palabra clave)	Error
paquete sin espacio de nombres de estereotipo	no hay nada que se pueda revisar	Ninguno
clase	herencia múltiple	Error

Nota: la revisión sintáctica no revisa las restricciones de los elementos de modelado porque no intervienen en el proceso de generación de código Java. Para más información consulte el apartado Crear restricciones en los elementos de modelado.

Control de código fuente

UModel es compatible con Microsoft SourceSafe y otros sistemas de control de versiones.

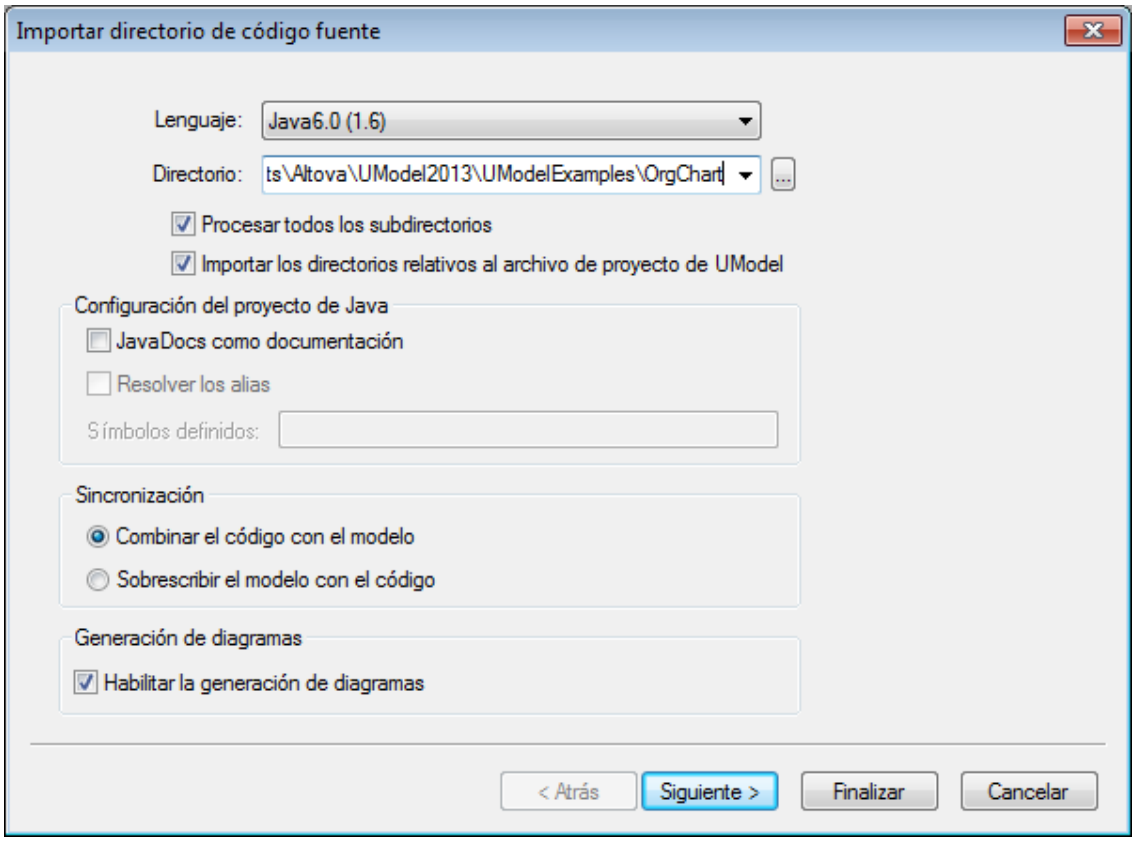
Microsoft define una entrada del registro, donde todos los programas compatibles con sistemas SCC (de control de versiones) se pueden registrar. UModel solamente lee esta entrada.

HKEY_LOCAL_MACHINE\SOFTWARE\SourceCodeControlProvider\InstalledSCCProviders

Recuerde también que los complementos de control de versiones no se instalan automáticamente con todos los sistemas SCC. Para más información consulte la documentación de su software de control de versiones. También puede consultar el apartado Control de código fuente para obtener más información sobre servidores y clientes de control de código fuente y cómo utilizarlos.

Importar directorio de código fuente...

Este comando abre el cuadro de diálogo "Importar directorio de código fuente" (*imagen siguiente*). Para ver un ejemplo concreto consulte el apartado Ingeniería de ida y vuelta (código - modelo - código).



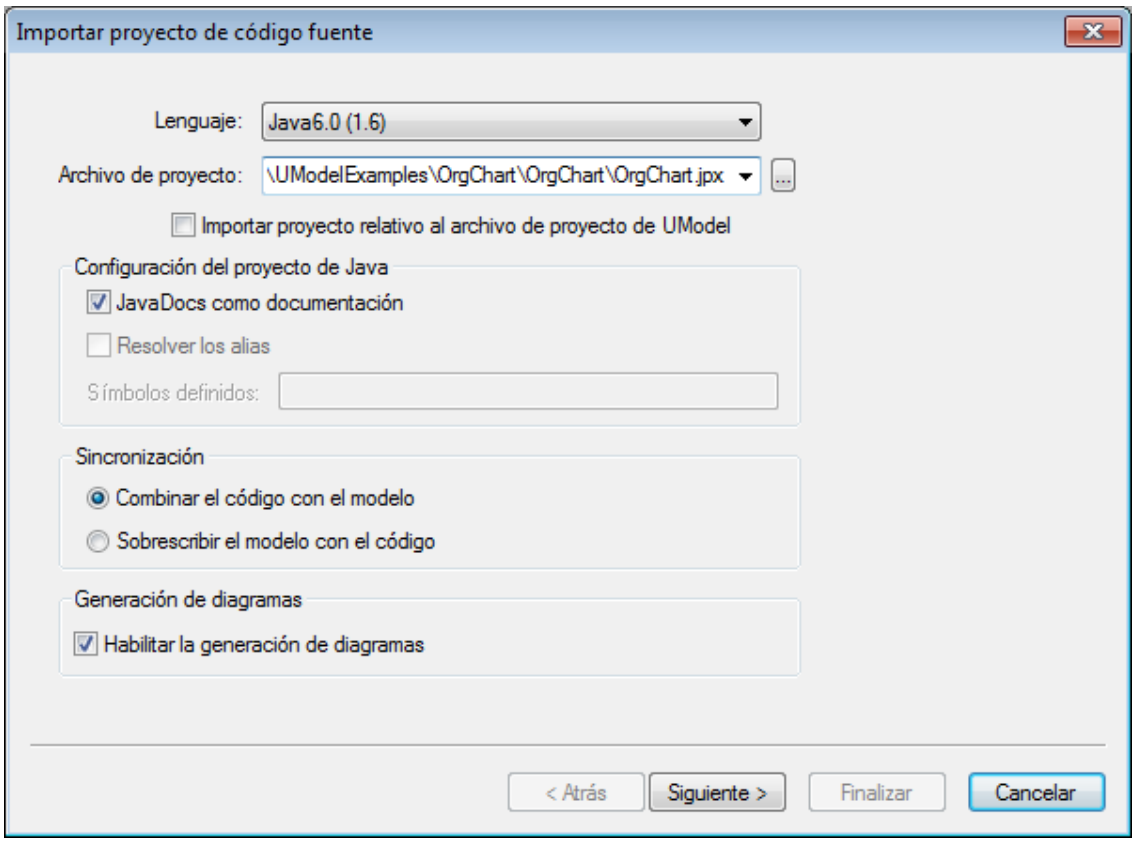
Importar proyecto de código fuente...

Este comando abre el cuadro de diálogo "Importar proyecto de código fuente" (*imagen siguiente*).

Haga clic en el botón **Examinar** ... para seleccionar el archivo de proyecto y el tipo de proyecto. Para ver un ejemplo concreto consulte el apartado Importar código fuente a los proyectos.

Proyectos Java:

- UModel admite proyectos de JBuilder **.jpx**, Eclipse **.project** y NetBeans (p. ej.: proyecto**.xml**).



Proyectos C#:

- Proyectos de MS Visual Studio (**csproj**, **csdprj**...)
- Proyectos de Borland **.bdsproj**

Proyectos de VB.NET:

- Proyectos de MS Visual Studio (**vbproj**, **vbdproj**.)

Importar tipos binarios

Abre el cuadro de diálogo "Importar tipos binarios", que sirve para importar archivos Java, C# y VB binarios. Para más información consulte el apartado Importar binarios Java, C# y VB.

Importar directorio del esquema XML

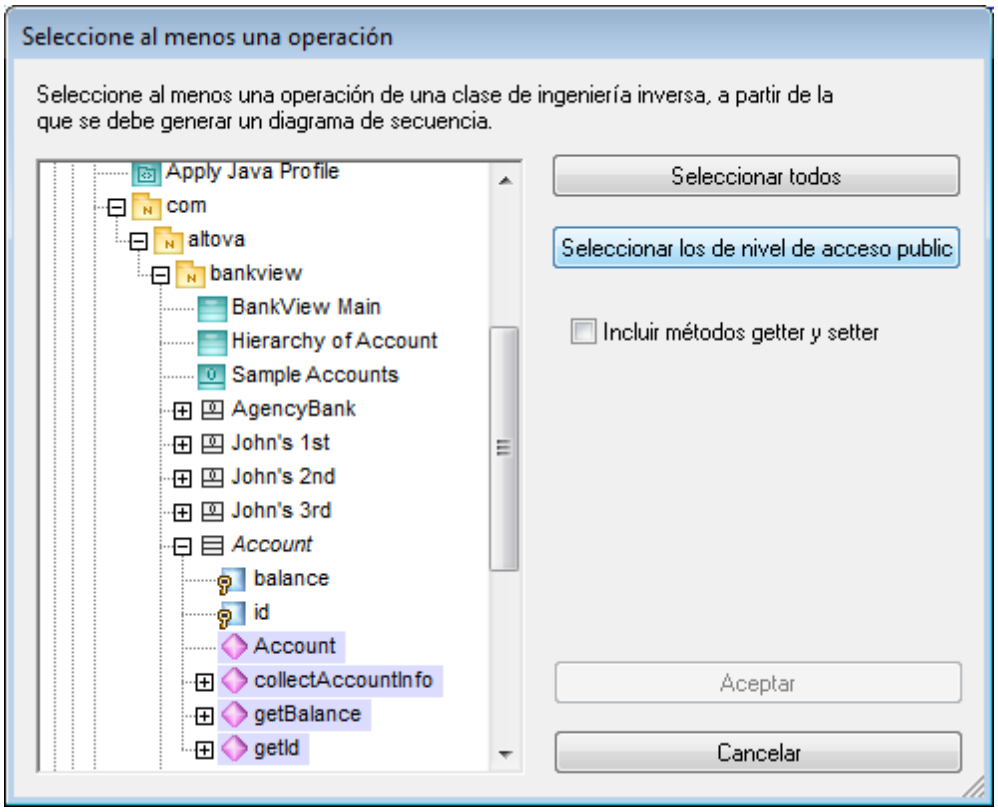
Abre el cuadro de diálogo "Importar el directorio del esquema XML", que sirve para importar todos los esquemas XML del directorio seleccionado y también los de sus subdirectorios.

Importar archivo de esquema XML

Abre el cuadro de diálogo "Importar archivo de esquema XML", que sirve para importar archivos de esquema. Consulte el apartado Diagramas de esquema XML para obtener más información.

Generar diagramas de secuencia a partir del código

Abre el cuadro de diálogo "Seleccione al menos una operación" (*imagen siguiente*), donde puede seleccionar las operaciones que deben utilizarse como base del diagrama de secuencia generado.



Tras seleccionar las operaciones, haga clic **Aceptar**. Se abre el cuadro de diálogo "Generación de diagrama de secuencia", donde puede configurar la generación.

Combinar el código de programa con el proyecto de UModel

Abre el cuadro de diálogo "Configurar sincronización" por la pestaña *Sincronizar el código con el modelo*. Haga clic en el botón **Configuración del proyecto** para seleccionar opciones de configuración para el lenguaje de programación elegido.

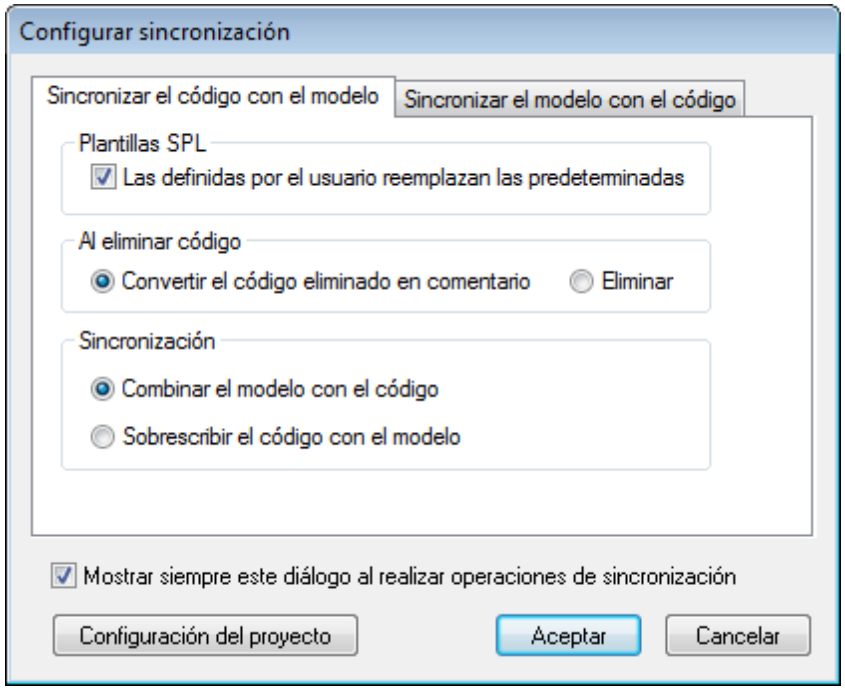
Combinar o sobrescribir código

Supongamos que el código se generó a partir de un modelo y que desde entonces se realizaron cambios tanto en el modelo como en el código. Por ejemplo, supongamos que se realizaron estos cambios:

- En UModel se añadieron elementos de modelado nuevos (p. e. se añadió una clase nueva llamada `X`).
- Al código externo se añadió una clase nueva llamada `Y`.

Combinar (el modelo con el código) significa que:
- la clase nueva Y que se añadió en el código externo **se conserva**.
- la clase nueva X que se añadió en UModel **se añade al código**.

Sobrescribir (el código con el modelo) significa que:
- la clase nueva Y que se añadió en el código externo **se elimina**.
- la clase nueva X que se añadió en UModel **se añade al código**.



Combinar el proyecto de UModel con el código de programa

Abre el cuadro de diálogo "Configurar sincronización" por la pestaña *Sincronizar el modelo con el código*. Haga clic en el botón **Configuración del proyecto** para seleccionar opciones de configuración para el lenguaje de programación elegido.

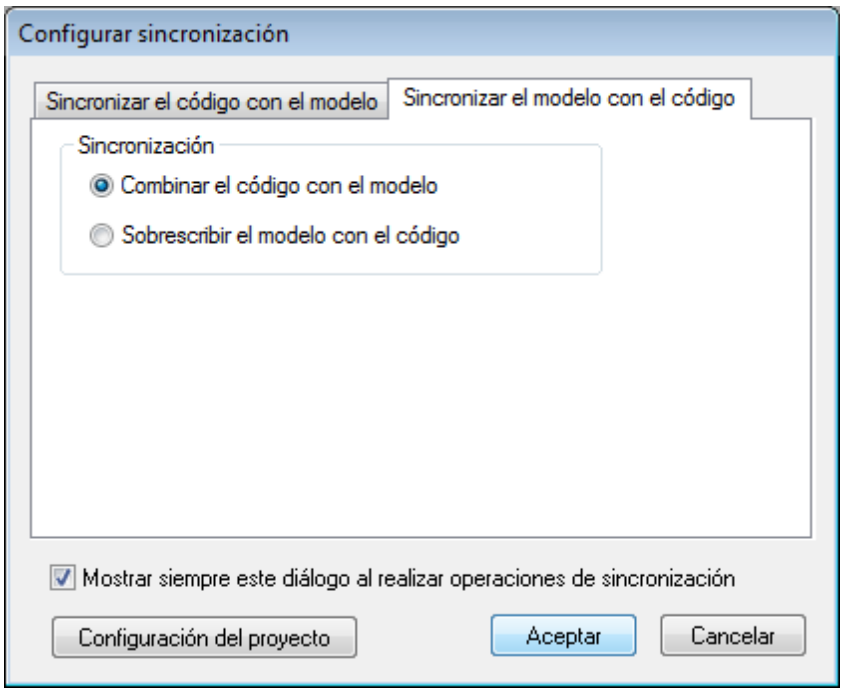
Combinar o sobrescribir el código

Supongamos que el código se generó a partir de un modelo y que desde entonces se realizaron cambios tanto en el modelo como en el código. Por ejemplo, supongamos que se realizaron estos cambios:

- En UModel se añadieron elementos de modelado nuevos (p. e. se añadió una clase nueva llamada `X`).
- Al código externo se añadió una clase nueva llamada `Y`.

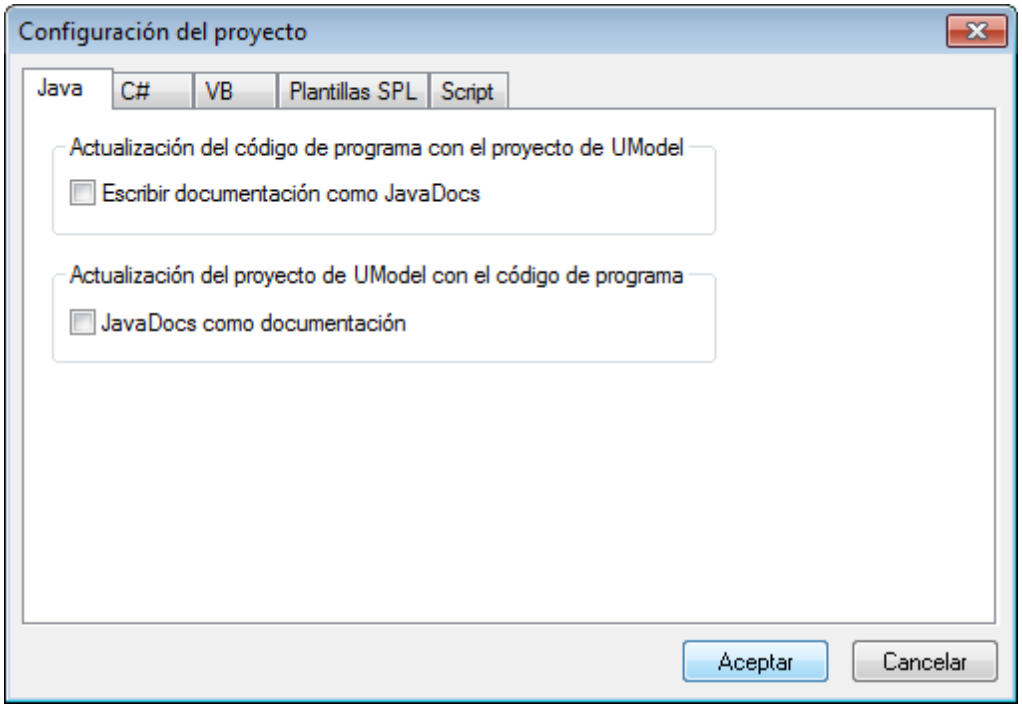
Combinar (el código con el modelo) significa que:
- la clase nueva X que se añadió en UModel **se conserva**.
- la clase nueva Y que se añadió en el código externo **se añade al modelo**.

Sobrescribir (el modelo con el código) significa que:
- la clase nueva X que se añadió en UModel **se elimina**.
- la clase nueva Y que se añadió en el código externo **se añade al modelo**.



Configuración del proyecto

Este comando sirve para definir opciones de configuración para cada lenguaje de programación.



Configurar sincronización...

Abre el cuadro de diálogo "Configurar sincronización" (*ver imágenes más arriba*).

Combinar el proyecto...
Combina dos proyectos de UModel en uno solo modelo. El primer archivo que se abre es con el que se combina el segundo archivo. Consulte el apartado Combinar proyectos de UModel para obtener más información.

Incluir un subproyecto
Consulte el apartado Incluir otros proyectos de UModel.

Abrir en forma de proyecto
Este comando abre el subproyecto seleccionado como un proyecto nuevo.

Borrar mensajes
Este comando borra los mensajes, errores y advertencias de revisión de sintaxis y de combinación de código de la ventana Mensajes.

Nota: los errores informan de problemas que deben solucionarse inmediatamente para poder generar código o para poder actualizar el código del modelo. Las advertencias, sin embargo, se pueden dejar para más tarde. En UModel los errores y advertencias los genera la función de revisión de sintaxis, el compilador de cada lenguaje de programación, el analizador de UModel que lee los archivos fuente generados y la función de importación XMI.

Generar documentación
Este comando sirve para generar documentación para el archivo activo en formato HTML, Word y RTF. (Consulte el apartado Generar documentación UML para obtener más información).

Mostrar elementos no utilizados en ningún diagrama
Este comando crea una lista con todos los elementos que no se utilizan en ningún diagrama del proyecto.

Mostrar paquetes compartidos
Este comando crea una lista con todos los paquetes compartidos del proyecto actual.

Mostrar paquetes incluidos
Este comando crea una lista con todos lo paquetes incluidos en el proyecto actual. Los paquetes `Java Profile.ump` y `Java Lang.ump` están incluidos automáticamente en el proyecto de muestra `Bankview` que viene con UModel.

14.4 Menú Diseño

Los comandos del menú **Diseño** sirven para alinear y poner en fila los elementos de los diagramas de modelado.

Cuando utilice el recuadro de selección (arrastrando el puntero del ratón mientras pulsa el botón principal del ratón) para seleccionar elementos, el elemento con una línea de puntos como contorno es el elemento activo (es decir, el último elemento seleccionado por el recuadro). Los comandos de alineación toman este elemento como referencia/base.

Alinear
Este grupo de comandos sirve para alinear los elementos de modelado con el borde elegido o en el centro, dependiendo de la opción elegida.

Espaciar uniformemente
Este grupo de comandos sirve para espaciar los elementos seleccionados uniformemente, en horizontal o en vertical.

Igualar tamaño
Este grupo de comandos sirve para ajustar el ancho y el alto de los elementos seleccionados al ancho/alto del elemento activo.

Poner en fila
Este grupo de comandos sirve para poner en fila vertical u horizontal los elementos seleccionados.

Estilo de la línea
Este grupo de comandos permite elegir el tipo de línea que se utiliza para conectar los diferentes elementos de modelado. Las líneas pueden ser líneas de dependencia o de asociación.

Tamaño automático
Este comando ajusta el tamaño de los elementos seleccionados hasta alcanzar un tamaño óptimo.

Aplicar diseño automático a todo
Este comando sirve para elegir el tipo de presentación para los elementos de modelado del diagrama UML activo:

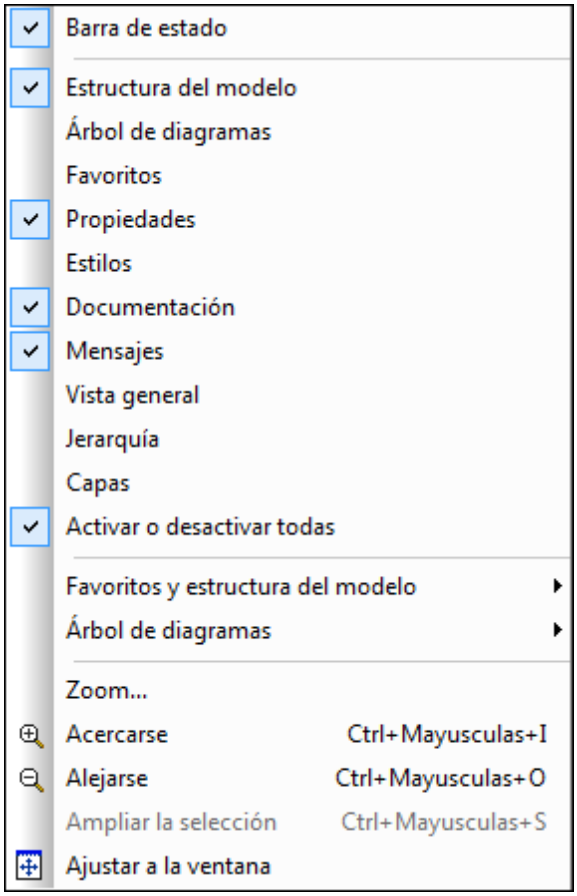
- **Diseño dirigido por fuerzas**
  Organiza los elementos de forma céntrica.
- **Diseño jerárquico**
  Organiza los elementos en función de las relaciones que existen entre ellos.
- **Diseño por bloques**
  Organiza los elementos según su tamaño y formando un rectángulo.

Ajustar la posición de las etiquetas de texto
Este comando pone el nombre de los elementos (de todos o de los seleccionados) en su posición predeterminada.

14.5 Menú Vista

Los comandos de este menú sirven para:

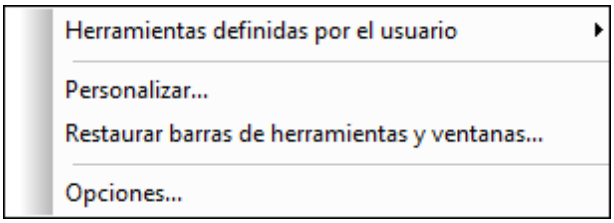
- Activar las pestañas de los diferentes paneles.
- Definir el criterio de ordenación de los elementos de modelado en las pestañas *Estructura del modelo* y *Favoritos*.
- Definir el criterio de agrupación de los diagramas en la pestaña *Árbol de diagramas*.
- Mostrar/ocultar determinados elementos UML en las pestañas *Estructura del modelo* y *Favoritos*.
- Definir el factor de zoom del diagrama actual.



14.6 Menú Herramientas

Los comandos del menú **Herramientas** sirven para:

- Personalizar la aplicación definiendo barras de herramientas, teclas de acceso rápido, menús y macros personales.
- Restaurar las barras de herramientas y ventanas de la aplicación a su estado predeterminado de instalación.
- Definir opciones de configuración globales para la aplicación.



14.6.1 Herramientas definidas por el usuario

Al hacer clic en el comando **Herramientas definidas por el usuario** aparece un submenú con comandos hechos a medida que usan aplicaciones externas. Para crear estos comandos, use la pestaña *Herramientas* del cuadro de diálogo "Personalizar". Al hacer clic en uno de estos comandos personalizados, se ejecuta la acción asociada al comando.

14.6.2 Personalizar

El comando **Herramientas | Personalizar** abre el cuadro de diálogo "Personalizar", que sirve para personalizar varios aspectos de la interfaz gráfica de UModel.

El cuadro de diálogo "Personalizar" está compuesto por estas pestañas:

Comandos
Barras de herramientas
Herramientas
Teclado
Menú
Opciones

14.6.2.1 Comandos

En la pestaña *Comandos* puede personalizar sus menús y barras de herramientas y añadir comandos a los menús y a las barras de herramientas, dependiendo de lo que necesite. No obstante, tenga en cuenta que no puede crear comandos ni menús nuevos para la aplicación.

Para añadir un comando a una barra de herramientas o menú:

1. Seleccione el comando **Herramientas | Personalizar**. Se abre el cuadro de diálogo "Personalizar".

2. Seleccione la pestaña ***Comandos***. En el cuadro de lista *Categorías* seleccione la opción **Todos los comandos**. Todos los comandos disponibles aparecen en el cuadro de lista *Comandos*.
3. Haga clic en un comando del cuadro de lista *Comandos* y arrástrelo a un menú o barra de herramientas ya existente. Al pasar el puntero por encima de una posición donde se puede colocar el comando aparece el icono **I**.
4. Cuando encuentre la posición donde desea colocar el comando, suelte el botón del ratón.

Tenga en cuenta que:

- Mientras arrastra el comando, aparece un pequeño botón al final del puntero del ratón. Esto indica que el comando está siendo arrastrado.
- Si el comando no se puede colocar en la posición actual del cursor, debajo del puntero aparece una **X**.
- Si el cursor está en una posición donde se puede colocar el comando (en una barra de herramientas o en un menú), la **X** desaparece y el icono **I** indica que la posición es válida.
- Los comandos se pueden colocar en menús o barras de herramientas. Si creó una barra de herramientas nueva, puede usar este mecanismo de personalización para rellenar la barra de herramientas con comandos.
- Si pasa el cursor por un menú que está cerrado, el menú se abre y puede insertar el comando en cualquier parte del menú.

Para agregar comandos a menús contextuales

También puede añadir comandos a menús contextuales arrastrando comandos del cuadro de lista *Comandos* hasta el menú contextual:

1. Haga clic en la pestaña *Menú* del cuadro de diálogo "Personalizar".
2. En el cuadro combinado del panel *Menús contextuales* seleccione un menú contextual. El menú contextual seleccionado aparece en pantalla.
3. Vuelva a la pestaña *Comandos* del cuadro de diálogo "Personalizar".
4. Seleccione un comando en el cuadro de lista *Comandos* y arrástrelo hasta la posición deseada del menú contextual.

Para eliminar un comando o menú

Por último, puede eliminar un comando de un menú, menú contextual (ver párrafo anterior) o barra de herramientas o eliminar un menú entero:

1. Abra el cuadro de diálogo "Personalizar" (**Herramientas | Personalizar**).
2. Seleccione cualquier pestaña del cuadro de diálogo "Personalizar". Haga clic con el botón derecho en un menú o comando de menú y seleccione **Eliminar** en el menú contextual que aparece. Si lo prefiere, también puede arrastrar el menú o comando de menú hasta que aparezca el icono **X** debajo del puntero del ratón y suelte el menú o comando de menú. Como resultado se elimina el menú o comando de menú.

Para volver a instalar los comandos de menú eliminados, utilice los mecanismos descritos en este apartado. Para restablecer un menú eliminado, seleccione **Herramientas | Personalizar | Menú** y pulse el botón **Restaurar** del panel *Menús del marco de la aplicación*. Otra opción es seleccionar **Herramientas | Personalizar | Barras de herramientas**, hacer clic en la barra de herramientas pertinente y pulsar el botón **Restaurar**.

14.6.2.2 Barras de herramientas

En la pestaña *Barras de herramientas* puede: (i) activar o desactivar barras de herramientas (es decir, decidir qué barras de herramientas aparecen en la interfaz), (ii) definir qué iconos aparecen en cada barra de herramientas y (iii) crear barras de herramientas personalizadas.

Las barras de herramientas incluyen iconos para los comandos de menú más utilizados. Además, al pasar el puntero sobre un icono, se ofrece información rápida sobre el icono en un mensaje emergente y en la barra de estado de la aplicación. Las barras de herramientas se pueden colocar en cualquier posición de la pantalla, donde aparece como ventana flotante.

Para activar/desactivar una barra de herramientas:
Marque la casilla de la barra de herramientas en el cuadro de lista *Barras de herramientas*.

Para aplicar los cambios a todas las vistas:
Marque la casilla situada al final de la pestaña. De lo contrario, los cambios realizados afectan solamente a la vista activa.
Recuerde que los cambios realizados **después** de marcar la casilla *Aplicar cambios en todas las vistas* afectarán a todas las vistas.

Para añadir una barra de herramientas nueva:
1. Pulse el botón **Nueva...** y escriba el nombre de la barra de herramientas nuevas en el cuadro de diálogo "Nombre de la barra de herramientas" que aparece.
2. Arrastre comandos desde la pestaña *Comandos* hasta la barra de herramientas nueva.

Para cambiar el nombre de una barra de herramientas nueva:
1. Seleccione la barra de herramientas en el panel *Barra de herramientas* y pulse el botón **Cambiar de nombre**.
2. Edite el nombre en el cuadro de diálogo "Nombre de la barra de herramientas" que aparece.

Para restaurar la barra de menús:
Seleccione `Barra de menús` en el panel *Barras de herramientas* y pulse el botón **Restaurar**.
La barra de menús vuelve a su estado original de instalación.

Para restaurar todas las barras de herramientas y comandos de menú:
Pulse el botón **Restaurar todo**.
Todas las barras de herramientas y menús vuelven a su estado original de instalación.

Para eliminar una barra de herramientas:
Seleccione la barra de herramientas en el panel *Barras de herramientas* y pulse el botón **Eliminar**.

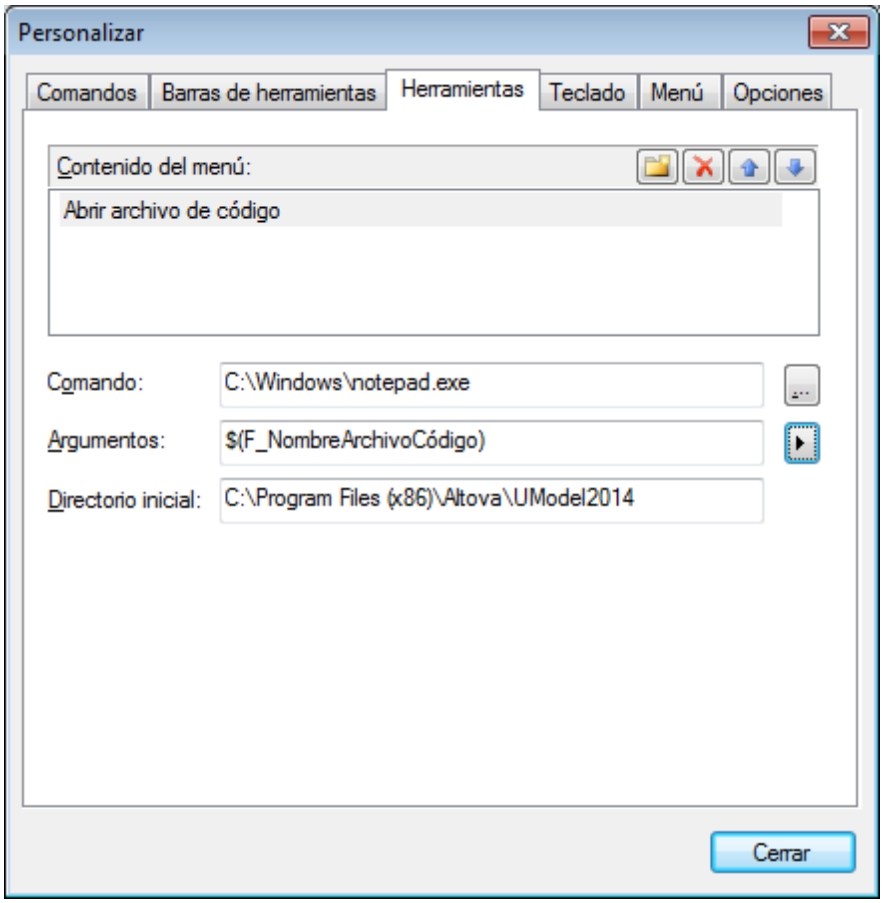
Para mostrar las etiquetas de texto de una barra de herramientas:
Seleccione la barra de herramientas y marque la casilla *Mostrar etiquetas de texto*.
Recuerde que debe activar las etiquetas de texto de cada barra de herramientas por separado.

14.6.2.3 Herramientas

En la pestaña *Herramientas* puede crear comandos para poder usar aplicaciones externas desde UModel. Estos comandos se añaden al menú **Herramientas | Herramientas definidas por el**

usuario.

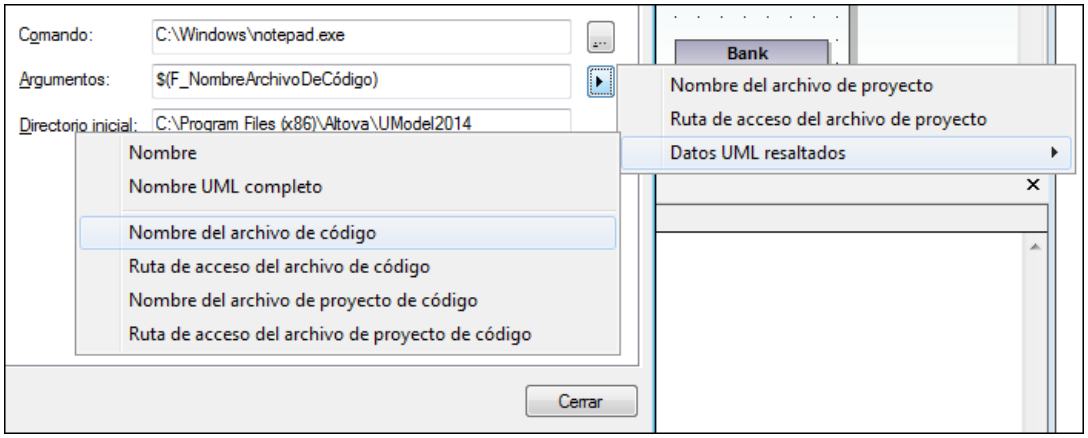
Haga clic en el icono en forma de carpeta (de la barra de iconos *Contenido del menú)* para añadir una entrada nueva y use el campo *Comando* para asociarlo a una aplicación.



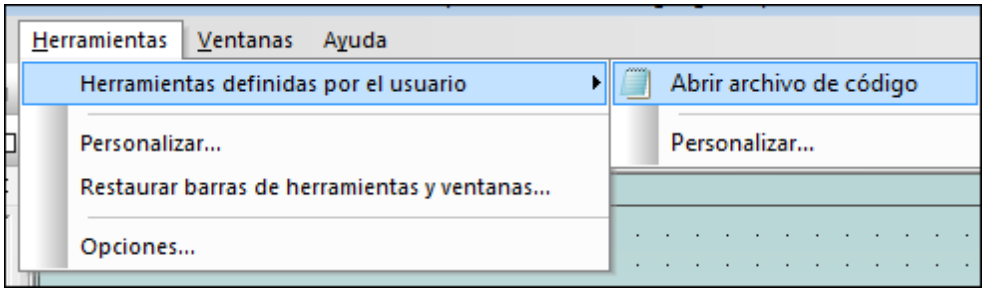
La pestaña *Herramientas* también permite definir argumentos. Los argumentos son variables a las que se asignan valores concretos cuando se inicia una herramienta externa desde el comando de menú.

P. ej. supongamos que queremos abrir en Notepad el archivo de código fuente de la clase UML seleccionada.

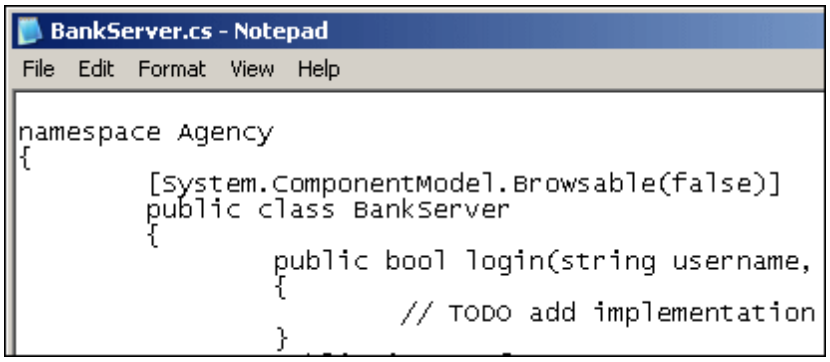
1. Haga clic en **Herramientas | Personalizar** y abra la pestaña *Herramientas* del cuadro de diálogo "Personalizar".
2. Escriba el nombre y la ruta de acceso de la aplicación externa (p. ej. `C:\...\notepad.exe`).
3. Haga clic en el botón flotante del campo *Argumentos* y seleccione el argumento que desea utilizar (p. ej. *Nombre del archivo de código*).



4. Haga clic en el botón **Cerrar** para finalizar.
5. Haga clic en el menú **Herramientas | Herramientas definidas por el usuario** y seleccione **Abrir archivo de código**.



Se abre el archivo `BankServer.cs` en Notepad.



Argumentos de UModel

Nombre del archivo de proyecto

El nombre de archivo del archivo de proyecto de UModel activo (p. ej. `Prueba.ump`).

Ruta de acceso del archivo de proyecto

La ruta de archivo absoluta del archivo de proyecto de UModel activo (p. ej. `c:\MiDirectorio\Prueba.ump`).

Datos UML resaltados – Nombre

El nombre del elemento UML resaltado (p. ej. `Clase1`).

Datos UML resaltados – Nombre UML completo
El nombre completo del elemento UML resaltado (p. ej. `Paquete1::Paquete2::Clase1`).

Datos UML resaltados – Nombre del archivo de código
El nombre del archivo de código de la clase, interfaz o enumeración UML resaltada, tal y como aparece en la ventana Propiedades (p. ej. `Clase1.cs` o `MiEspacioDeNombres\Clase1.Java`).

Datos UML resaltados – Ruta de acceso del archivo de código
La ruta de acceso del archivo de código de la clase, interfaz o enumeración UML resaltada, tal y como aparece en la ventana Propiedades (p. ej. `C:\Temp\MiCódigo\Clase1.cs`).

Datos UML resaltados – Nombre del archivo de proyecto de código
El nombre de archivo del proyecto de código al que pertenece la clase, interfaz o enumeración UML resaltada.

El nombre de archivo del proyecto puede ser relativo al archivo de proyecto de UModel y es el mismo que aparece en la ventana Propiedades del componente (p. ej. `C:\Temp\MiCódigoFuente\MiProyecto.vcproj` o `MiCódigoFuente\MiProyecto.vcproj`)

Datos UML resaltados – Ruta de acceso del archivo de proyecto de código
La ruta de acceso de archivo del proyecto de código al que pertenece la clase, interfaz o enumeración UML resaltada (p. ej. `c:\Temp\MiCódigoFuente\MiProyecto.vcproj`").

14.6.2.4 Teclado

En la pestaña *Teclado* puede crear teclas de acceso rápido nuevas o cambiar las teclas de acceso rápido ya existentes para cualquier comando de la aplicación.

Para asignar una tecla de acceso rápido nueva a un comando o cambiar una tecla de acceso rápido ya existente:

1. En el cuadro combinado *Categoría* seleccione la opción *Todos los comandos*.
2. En el cuadro de lista *Comandos* seleccione el comando al que desea asignar una tecla de acceso rápido nueva o el comando cuya tecla de acceso rápido desea cambiar.
3. Haga clic dentro del cuadro *Pulsar tecla de acceso rápido nueva* y pulse la tecla de acceso rápido que desea asignar al comando. La tecla de acceso rápido aparece en el cuadro *Pulsar tecla de acceso rápido nueva.* Si la tecla de acceso rápido no se asignó todavía a ningún comando, se habilita el botón **Asignar**. Si la tecla ya se asignó a un comando, el comando aparece debajo del cuadro y el botón **Asignar** está deshabilitado. (Para borrar el contenido del cuadro *Pulsar tecla de acceso rápido nueva* pulse **Ctrl**, **Alt** o **Mayús**).
4. Haga clic en el botón **Asignar**. La tecla de acceso rápido aparece ahora en el cuadro de lista *Teclas actuales*. Puede asignar varias teclas de acceso rápido al mismo comando si lo desea.
5. Para confirmar los cambios pulse el botón **Cerrar**.

Para eliminar una tecla de acceso rápido:

1. En el cuadro de lista *Teclas actuales* seleccione la tecla de acceso rápido que desea

eliminar.
2. Pulse el botón **Quitar**.
3. Para confirmar los cambios pulse el botón **Cerrar**.

14.6.2.5 Menú

En la pestaña *Menú* puede personalizar las dos barras de menú principales (la barra de menú predeterminada y la barra de menú de la aplicación) así como los menús contextuales de la aplicación.

Para personalizar la barra de menú predeterminada y la barra de menú de la aplicación:

La barra de menú predeterminada es la barra de menú que aparece cuando no hay ningún documento abierto en la ventana principal. La barra de menú de la aplicación es la barra que aparece cuando hay un documento abierto en la ventana principal. Cada una de estas barras de menú se puede personalizar y los cambios realizados en una de las barras de menú no afecta a la otra.

Para personalizar una barra de menú, selecciónela en el cuadro combinado *Mostrar menús para:* de la pestaña *Menú* (*imagen anterior*). Después cambie a la pestaña *Comandos* del cuadro de diálogo "Personalizar" y arrastre comandos desde el cuadro de lista *Comandos* hasta la barra de menú.

Para eliminar comandos de menús y restaurar las barras de menú:

Para eliminar un menú entero o un comando de un menú, seleccione el menú o el comando de menú y después (i) haga clic con el botón derecho y seleccione **Eliminar** o (ii) arrastre el comando fuera de la barra de menú o del menú.

Para restaurar estas dos barras de menú (la barra de menú predeterminada y la de la aplicación) a su estado original de instalación seleccione el menú en el cuadro combinado *Mostrar menús para:* y haga clic en el botón **Restaurar** situado bajo el cuadro combinado.

Para personalizar los menús contextuales de la aplicación:

Los menús contextuales son los menús que aparecen cuando se hace clic con el botón derecho en determinados objetos de la interfaz de la aplicación. Siga estos pasos para personalizar un menú contextual:

1. Seleccione el menú contextual en el cuadro combinado *Seleccionar menú contextual*. Aparece el menú contextual.
2. Pase a la pestaña *Comandos* del cuadro de diálogo Personalizar.
3. Arrastre un comando del cuadro de lista *Comandos* al menú contextual.
4. Si desea eliminar un comando del menú contextual, haga clic en él con el botón derecho y seleccione **Eliminar**. También puede seleccionar el comando y arrastrarlo fuera del menú contextual.

Para restaurar un menú contextual a su estado original de instalación seleccione el menú en el cuadro combinado *Seleccionar menú contextual* y después pulse el botón **Restaurar**, situado bajo el cuadro combinado.

Sombras de menú

Marque la casilla *Sombras de menú* para dar sombra a todos los menús.

14.6.2.6 Opciones

En la pestaña *Opciones* puede configurar varias opciones generales del entorno.

Si marca la opción *Mostrar información en pantalla en las barras de herramientas*, cuando pase el puntero sobre los botones de las barras de herramientas aparecerá una etiqueta con información. La etiqueta incluirá una descripción breve de la función del botón.

Si marca la opción *Mostrar teclas de acceso rápido en la información en pantalla*, la etiqueta de información rápida incluirá la tecla de acceso rápido asociada al botón (siempre y cuando se asignara uno).

Si marca la opción *Iconos grandes*, la interfaz gráfica mostrará los iconos en un tamaño más grande.

14.6.3 Restaurar barras de herramientas y ventanas

El comando **Restaurar barras de herramientas y ventanas** cierra UModel y lo reinicia con su configuración predeterminada. Antes de cerrarse, UModel le pregunta si desea cerrar o no la aplicación.

Este comando es muy práctico si movió ventanas o barras de herramientas de sitio, si las ocultó o si ajustó su tamaño y desea poner todas estas barras de herramientas y ventanas como estaban en un principio.

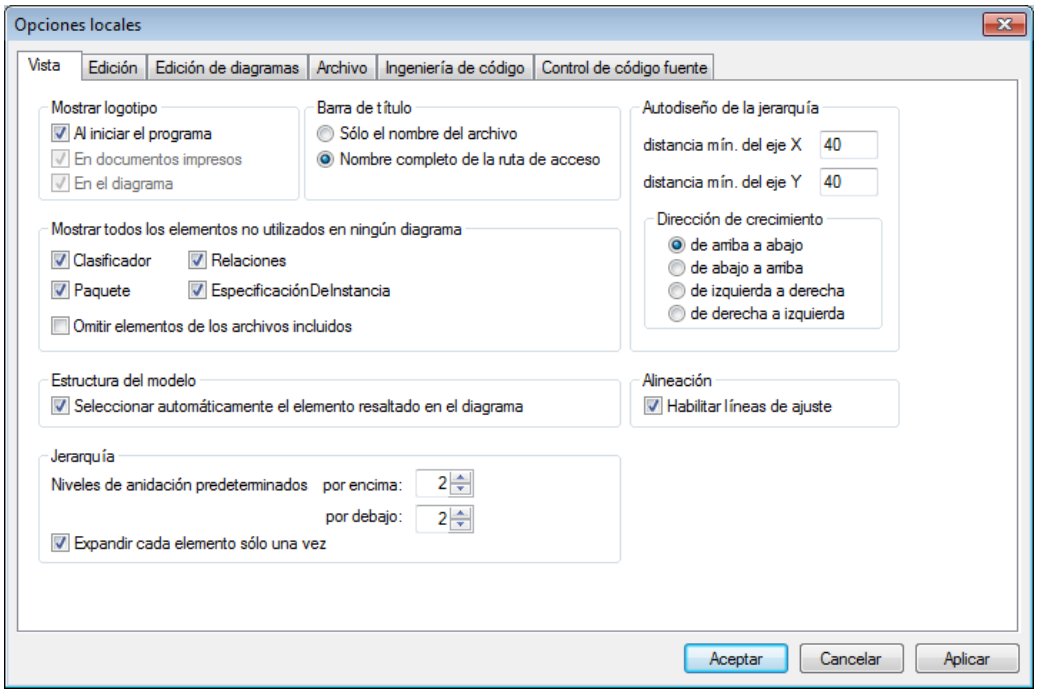
14.6.4 Opciones

El comando **Herramientas | Opciones** abre el cuadro de diálogo "Opciones locales".

Pestaña *Vista*
En esta pestaña puede definir:

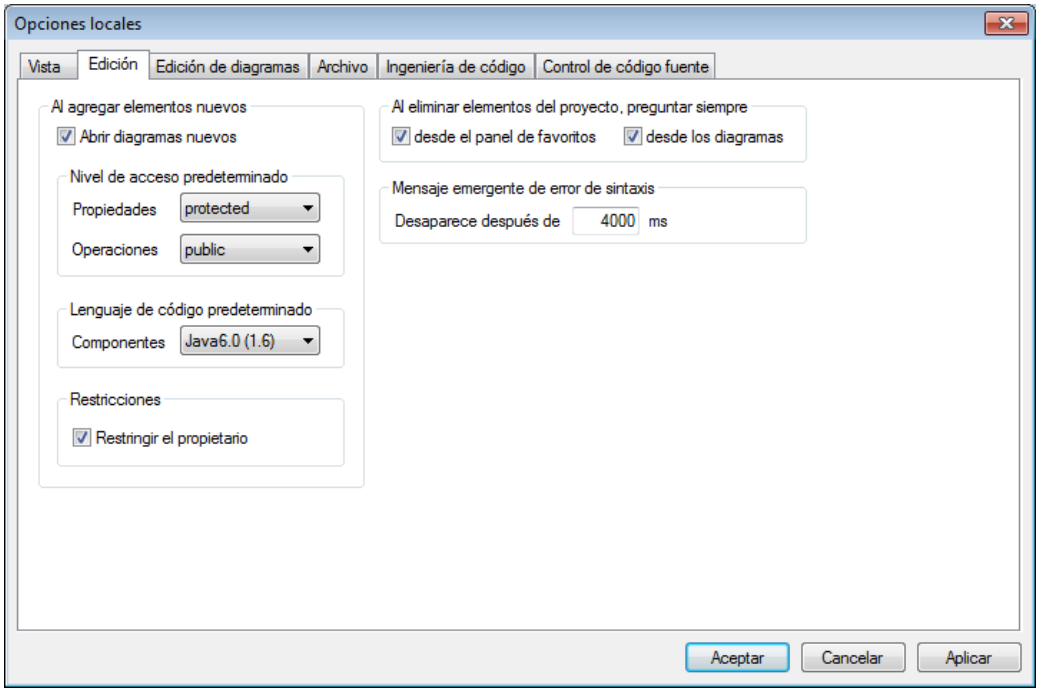
- donde debe aparecer el logotipo del programa.
- el contenido de la barra de título de la aplicación.
- el tipo de elementos que deben aparecer en la lista generada por el comando *Mostrar elementos no utilizados en ningún diagrama* del menú contextual de los paneles Estructura del modelo y Favoritos. También tiene la opción de omitir los elementos de los archivos incluidos.
- si un elemento seleccionado de un diagrama se selecciona/sincroniza automáticamente o no en la Estructura del modelo.
- la profundidad predeterminada de la vista jerárquica generada por el comando **Mostrar en forma de diagrama** en el panel Jerarquía.
- la profundidad de los niveles del panel Jerarquía (con las opciones del grupo *Autodiseño de la jerarquía*).

- que se expanda solo uno de los clasificadores del mismo tipo de la misma imagen / del mismo diagrama (con la opción *Expandir cada elemento solo una vez*).
- si se habilitan las líneas de ajuste para ayudarle durante la alineación de elementos en el diagrama.



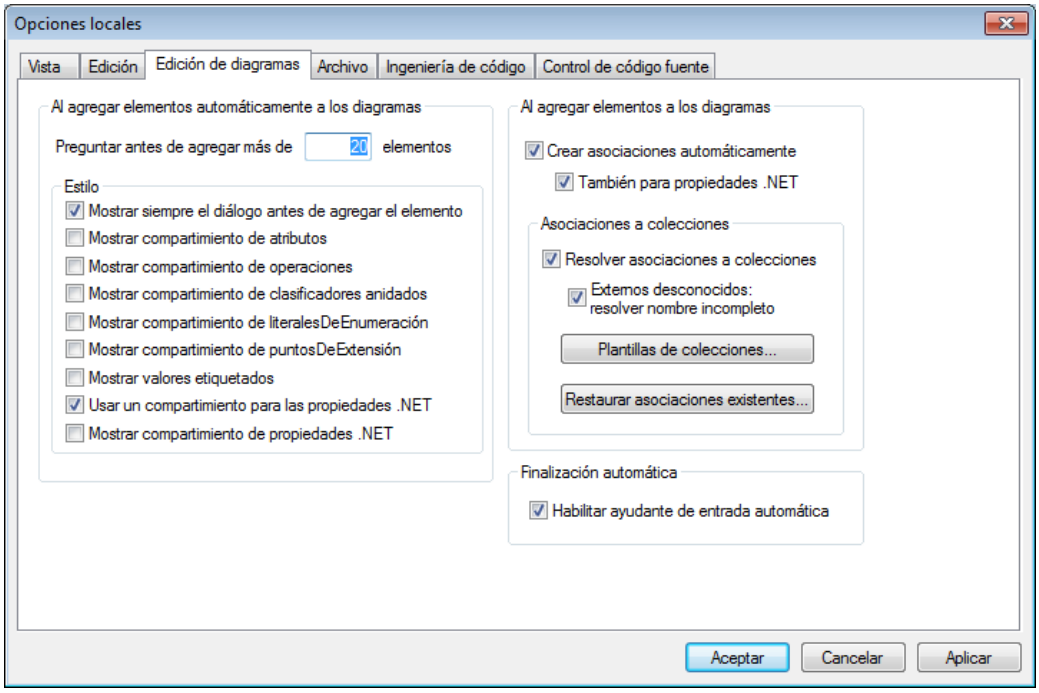
Pestaña *Edición*
En esta pestaña puede definir:

- que cuando se cree un diagrama nuevo en la Estructura del modelo, el diagrama se abra automáticamente en el área de trabajo.
- el nivel de acceso predeterminado de los elementos nuevos (propiedades y operaciones).
- el lenguaje de código predeterminado para los componentes nuevos.
- si las restricciones también deben restringir automáticamente su propietario.
- si debe aparecer un aviso cuando se **eliminen** elementos del proyecto, del panel Favoritos o de un diagrama. Este aviso se puede desactivar cuando se eliminen elementos.
- cuánto tardan en cerrarse los mensajes emergentes de error de sintaxis.


Opciones locales
Vista
Edición
Edición de diagramas
Archivo
Ingeniería de código
Control de código fuente
Al agregar elementos nuevos
Abrir diagramas nuevos
Nivel de acceso predeterminado
Propiedades
protected
Operaciones
public
Lenguaje de código predeterminado
Componentes
Java6.0 (1.6)
Restricciones
Restringir el propietario
Al eliminar elementos del proyecto, preguntar siempre
desde el panel de favoritos
desde los diagramas
Mensaje emergente de error de sintaxis
Desaparece después de
4000
ms
Aceptar
Cancelar
Aplicar

Pestaña *Edición de diagramas*:
En esta pestaña puede definir:

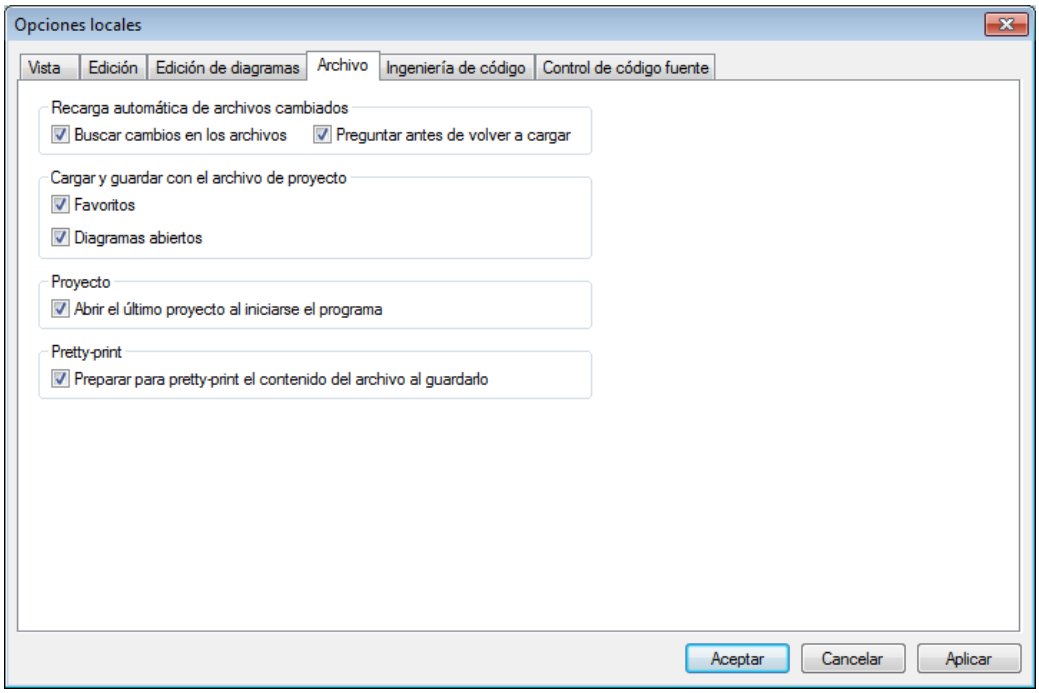
- cuántos elementos se pueden añadir automáticamente a un diagrama sin que aparezca un aviso.
- la presentación de los estilos cuando se añaden automáticamente a un diagrama.
- si se debe crear automáticamente asociaciones entre los elementos de modelado cuando se añaden elementos a un diagrama.
- si se deben resolver las asociaciones a colecciones.
- si las plantillas de elementos externos desconocidos se deben resolver como plantillas no completas.
- o si se deben usar plantillas de colecciones ya disponibles o definir plantillas nuevas. Debe definir las plantillas de colecciones como plantillas completas (es decir, `a.b.c.Lista`). Si la plantilla tiene este espacio de nombres, UModel crea una asociación de colecciones automáticamente. Excepción: si la plantilla pertenece al paquete **`Elementos externos desconocidos`** y está habilitada la opción Externos desconocidos: resolver nombre incompleto, entonces se tiene en cuenta el nombre de la plantilla solamente (es decir, `Lista` en lugar de `a.b.c.Lista`).
- si la ventana de finalización automática está disponible mientras se editan atributos u operaciones en los diagramas de clases.



Pestaña *Archivo*:
En esta pestaña puede definir:

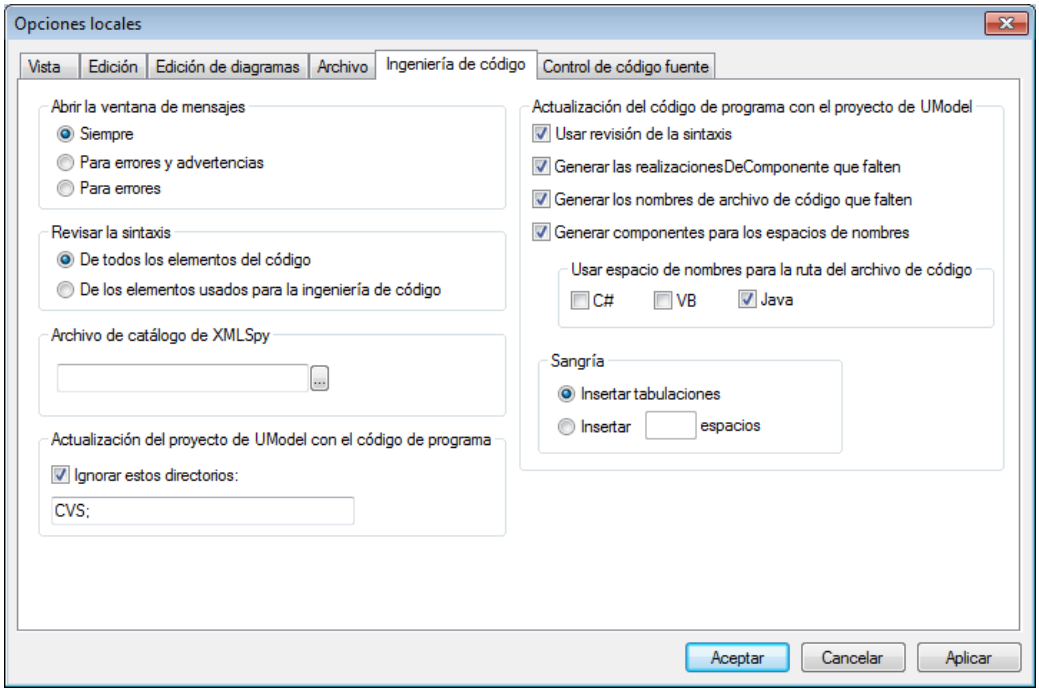
- qué ocurre cuando los archivos cambian.
- si el contenido del panel Favoritos y los diagramas abiertos deben cargarse y guardarse con el proyecto actual.

- si el último proyecto que se abrió se debe abrir automáticamente cuando se inicia la aplicación.
- si se añaden retornos de carro/saltos de línea y tabulaciones al archivo de proyecto para darle formato **pretty-print**.



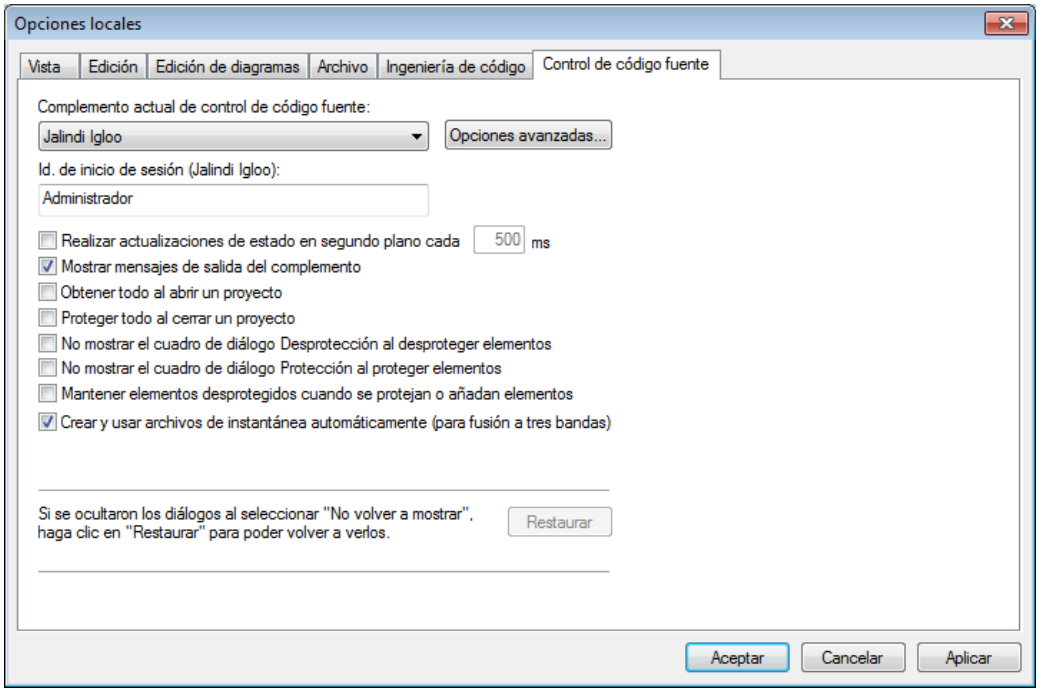
Pestaña *Ingeniería de código*:
En esta pestaña puede definir:

- en qué circunstancias se abre la ventana Mensajes.
- si se revisan **todos los elementos** de código (es decir, los que están en una raíz de espacio de nombres Java / C# / VB y los que están asignados a un componente Java / C# / VB) o **solo los elementos utilizados para ingeniería de código** (es decir, los que tienen activa la casilla `Usar para ingeniería de código`).
- si durante la actualización del código de programa:
  - se revisa la sintaxis o no.
  - se generan automáticamente las `RealizacionesDeComponente` que faltan o no.
  - se generan los nombres de archivo de código que faltan en el código combinado o no.
  - se utilizan espacios de nombres en la ruta de acceso del archivo de código o no.
- el tipo de sangría que se aplica al código (es decir, tabulaciones o espacios).
- qué directorios se omiten cuando se actualice un proyecto de UModel con código. Separe los directorios con un punto y coma. Los directorios secundarios que se llamen igual también se pasarán por alto.
- la ubicación del archivo de catálogo de XMLSpy `RootCatalog.xml`, que permite a UModel y a XMLSpy recuperar esquemas, hojas de estilos y otros archivos utilizados con frecuencia desde carpetas de usuario locales. Esto aumenta la velocidad de procesamiento y permite al usuario trabajar sin conexión.



Pestaña *Control de código fuente*:
En esta pestaña puede elegir:

- el complemento de control de código fuente actual. Con el botón **Opciones avanzadas** puede configurar algunas opciones avanzadas del sistema de control de código seleccionado. Estas opciones avanzadas dependen del control de código fuente elegido.
- el id. de inicio de sesión para el proveedor de control de código fuente.
- otras opciones relacionadas con la protección y desprotección de archivos.
- el botón **Restaurar** se habilita si el usuario marca la casilla *No volver a mostrar* en un cuadro de diálogo. Con el botón **Restaurar** puede volver a habilitar los avisos.



14.7 Menú Ventanas

En cascada:
Este comando reorganiza todos las ventanas de documento que están abiertas en forma de **cascada** (es decir, las ventanas se apilan una encima de otra).

En mosaico horizontal:
Este comando reorganiza todas las ventanas de documento que están abiertas en forma de **mosaico horizontal** (es decir, se pueden ver todas las ventanas a la vez y se distribuyen de forma horizontal).

En mosaico vertical:
Este comando reorganiza todas las ventanas de documento que están abiertas en forma de **mosaico vertical** (es decir, se pueden ver todas las ventanas a la vez y se distribuyen de forma vertical).

Ordenar iconos:
Este comando reorganiza los elementos que estén dispersos por el diagrama y los coloca en la parte inferior del área de trabajo.

Cerrar:
Este comando cierra la pestaña del diagrama activo.

Cerrar todas:
Este comando cierra todas las pestañas de diagrama que están abiertas.

Cerrar ventanas inactivas:
Este comando cierra todas las pestañas de diagrama que están abiertas excepto la pestaña activa.

Adelante:
Este comando abre la siguiente pestaña de diagrama o el siguiente elemento con hipervínculo.

Atrás:
Este comando abre la pestaña de diagrama anterior o el elemento con hipervínculo anterior.

Lista de ventanas abiertas:
Esta lista muestra todas las ventanas que están abiertas en cada momento y permite cambiar de una ventana a otra rápidamente.
También puede usar las teclas de acceso rápido **Ctrl+Tabulador** o **Ctrl+F6** para recorrer todas las ventanas que están abiertas.

14.8 Menú Ayuda

Contenido

Descripción

Abre la ayuda en pantalla de UModel por la tabla de contenido, que aparece en el panel izquierdo de la ventana de ayuda. Esta tabla de contenido ofrece un resumen de la documentación. Haga clic en una entrada de la tabla para abrir la sección correspondiente de la documentación.

Índice

Descripción

Abre la ayuda en pantalla de UModel por el índice de palabras clave, que aparece en el panel izquierdo de la ventana de ayuda. Este índice enumera todas las palabras clave de la ayuda y permite navegar a un tema con solo hacer doble clic en la palabra clave correspondiente. Si una palabra clave está asociada a varios temas, la ventana de ayuda muestra todos estos temas en pantalla.

Buscar

Descripción

Abre la ayuda en pantalla de UModel por la función de búsqueda, que aparece en el panel izquierdo de la ventana de ayuda. Para buscar un término introduzca el término de búsqueda en el campo de consulta y pulse **Entrar**. El sistema de ayuda realiza una búsqueda en toda la documentación y devuelve una lista de resultados. Haga doble clic en una entrada para abrir la sección correspondiente de la documentación.

Activación del software

Descripción

Tras descargar el producto de software de Altova puede desactivarlo con una clave de evaluación gratuita o con una clave de licencia permanente.

- ***Clave de evaluación gratuita:*** cuando inicie el software por primera vez aparecerá el cuadro de diálogo "Activación del software". Este cuadro de diálogo incluye un botón para solicitar un código clave de evaluación gratuita. Introduzca su nombre, el nombre de su compañía y su dirección de correo electrónico y haga clic en **Enviar solicitud**. Altova le enviará la clave de evaluación al correo electrónico proporcionado. Ahora introduzca el código clave en el cuadro de diálogo "Activación del software" y haga clic en **Aceptar**

para empezar a trabajar con el software, que permanecerá desbloqueado durante 30 días.

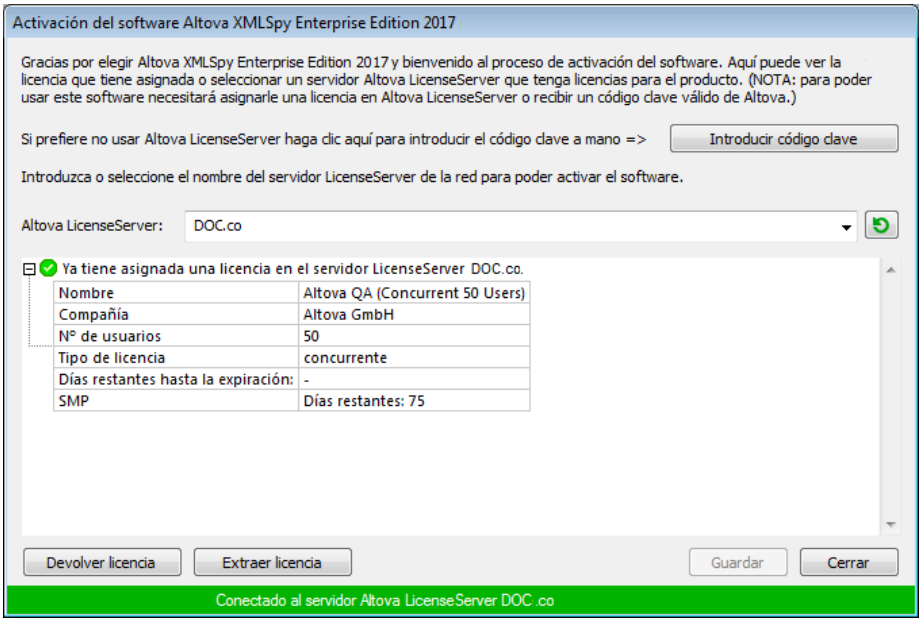
- ***Clave de licencia permanente:*** el cuadro de diálogo "Activación del software" también incluye un botón para comprar una clave de licencia permanente. Este botón conduce a la tienda en línea de Altova, donde podrá adquirir una clave de licencia permanente para el producto. Altova ofrece licencias para un solo usuario y licencias para varios usuarios (ambos tipos estarán en el correo electrónico que reciba de Altova). Las licencias para un solo usuario contienen los datos de la licencia, su nombre, el nombre de su compañía, su correo electrónico y un código clave. Las licencias para varios usuarios contienen los datos de la licencia, el nombre de su compañía y un código clave. Recuerde que el contrato de licencia prohíbe instalar más copias de las permitidas por la licencia adquirida. Recuerde introducir los datos que solicita el cuadro de diálogo de registro tal y como aparecen en el correo electrónico que recibió con las licencias.

Nota: recuerde que los datos de la licencia en el cuadro de diálogo "Activación del software" deben ser idénticos a los que aparecen en el correo electrónico que recibió con las licencias. En el caso de licencias para varios usuarios, cada usuario debe introducir su nombre en el campo *Nombre*.

El cuadro de diálogo "Activación del software" (*imagen siguiente*) se abre con el comando **Ayuda | Activación del software**.

Hay dos maneras de activar el software:

- Introduciendo la información de la licencia (con el botón **INTRODUCIR código clave...**) o
- Adquiriendo una licencia a través de un servidor Altova LicenseServer de la red (haciendo clic en el botón **Usar Altova LicenseServer**, situado al final del cuadro de diálogo). Para ello es necesario que el servidor LicenseServer tenga una licencia para su producto en el repositorio de licencias. Si así es, el cuadro de diálogo "Activación del software" emite un mensaje a tal efecto (*imagen siguiente*) y basta con hacer clic en el botón **Guardar** para adquirir la licencia.



Observe que, una vez adquirida, la licencia no se puede devolver al servidor LicenseServer hasta 7 días después. Transcurridos estos 7 días podrá devolver la licencia (con el botón **Devolver licencia**) para que pueda ser adquirida por otro cliente. No obstante, el administrador de LicenseServer puede anular asignaciones de licencias desde la interfaz web de LicenseServer en cualquier momento.

Extracción de licencias

Puede extraer una licencia del repertorio durante un período máximo de 30 días de modo que la licencia se almacene en el equipo donde se ejecuta el producto. Esto le permitirá trabajar sin conexión a Internet, lo cual puede ser útil si desea trabajar en un entorno que no dispone de acceso a su servidor Altova LicenseServer (p. ej. cuando el producto servidor de Altova está instalado en un equipo portátil y el usuario se encuentra de viaje). Mientras la licencia esté extraída, LicenseServer indicará que la licencia está en uso y no podrá ser utilizada por ningún otro equipo. La licencia vuelve automáticamente a su estado insertado cuando finaliza el período de extracción de la licencia. La licencia extraída también se puede insertar en el servidor en cualquier momento con el botón **Insertar** del cuadro de diálogo "Activación del software".

Siga estas instrucciones para extraer una licencia:

1. En el cuadro de diálogo "Activación del software" haga clic en el botón **Extraer licencia** (*imagen anterior*).
2. Aparece el cuadro de diálogo "Extracción de licencias". Seleccione el periodo de extracción y haga clic en **Extraer**.
3. La licencia se extrae y el cuadro de diálogo "Activación del software" muestra información sobre la extracción de la licencia, incluida la fecha y la hora en la que expira el plazo de extracción. Ahora, en lugar del botón **Extraer licencia**, aparece el botón **Insertar licencia**. Para insertar la licencia basta con hacer clic en este botón. Como la

licencia vuelve automáticamente a su estado de inserción cuando finaliza el plazo de extracción, compruebe que el plazo seleccionado coincide con el período de tiempo que tiene pensado trabajar sin conexión a Internet.

Nota: para poder extraer licencias esta característica debe estar habilitada en el servidor LicenseServer. Si esta característica no está habilitada, recibirá un mensaje de error a tal efecto. Cuando esto ocurra, póngase en contacto con el administrador de su servidor LicenseServer.

Altova LicenseServer es una práctica herramienta para administrar en tiempo real todas las licencias de Altova de la red y ofrece información detallada sobre cada licencia, asignaciones a clientes y uso de las licencias. La ventaja de usar este producto está en sus características administrativas. Altova LicenseServer puede descargarse gratis del sitio web de Altova. Para más información consulte la documentación de Altova LicenseServer.

Formulario de pedido

Descripción

Hay dos maneras de comprar licencias para los productos de Altova: con el botón **Comprar código clave...** del cuadro de diálogo "Activación del software" (ver apartado anterior) o con el comando **Ayuda | Formulario de pedido,** que le lleva directamente a la tienda en línea de Altova.

Registro del software

Descripción

Este comando abre la página de registro de productos de Altova en una pestaña del explorador. Si registra el software, recibirá información sobre actualizaciones y versiones nuevas del producto.

Buscar actualizaciones

Descripción

Comprueba si existe una versión más reciente del producto en el servidor de Altova y emite un mensaje a tal efecto.

Centro de soporte técnico

Descripción

Es un enlace al centro de soporte técnico del sitio web de Altova. El centro de soporte técnico incluye preguntas frecuentes, foros de debate y un formulario para ponerse en contacto con el equipo de soporte técnico de Altova.

Preguntas más frecuentes

Descripción

Es un enlace a la página de preguntas frecuentes del sitio web de Altova. Esta página se actualiza constantemente con las preguntas que recibimos de nuestros clientes.

Descargar herramientas gratis y componentes

Descripción

Es un enlace al centro de descargas de componentes del sitio web de Altova. Aquí puede descargar software adicional para usarlo con los productos de Altova, como procesadores XSLT y XSL-FO y paquetes de integración. Estos componentes suelen ser totalmente gratis.

UModel en Internet

Descripción

Es un enlace al sitio web de Altova, donde encontrará más información sobre UModel, otros productos de Altova y tecnologías relacionadas.

Acerca de UModel

Descripción

Abre la pantalla de presentación de la aplicación, que incluye el número de versión del producto e información sobre copyright.

Chapter 15

Generador de código

15 Generador de código

15.1 Códigos de error

Códigos de error del sistema operativo

201 No se encontró el archivo: "%s"
202 No se puede crear el archivo "%s"
203 No se puede abrir el archivo "%s"
204 No se puede copiar el archivo "%s" en "%s"

Códigos de error de sintaxis

401 Se esperaba una palabra clave
402 Se esperaba "%s"
403 No se especificó el archivo de salida
404 Fin inesperado del archivo
405 Palabra clave no permitida

Códigos de error de tiempo de ejecución

501 Variable desconocida "%s"
502 Redefinición de la variable "%s"
503 La variable "%s" no es un contenedor
504 Propiedad desconocida "%s"
505 No se puede convertir %s en %s
507 Función desconocida
508 Ya se definió esta función
509 Parámetro no válido
510 División por cero
511 Método desconocido
512 Número incorrecto de parámetros
513 Desbordamiento de pila

Chapter 16

Información sobre licencias

16 Información sobre licencias

En esta sección encontrará:

- información sobre la distribución de este producto de software
- información sobre activación del software y medición de licencias
- información sobre derechos de propiedad intelectual de este producto de software
- el contrato de licencia para el usuario final que rige el uso de este producto de software

Los términos del contrato de licencia que aceptó al instalar el producto de software son vinculantes, por lo que rogamos lea atentamente toda esta información.

16.1 Distribución electrónica de software

Este producto está disponible por distribución electrónica de software, un método de distribución que ofrece ventajas únicas:

- Puede evaluar el software de forma totalmente gratuita antes de decidir si compra el producto.
- Si decide comprarlo, puede hacer un pedido en línea en el sitio web de Altova y conseguir en pocos minutos el software con licencia.
- Si realiza el pedido en línea, siempre recibirá la versión más reciente de nuestro software.
- El paquete de instalación del producto incluye un sistema de ayuda en pantalla totalmente integrado. La versión más reciente del manual del usuario está disponible en www.altova.com (i) en formato HTML y (ii) en formato PDF para descargar e imprimir si lo desea.

Período de evaluación de 30 días

Después de descargar el producto de software, puede probarlo de forma totalmente gratuita durante un plazo de 30 días. Pasados unos 20 días, el software empieza a recordarle que no tiene una licencia. El mensaje de aviso aparece una sola vez cada vez que se inicie la aplicación. Para seguir utilizando el programa una vez pasado el plazo de 30 días, deberá comprar una licencia permanente y aceptar el contrato de licencia de software de Altova, que se entrega en forma de código clave. La licencia puede comprarse directamente en la tienda en línea del sitio web de Altova. Después de comprar la licencia recibirá el código clave, que debe introducir en el cuadro de diálogo "Activación del software" para desbloquear el producto de forma permanente.

Distribuir la versión de evaluación a otros usuarios de su organización

Si desea distribuir la versión de evaluación en la red de su compañía o si desea usarlo en un PC que no está conectado a Internet, solamente puede distribuir los programas de instalación (siempre y cuando no se modifiquen de forma alguna). Todo usuario que acceda al instalador debe solicitar su propio código clave de evaluación (de 30 días). Una vez pasado este plazo de 30 días, todos los usuarios deben comprar también una licencia para poder seguir usando el producto.

Para más información consulte el contrato de licencia de software de Altova que aparece al final de esta sección.

16.2 Activación del software y medición de licencias

Durante el proceso de activación del software de Altova, puede que la aplicación utilice su red interna y su conexión a Internet para transmitir datos relacionados con la licencia durante la instalación, registro, uso o actualización del software a un servidor de licencias operado por Altova y para validar la autenticidad de los datos relacionados con la licencia y proteger a Altova de un uso ilegítimo del software y mejorar el servicio a los clientes. La activación es posible gracias al intercambio de datos de la licencia (como el sistema operativo, la dirección IP, la fecha y hora, la versión del software, el nombre del equipo, etc.) entre su equipo y el servidor de licencias de Altova.

Su producto incluye un módulo integrado de medición de licencias que le ayudará a evitar infracciones del contrato de licencia para el usuario final. Puede comprar una licencia de un solo usuario o de varios usuarios para el producto de software y el módulo de medición de licencias se asegura de que no se utiliza un número de licencias mayor al permitido.

Esta tecnología de medición de licencias usa su red de área local (LAN) para comunicarse con las instancias de la aplicación que se ejecutan en equipos diferentes.

Licencia de un solo usuario

Cuando se inicia la aplicación, se inicia el proceso de medición de licencias y el software envía un breve datagrama de multidifusión para averiguar si hay otras instancias del producto activas en otros equipos del mismo segmento de red al mismo tiempo. Si no recibe ninguna respuesta, la aplicación abre un puerto para escuchar a otras instancias de la aplicación.

Licencia de varios usuarios

Si se usa más de una instancia de la aplicación dentro de la misma red LAN, estas instancias se comunicarán entre ellas al iniciarse. Estas instancias intercambian códigos claves para que ayudarle a no sobrepasar por error el número máximo de licencias concurrentes. Se trata de la misma tecnología de medición de licencias que suele utilizarse en Unix y en otras herramientas de desarrollo de bases de datos. Gracias a ella puede comprar licencias de varios usuarios de uso concurrente a un precio razonable.

Las aplicaciones se diseñaron de tal modo que envían pocos paquetes pequeños de red y no cargan demasiado su red. Los puertos TCP/IP (2799) utilizados por su producto de Altova están registrados oficialmente en la IANA (para más información consulte el sitio web de la IANA www.iana.org) y nuestro módulo de medición de licencias es una tecnología probada y eficaz.

Si usa un servidor de seguridad, puede notar las comunicaciones del puerto 2799 entre los equipos que ejecutan los productos de Altova. Si quiere, puede bloquear ese tráfico, siempre y cuando esto no resulte en una infracción del contrato de licencia.

También notará que su producto de Altova ofrece varias funciones prácticas si está conectado a Internet. Estas funciones no tienen nada que ver con la tecnología de medición de licencias.

16.3 Derechos de propiedad intelectual

El software de Altova y sus copias (si tiene permiso de Altova para realizar copias) es propiedad intelectual de Altova y de sus proveedores. La estructura, la organización y el código del software se considera secreto comercial e información confidencial de Altova y de sus proveedores. El software está protegido por las leyes de derechos de autor, como la ley de derechos de autor de EE UU, tratados internacionales y la legislación vigente del país donde se utiliza, entre otras. Altova conserva los derechos de propiedad de todas las patentes, derechos de autor, secretos comerciales, marcas registradas y otros derechos de propiedad intelectual pertenecientes al software y los derechos de propiedad de Altova abarcan también imágenes, fotografías, animaciones, vídeos, audio, música, texto y otros applets incorporados al software y al material impreso que viene con el software. Las notificaciones de infracción de dichos derechos de autor debe enviarse al agente de derechos de autor de Altova, cuyos datos de contacto aparecen en el sitio web de Altova.

El software de Altova contiene software de terceros que también está protegido por las leyes de propiedad intelectual, incluida, entre otras, la legislación de derechos de autor mencionada en http://www.altova.com/es/legal_3rdparty.html.

Los demás nombres y marcas registradas son propiedad de sus respectivos propietarios.

16.4 Contrato de licencia para el usuario final

THIS IS A LEGAL DOCUMENT -- RETAIN FOR YOUR RECORDS

ALTOVA® END USER LICENSE AGREEMENT

Licensor:
Altova GmbH
Rudolfsplatz 13a/9
A-1010 Wien
Austria

Important - Read Carefully. Notice to User:

This End User License Agreement ("Agreement") is a legal document between you and Altova GmbH ("Altova"). It is important that you read this document before using the Altova-provided software ("Software") and any accompanying documentation, including, without limitation printed materials, 'online' files, or electronic documentation ("Documentation"). By clicking the "I accept" and "Next" buttons below, or by installing, or otherwise using the Software, you agree to be bound by the terms of this Agreement as well as the Altova Privacy Policy ("Privacy Policy") including, without limitation, the warranty disclaimers, limitation of liability, data use and termination provisions below, whether or not you decide to purchase the Software. You agree that this agreement is enforceable like any written agreement negotiated and signed by you. If you do not agree, you are not licensed to use the Software, and you must destroy any downloaded copies of the Software in your possession or control. You may print a copy of this Agreement as part of the installation process at the time of acceptance. Alternatively, a copy of this Agreement may be found at http://www.altova.com/eula and a copy of the Privacy Policy may be found at http://www.altova.com/privacy.

1. SOFTWARE LICENSE

(a) License Grant.

(i) Upon your acceptance of this Agreement Altova grants you a non-exclusive, non-transferable (except as provided below), limited license, without the right to grant sublicenses, to install and use a copy of the Software on one compatible personal computer or workstation in the same local area network (LAN) up to the Permitted Number of computers. Subject to the limitations set forth in Section 1(c), you may install and use a copy of the Software on more than one of your compatible personal computers or workstations if you have purchased a Named-User license. Subject to the limitations set forth in Sections 1(d) and 1(e), users may use the software concurrently on a network. The Permitted Number of computers and/or users and the type of license, e.g. Installed, Named-Users, and Concurrent-User, shall be determined and specified at such time as you elect to purchase the Software. Installed user licenses are intended to be fixed and not concurrent. In other words, you cannot uninstall the Software on one machine in order to reinstall that license to a different machine and then uninstall and reinstall back to the original machine. Installations should be static. Notwithstanding the foregoing, permanent uninstallations and redeployments are acceptable in limited circumstances such as if an employee leaves the company or the machine is permanently decommissioned. During the evaluation period, hereinafter defined, only a single user may install and use the software on one (1) personal computer or workstation. If you have licensed the Software as part of a suite of Altova software products (collectively, the "Suite") and have not installed each product individually, then the

Agreement governs your use of all of the software included in the Suite.

(ii) If you have licensed SchemaAgent, then the terms and conditions of this Agreement apply to your use of the SchemaAgent server software ("SchemaAgent Server") included therein, as applicable, and you are licensed to use SchemaAgent Server solely in connection with your use of Altova Software and solely for the purposes described in the accompanying documentation.

(iii) If you have licensed Software that enables users to generate source code, your license to install and use a copy of the Software as provided herein permits you to generate source code based on (i) Altova Library modules that are included in the Software (such generated code hereinafter referred to as the "Restricted Source Code") and (ii) schemas or mappings that you create or provide (such code as may be generated from your schema or mapping source materials hereinafter referred to as the "Unrestricted Source Code"). In addition to the rights granted herein, Altova grants you a non-exclusive, non-transferable, limited license to compile the complete generated code (comprised of the combination of the Restricted Source Code and the Unrestricted Source Code) into executable object code form, and to use, copy, distribute or license that executable. You may not distribute or redistribute, sublicense, sell, or transfer the Restricted Source Code to a third-party in the un-compiled form unless said third-party already has a license to the Restricted Source Code through their separate agreement with Altova. Notwithstanding anything to the contrary herein, you may not distribute, incorporate or combine with other software, or otherwise use the Altova Library modules or Restricted Source Code, or any Altova intellectual property embodied in or associated with the Altova Library modules or Restricted Source Code, in any manner that would subject the Restricted Source Code to the terms of a copyleft, free software or open source license that would require the Restricted Source Code or Altova Library modules source code to be disclosed in source code form. Notwithstanding anything to the contrary herein, you may not use the Software to develop and distribute other software programs that directly compete with any Altova software or service without prior written permission. Altova reserves all other rights in and to the Software. With respect to the feature(s) of UModel that permit reverse-engineering of your own source code or other source code that you have lawfully obtained, such use by you does not constitute a violation of this Agreement. Except as otherwise expressly permitted in Section 1(j) reverse engineering of the Software is strictly prohibited as further detailed therein.

(iv) In the event Restricted Source Code is incorporated into executable object code form, you will include the following statement in (1) introductory splash screens, or if none, within one or more screens readily accessible by the end-user, and (2) in the electronic and/or hard copy documentation: "Portions of this program were developed using Altova® [name of Altova Software, e.g. MapForce® 2016] and includes libraries owned by Altova GmbH, Copyright © 2007-2016 Altova GmbH (www.altova.com)."

(b) Server Use for Installation and Use of SchemaAgent. You may install one (1) copy of the Software on a computer file server within your internal network solely for the purpose of downloading and installing the Software onto other computers within your internal network up to the Permitted Number of computers in a commercial environment only. If you have licensed SchemaAgent, then you may install SchemaAgent Server on any server computer or workstation and use it in connection with your Software. No other network use is permitted, including without limitation using the Software either directly or through commands, data or instructions from or to a computer not part of your internal network, for Internet or Web-hosting services or by any user not licensed to use this copy of the Software through a valid license from Altova.

(c) Named-Use. If you have licensed the "Named-User" version of the software, you may install the Software on up to five (5) compatible personal computers or workstations of which you

are the primary user thereby allowing you to switch from one computer to the other as necessary provided that only one (1) instance of the Software will be used by you as the Named-User at any given time. If you have purchased multiple Named-User licenses, each individual Named-User will receive a separate license key code.

(d) Concurrent Use in Same Local Area Network (LAN). If you have licensed a "Concurrent-User" version of the Software, you may install the Software on any compatible computers in a commercial environment only, up to ten (10) times the Permitted Number of users, provided that only the Permitted Number of users actually use the Software at the same time and further provided that the computers on which the Software is installed are on the same local area network (LAN). The Permitted Number of concurrent users shall be delineated at such time as you elect to purchase the Software licenses. Each separate local area network (LAN) requires its own set of separate Concurrent User Licenses for those wishing to use the Concurrent User versions of the Software in more than one location or on more than one network, all subject to the above Permitted Number limitations and based on the number of users using the Software. If a computer is not on the same local area network (LAN), then a locally installed user license or a license dedicated to concurrent use in a virtual environment is required.

(e) Concurrent Use in Virtual Environment. If you have purchased Concurrent-User Licenses, you may install a copy of the Software on a single host terminal server (Microsoft Terminal Server or Citrix Metaframe), application virtualization server (Microsoft App-V, Citrix XenApp, or VMWare ThinApp) or virtual machine environment within your internal network for the sole and exclusive purpose of permitting individual users within your organization to access and use the Software through a terminal server, application virtualization session, or virtual machine environment from another computer provided that the total number of users that access or use the Software concurrently at any given point in time on such network, virtual machine or terminal server does not exceed the Permitted Number; and provided that the total number of users authorized to use the Software through the terminal server, application virtualization session, or virtual machine environment does not exceed ten (10) times the Permitted Number of users. Key codes for concurrent users cannot be deployed to more than one host terminal server, application virtualization server or virtual machine environment. You must deploy a reliable and accurate means of preventing users from exceeding the Permitted Number of concurrent users. Altova makes no warranties or representations about the performance of Altova software in a terminal server, application virtualization session, or virtual machine environment and the foregoing are expressly excluded from the limited warranty in Section 5 hereof. Technical support is not available with respect to issues arising from use in such environments.

(f) Backup and Archival Copies. You may make one (1) backup and one (1) archival copy of the Software, provided your backup and archival copies are not installed or used on any computer and further provided that all such copies shall bear the original and unmodified copyright, patent and other intellectual property markings that appear on or in the Software. You may not transfer the rights to a backup or archival copy unless you transfer all rights in the Software as provided under Section 3.

(g) Key Codes, Upgrades and Updates. Prior to your purchase and as part of the registration for the thirty (30) day evaluation period, as applicable, you will receive an evaluation key code. You will receive a purchase key code when you elect to purchase the Software from either Altova GmbH or an authorized reseller. The purchase key code will enable you to activate the Software beyond the initial evaluation period. You may not re-license, reproduce or distribute any key code except with the express written permission of Altova. If the Software that you have licensed is an upgrade or an update, then the latest update or upgrade that you download and install replaces all or part of the Software previously licensed. The update or upgrade and the associated license keys does not constitute the granting of a second license to the Software in

that you may not use the upgrade or updated copy in addition to the copy of the Software that it is replacing and whose license has terminated.

(h) Title. Title to the Software is not transferred to you. Ownership of all copies of the Software and of copies made by you is vested in Altova, subject to the rights of use granted to you in this Agreement. As between you and Altova, documents, files, stylesheets, generated program code (including the Unrestricted Source Code) and schemas that are authored or created by you via your utilization of the Software, in accordance with its Documentation and the terms of this Agreement, are your property unless they are created using Evaluation Software, as defined in Section 4 of this Agreement, in which case you have only a limited license to use any output that contains generated program code (including Unrestricted Source Code) such as Java, C++, C#, VB.NET or XSLT and associated project files and build scripts, as well as generated XML, XML Schemas, documentation, UML diagrams, and database structures only for the thirty (30) day evaluation period.

(i) Reverse Engineering. Except and to the limited extent as may be otherwise specifically provided by applicable law in the European Union, you may not reverse engineer, decompile, disassemble or otherwise attempt to discover the source code, underlying ideas, underlying user interface techniques or algorithms of the Software by any means whatsoever, directly or indirectly, or disclose any of the foregoing, except to the extent you may be expressly permitted to decompile under applicable law in the European Union, if it is essential to do so in order to achieve operability of the Software with another software program, and you have first requested Altova to provide the information necessary to achieve such operability and Altova has not made such information available. Altova has the right to impose reasonable conditions and to request a reasonable fee before providing such information. Any information supplied by Altova or obtained by you, as permitted hereunder, may only be used by you for the purpose described herein and may not be disclosed to any third party or used to create any software which is substantially similar to the expression of the Software. Requests for information from users in the European Union with respect to the above should be directed to the Altova Customer Support Department.

(j) Other Restrictions. You may not loan, rent, lease, sublicense, distribute or otherwise transfer all or any portion of the Software to third parties except to the limited extent set forth in Section 3 or as otherwise expressly provided. You may not copy the Software except as expressly set forth above, and any copies that you are permitted to make pursuant to this Agreement must contain the same copyright, patent and other intellectual property markings that appear on or in the Software. You may not modify, adapt or translate the Software. You may not, directly or indirectly, encumber or suffer to exist any lien or security interest on the Software; knowingly take any action that would cause the Software to be placed in the public domain; or use the Software in any computer environment not specified in this Agreement. You may not permit any use of or access to the Software by any third party in connection with a commercial service offering, such as for a cloud-based or web-based SaaS offering.

You will comply with applicable law and Altova's instructions regarding the use of the Software. You agree to notify your employees and agents who may have access to the Software of the restrictions contained in this Agreement and to ensure their compliance with these restrictions.

(k) NO GUARANTEE. THE SOFTWARE IS NEITHER GUARANTEED NOR WARRANTED TO BE ERROR-FREE NOR SHALL ANY LIABILITY BE ASSUMED BY ALTOVA IN THIS RESPECT. NOTWITHSTANDING ANY SUPPORT FOR ANY TECHNICAL STANDARD, THE SOFTWARE IS NOT INTENDED FOR USE IN OR IN CONNECTION WITH, WITHOUT LIMITATION, THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION,

COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL EQUIPMENT, MEDICAL DEVICES OR LIFE SUPPORT SYSTEMS, MEDICAL OR HEALTH CARE APPLICATIONS, OR OTHER APPLICATIONS WHERE THE FAILURE OF THE SOFTWARE OR ERRORS IN DATA PROCESSING COULD LEAD TO DEATH, PERSONAL INJURY OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE. YOU AGREE THAT YOU ARE SOLELY RESPONSIBLE FOR THE ACCURACY AND ADEQUACY OF THE SOFTWARE AND ANY DATA GENERATED OR PROCESSED BY THE SOFTWARE FOR YOUR INTENDED USE AND YOU WILL DEFEND, INDEMNIFY AND HOLD ALTOVA, ITS OFFICERS AND EMPLOYEES HARMLESS FROM ANY THIRD PARTY CLAIMS, DEMANDS, OR SUITS THAT ARE BASED UPON THE ACCURACY AND ADEQUACY OF THE SOFTWARE IN YOUR USE OR ANY DATA GENERATED BY THE SOFTWARE IN YOUR USE.

2. INTELLECTUAL PROPERTY RIGHTS

You acknowledge that the Software and any copies that you are authorized by Altova to make are the intellectual property of and are owned by Altova and its suppliers. The structure, organization and code of the Software are the valuable trade secrets and confidential information of Altova and its suppliers. The Software is protected by copyright, including without limitation by United States Copyright Law, international treaty provisions and applicable laws in the country in which it is being used. You acknowledge that Altova retains the ownership of all patents, copyrights, trade secrets, trademarks and other intellectual property rights pertaining to the Software, and that Altova's ownership rights extend to any images, photographs, animations, videos, audio, music, text and "applets" incorporated into the Software and all accompanying printed materials. You will take no actions which adversely affect Altova's intellectual property rights in the Software. Trademarks shall be used in accordance with accepted trademark practice, including identification of trademark owners' names. Trademarks may only be used to identify printed output produced by the Software, and such use of any trademark does not give you any right of ownership in that trademark. Altova®, XMLSpy®, Authentic®, StyleVision®, MapForce®, UModel®, DatabaseSpy®, DiffDog®, SchemaAgent®, SemanticWorks®, MissionKit®, Markup Your Mind®, Nanonull™, RaptorXML™, RaptorXML Server™, RaptorXML +XBRL Server™, Powered By RaptorXML™, FlowForce Server™, StyleVision Server™, and MapForce Server™ are trademarks of Altova GmbH. (pending or registered in numerous countries). Unicode and the Unicode Logo are trademarks of Unicode, Inc. Windows, Windows XP, Windows Vista, Windows 7, and Windows 8 are trademarks of Microsoft. W3C, CSS, DOM, MathML, RDF, XHTML, XML and XSL are trademarks (registered in numerous countries) of the World Wide Web Consortium (W3C); marks of the W3C are registered and held by its host institutions, MIT, INRIA and Keio. Except as expressly stated above, this Agreement does not grant you any intellectual property rights in the Software. Notifications of claimed copyright infringement should be sent to Altova's copyright agent as further provided on the Altova Web Site.

3. LIMITED TRANSFER RIGHTS

Notwithstanding the foregoing, you may transfer all your rights to use the Software to another person or legal entity provided that: (a) you also transfer this Agreement, the Software and all other software or hardware bundled or pre-installed with the Software, including all copies, updates and prior versions, and all copies of font software converted into other formats, to such person or entity; (b) you retain no copies, including backups and copies stored on a computer; (c) the receiving party secures a personalized key code from Altova; and (d) the receiving party accepts the terms and conditions of this Agreement and any other terms and conditions upon which you legally purchased a license to the Software. Notwithstanding the foregoing, you may not transfer education, pre-release, or not-for-resale copies of the Software.

4. PRE-RELEASE AND EVALUATION PRODUCT ADDITIONAL TERMS

If the product you have received with this license is pre-commercial release or beta Software ("Pre-release Software"), then this Section applies. In addition, this section applies to all evaluation and/ or demonstration copies of Altova software ("Evaluation Software") and continues in effect until you purchase a license. To the extent that any provision in this section is in conflict with any other term or condition in this Agreement, this section shall supersede such other term(s) and condition(s) with respect to the Pre-release and/or Evaluation Software, but only to the extent necessary to resolve the conflict. You acknowledge that the Pre-release Software is a pre-release version, does not represent final product from Altova, and may contain bugs, errors and other problems that could cause system or other failures and data loss. CONSEQUENTLY, THE PRE-RELEASE AND/OR EVALUATION SOFTWARE IS PROVIDED TO YOU **"AS-IS" WITH NO WARRANTIES FOR USE OR PERFORMANCE,** AND ALTOVA DISCLAIMS ANY WARRANTY OR LIABILITY OBLIGATIONS TO YOU OF ANY KIND, WHETHER EXPRESS OR IMPLIED. WHERE LEGALLY LIABILITY CANNOT BE EXCLUDED FOR PRE-RELEASE AND/OR EVALUATION SOFTWARE, BUT IT MAY BE LIMITED, ALTOVA'S LIABILITY AND THAT OF ITS SUPPLIERS SHALL BE LIMITED TO THE SUM OF FIFTY DOLLARS (USD $50) IN TOTAL. If the Evaluation Software has a time-out feature, then the software will cease operation after the conclusion of the designated evaluation period. Upon such expiration date, your license will expire unless otherwise extended. Your license to use any output created with the Evaluation Software that contains generated program code (including Unrestricted Source Code) such as Java, C++, C, VB.NET or XSLT and associated project files and build scripts as well as generated XML, XML Schemas, documentation, UML diagrams, and database structures terminates automatically upon the expiration of the designated evaluation period but the license to use such output is revived upon your purchase of a license for the Software that you evaluated and used to create such output. Access to any files created with the Evaluation Software is entirely at your risk. You acknowledge that Altova has not promised or guaranteed to you that Pre-release Software will be announced or made available to anyone in the future, that Altova has no express or implied obligation to you to announce or introduce the Pre-release Software, and that Altova may not introduce a product similar to or compatible with the Pre-release Software. Accordingly, you acknowledge that any research or development that you perform regarding the Pre-release Software or any product associated with the Pre-release Software is done entirely at your own risk. During the term of this Agreement, if requested by Altova, you will provide feedback to Altova regarding testing and use of the Pre-release Software, including error or bug reports. If you have been provided the Pre-release Software pursuant to a separate written agreement, your use of the Software is governed by such agreement. You may not sublicense, lease, loan, rent, distribute or otherwise transfer the Pre-release Software. Upon receipt of a later unreleased version of the Pre-release Software or release by Altova of a publicly released commercial version of the Software, whether as a stand-alone product or as part of a larger product, you agree to return or destroy all earlier Pre-release Software received from Altova and to abide by the terms of the license agreement for any such later versions of the Pre-release Software.

5. LIMITED WARRANTY AND LIMITATION OF LIABILITY

(a) Limited Warranty and Customer Remedies. Altova warrants to the person or entity that first purchases a license for use of the Software pursuant to the terms of this Agreement that (i) the Software will perform substantially in accordance with any accompanying Documentation for a period of ninety (90) days from the date of receipt, and (ii) any support services provided by Altova shall be substantially as described in Section 6 of this agreement. Some states and jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you. To the extent allowed by applicable law, implied warranties on the Software, if any, are limited to ninety (90) days. Altova's and its suppliers' entire liability and your exclusive remedy shall be, at Altova's option, either (i) return of the price paid, if any, or (ii) repair

or replacement of the Software that does not meet Altova's Limited Warranty and which is returned to Altova with a copy of your receipt. This Limited Warranty is void if failure of the Software has resulted from accident, abuse, misapplication, abnormal use, Trojan horse, virus, or any other malicious external code. Any replacement Software will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. This limited warranty does not apply to Evaluation and/or Pre-release Software.

(b) No Other Warranties and Disclaimer. THE FOREGOING LIMITED WARRANTY AND REMEDIES STATE THE SOLE AND EXCLUSIVE REMEDIES FOR ALTOVA OR ITS SUPPLIER'S BREACH OF WARRANTY. ALTOVA AND ITS SUPPLIERS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, AND FOR ANY WARRANTY, CONDITION, REPRESENTATION OR TERM TO THE EXTENT WHICH THE SAME CANNOT OR MAY NOT BE EXCLUDED OR LIMITED BY LAW APPLICABLE TO YOU IN YOUR JURISDICTION, ALTOVA AND ITS SUPPLIERS MAKE NO WARRANTIES, CONDITIONS, REPRESENTATIONS OR TERMS, EXPRESS OR IMPLIED, WHETHER BY STATUTE, COMMON LAW, CUSTOM, USAGE OR OTHERWISE AS TO ANY OTHER MATTERS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, ALTOVA AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, SATISFACTORY QUALITY, INFORMATIONAL CONTENT OR ACCURACY, QUIET ENJOYMENT, TITLE AND NON-INFRINGEMENT, WITH REGARD TO THE SOFTWARE, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

(c) Limitation of Liability. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW EVEN IF A REMEDY FAILS ITS ESSENTIAL PURPOSE, IN NO EVENT SHALL ALTOVA OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE OR THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF ALTOVA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, ALTOVA'S ENTIRE LIABILITY UNDER ANY PROVISION OF THIS AGREEMENT SHALL BE LIMITED TO THE AMOUNT ACTUALLY PAID BY YOU FOR THE SOFTWARE PRODUCT. Because some states and jurisdictions do not allow the exclusion or limitation of liability, the above limitation may not apply to you. In such states and jurisdictions, Altova's liability shall be limited to the greatest extent permitted by law and the limitations or exclusions of warranties and liability contained herein do not prejudice applicable statutory consumer rights of person acquiring goods otherwise than in the course of business. The disclaimer and limited liability above are fundamental to this Agreement between Altova and you.

(d) Infringement Claims. Altova will indemnify and hold you harmless and will defend or settle any claim, suit or proceeding brought against you by a third party that is based upon a claim that the content contained in the Software infringes a copyright or violates an intellectual or proprietary right protected by United States or European Union law ("Claim"), but only to the extent the Claim arises directly out of the use of the Software and subject to the limitations set forth in Section 5 of this Agreement except as otherwise expressly provided. You must notify Altova in writing of any Claim within ten (10) business days after you first receive notice of the Claim, and you shall provide to Altova at no cost such assistance and cooperation as Altova may reasonably request from time to time in connection with the defense of the Claim. Altova shall have sole control over any Claim (including, without limitation, the selection of counsel and the

right to settle on your behalf on any terms Altova deems desirable in the sole exercise of its discretion). You may, at your sole cost, retain separate counsel and participate in the defense or settlement negotiations. Altova shall pay actual damages, costs, and attorney fees awarded against you (or payable by you pursuant to a settlement agreement) in connection with a Claim to the extent such direct damages and costs are not reimbursed to you by insurance or a third party, to an aggregate maximum equal to the purchase price of the Software. If the Software or its use becomes the subject of a Claim or its use is enjoined, or if in the opinion of Altova's legal counsel the Software is likely to become the subject of a Claim, Altova shall attempt to resolve the Claim by using commercially reasonable efforts to modify the Software or obtain a license to continue using the Software. If in the opinion of Altova's legal counsel the Claim, the injunction or potential Claim cannot be resolved through reasonable modification or licensing, Altova, at its own election, may terminate this Agreement without penalty, and will refund to you on a pro rata basis any fees paid in advance by you to Altova. THE FOREGOING CONSTITUTES ALTOVA'S SOLE AND EXCLUSIVE LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT. This indemnity does not apply to situations where the alleged infringement, whether patent or otherwise, is the result of a combination of the Altova software and additional elements supplied by you.

6. SUPPORT AND MAINTENANCE

Altova offers multiple optional "Support & Maintenance Package(s)" ("SMP") for the version of Software product edition that you have licensed, which you may elect to purchase in addition to your Software license. The Support Period, hereinafter defined, covered by such SMP shall be delineated at such time as you elect to purchase a SMP. Your rights with respect to support and maintenance as well as your upgrade eligibility depend on your decision to purchase SMP and the level of SMP that you have purchased:

(a) If you have not purchased SMP, you will receive the Software AS IS and will not receive any maintenance releases or updates. However, Altova, at its option and in its sole discretion on a case by case basis, may decide to offer maintenance releases to you as a courtesy, but these maintenance releases will not include any new features in excess of the feature set at the time of your purchase of the Software. In addition, Altova will provide free technical support to you for thirty (30) days after the date of your purchase (the "Support Period" for the purposes of this paragraph 6(a), and Altova, in its sole discretion on a case by case basis, may also provide free courtesy technical support during your thirty (30) day evaluation period. Technical support is provided via a Web-based support form only, and there is no guaranteed response time.

(b) If you have purchased SMP, then solely for the duration of its delineated Support Period, **you are eligible to receive the version of the Software edition** that you have licensed and all maintenance releases and updates for that edition that are released during your Support Period. For the duration of your SMP's Support Period, you will also be eligible to receive upgrades to the comparable edition of the next version of the Software that succeeds the Software edition that you have licensed for applicable upgrades released during your Support Period. The specific upgrade edition that you are eligible to receive based on your Support Period is further detailed in the SMP that you have purchased. Software that is introduced as separate product is not included in SMP. Maintenance releases, updates and upgrades may or may not include additional features. In addition, Altova will provide Priority Technical Support to you for the duration of the Support Period. Priority Technical Support is provided via a Web-based support form only and Altova will make commercially reasonable efforts to respond via e-mail to all requests within forty-eight (48) hours during Altova's business hours (MO-FR, 8am UTC – 10pm UTC, Austrian and US holidays excluded) and to make reasonable efforts to provide work-arounds to errors reported in the Software.

During the Support Period you may also report any Software problem or error to Altova. If Altova

determines that a reported reproducible material error in the Software exists and significantly impairs the usability and utility of the Software, Altova agrees to use reasonable commercial efforts to correct or provide a usable work-around solution in an upcoming maintenance release or update, which is made available at certain times at Altova's sole discretion.

If Altova, in its discretion, requests written verification of an error or malfunction discovered by you or requests supporting example files that exhibit the Software problem, you shall promptly provide such verification or files, by email, telecopy, or overnight mail, setting forth in reasonable detail the respects in which the Software fails to perform. You shall use reasonable efforts to cooperate in diagnosis or study of errors. Altova may include error corrections in maintenance releases, updates, or new major releases of the Software. Altova is not obligated to fix errors that are immaterial. Immaterial errors are those that do not significantly impact use of the Software as determined by Altova in its sole discretion. Whether or not you have purchased the Support & Maintenance Package, technical support only covers issues or questions resulting directly out of the operation of the Software and Altova will not provide you with generic consultation, assistance, or advice under any circumstances.

Updating Software may require the updating of software not covered by this Agreement before installation. Updates of the operating system and application software not specifically covered by this Agreement are your responsibility and will not be provided by Altova under this Agreement. Altova's obligations under this Section 6 are contingent upon your proper use of the Software and your compliance with the terms and conditions of this Agreement at all times. Altova shall be under no obligation to provide the above technical support if, in Altova's opinion, the Software has failed due to the following conditions: (i) damage caused by the relocation of the Software to another location or CPU; (ii) alterations, modifications or attempts to change the Software without Altova's written approval; (iii) causes external to the Software, such as natural disasters, the failure or fluctuation of electrical power, or computer equipment failure; (iv) your failure to maintain the Software at Altova's specified release level; or (v) use of the Software with other software without Altova's prior written approval. It will be your sole responsibility to: (i) comply with all Altova-specified operating and troubleshooting procedures and then notify Altova immediately of Software malfunction and provide Altova with complete information thereof; (ii) provide for the security of your confidential information; (iii) establish and maintain backup systems and procedures necessary to reconstruct lost or altered files, data or programs.

7. SOFTWARE ACTIVATION, UPDATES AND LICENSE METERING

(a) License Metering. The Software includes a built-in license metering module that is designed to assist you with monitoring license compliance in small local area networks (LAN). The metering module attempts to communicate with other machines on your local area network (LAN). You permit Altova to use your internal network for license monitoring for this purpose. This license metering module may be used to assist with your license compliance but should not be the sole method. Should your firewall settings block said communications, you must deploy an accurate means of monitoring usage by the end user and preventing users from using the Software more than the Permitted Number.

(b) License Compliance Monitoring. You are required to utilize a process or tool to ensure that the Permitted Number is not exceeded. Without prejudice or waiver of any potential violations of the Agreement, Altova may provide you with additional compliance tools should you be unable to accurately account for license usage within your organization. If provided with such a tool by Altova, you (a) are required to use it in order to comply with the terms of this Agreement and (b) permit Altova to use your internal network for license monitoring and metering and to generate compliance reports that are communicated to Altova from time to time.

(c) Software Activation. **The Software may use your internal network and Internet connection for the purpose of transmitting license-related data at the time of installation, registration, use, or update to an Altova Master License Server and validating the authenticity of the license-related data in order to protect Altova against unlicensed or illegal use of the Software and to improve customer service. Activation is based on the exchange of license related data between your computer and the Altova Master License Server. You agree that Altova may use these measures and you agree to follow any applicable requirements. You further agree that use of license key codes that are not or were not generated by Altova and lawfully obtained from Altova, or an authorized reseller as part of an effort to activate or use the Software violates Altova's intellectual property rights as well as the terms of this Agreement. You agree that efforts to circumvent or disable Altova's copyright protection mechanisms, the license management mechanism, or the Altova Master License Server violate Altova's intellectual property rights as well as the terms of this Agreement. Altova expressly reserves the rights to seek all available legal and equitable remedies to prevent such actions and to recover lost profits, damages and costs.**

(d) LiveUpdate. Altova provides a new LiveUpdate notification service to you, which is free of charge. Altova may use your internal network and Internet connection for the purpose of transmitting license-related data to an Altova-operated LiveUpdate server to validate your license at appropriate intervals and determine if there is any update available for you.

(e) Use of Data. The terms and conditions of the Privacy Policy are set out in full at http://www.altova.com/privacy and are incorporated by reference into this Agreement. By your acceptance of the terms of this Agreement and/or use of the Software, you authorize the collection, use and disclosure of information collected by Altova for the purposes provided for in this Agreement and/or the Privacy Policy. Altova has the right in its sole discretion to amend this provision of the Agreement and/or Privacy Policy at any time. You are encouraged to review the terms of the Privacy Policy as posted on the Altova Web site from time to time.

(f) Audit Rights. You agree that Altova may audit your use of the Software for compliance with the terms of this Agreement at any time, upon reasonable notice. In the event that such audit reveals any use of the Software by you other than in full compliance with the terms of this Agreement, you shall reimburse Altova for all reasonable expenses related to such audit in addition to any other liabilities you may incur as a result of such non-compliance.

(g) Notice to European Users. Please note that the information as described in paragraph 7(d) above may be transferred outside of the European Economic Area, for purposes of processing, analysis, and review, by Altova, Inc., a company located in Beverly, Massachusetts, U.S.A., or its subsidiaries or Altova's subsidiaries or divisions, or authorized partners, located worldwide. You are advised that the United States uses a sectoral model of privacy protection that relies on a mix of legislation, governmental regulation, and self-regulation. You are further advised that the Council of the European Union has found that this model does not provide "adequate" privacy protections as contemplated by Article 25 of the European Union's Data Directive. (Directive 95/46/EC, 1995 O.J. (L 281) 31). Article 26 of the European Union's Data Directive allows for transfer of personal data from the European Union to a third country if the individual has unambiguously given his consent to the transfer of personal information, regardless of the third country's level of protection. By agreeing to this Agreement, you consent to the transfer of all such information to the United States and the processing of that information as described in this Agreement and the Privacy Policy.

8. TERM AND TERMINATION

This Agreement may be terminated (a) by your giving Altova written notice of termination; (b) by Altova, at its option, giving you written notice of termination if you commit a breach of this Agreement and fail to cure such breach within ten (10) days after notice from Altova; or (c) at the request of an authorized Altova reseller in the event that you fail to make your license payment or other monies due and payable. In addition the Agreement governing your use of a previous version of the Software that you have upgraded or updated is terminated upon your acceptance of the terms and conditions of the Agreement accompanying such upgrade or update. Upon any termination of the Agreement, you must cease all use of the Software that this Agreement governs, destroy all copies then in your possession or control and take such other actions as Altova may reasonably request to ensure that no copies of the Software remain in your possession or control. The terms and conditions set forth in Sections 1(h), 1(i), 1(j), 1(k), 1(l), 2, 5, 7, 9, 10, 11, and 11 survive termination as applicable.

9. RESTRICTED RIGHTS NOTICE AND EXPORT RESTRICTIONS

The Software was developed entirely at private expense and is commercial computer software provided with **RESTRICTED RIGHTS**. Use, duplication or disclosure by the U.S. Government or a U.S. Government contractor or subcontractor is subject to the restrictions set forth in this Agreement and as provided in FAR 12.211 and 12.212 (48 C.F.R. §12.211 and 12.212) or DFARS 227. 7202 (48 C.F.R. §227-7202) as applicable. Consistent with the above as applicable, Commercial Computer Software and Commercial Computer Documentation licensed to U.S. government end users only as commercial items and only with those rights as are granted to all other end users under the terms and conditions set forth in this Agreement. Manufacturer is Altova GmbH, Rudolfsplatz 13a/9, A-1010 Vienna, Austria/EU. You may not use or otherwise export or re-export the Software or Documentation except as authorized by United States law and the laws of the jurisdiction in which the Software was obtained. In particular, but without limitation, the Software or Documentation may not be exported or re-exported (i) into (or to a national or resident of) any U.S. embargoed country or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Department of Commerce's Table of Denial Orders. By using the Software, you represent and warrant that you are not located in, under control of, or a national or resident of any such country or on any such list.

10. U.S. GOVERNMENT ENTITIES

Notwithstanding the foregoing, if you are an agency, instrumentality or department of the federal government of the United States, then this Agreement shall be governed in accordance with the laws of the United States of America, and in the absence of applicable federal law, the laws of the Commonwealth of Massachusetts will apply. Further, and notwithstanding anything to the contrary in this Agreement (including but not limited to Section 5 (Indemnification)), all claims, demands, complaints and disputes will be subject to the Contract Disputes Act (41 U.S.C. §§7101 *et seq.*), the Tucker Act (28 U.S.C. §1346(a) and §1491), or the Federal Tort Claims Act (28 U.S.C. §§1346(b), 2401-2402, 2671-2672, 2674-2680), FAR 1.601(a) and 43.102 (Contract Modifications); FAR 12.302(b), as applicable, or other applicable governing authority. For the avoidance of doubt, if you are an agency, instrumentality, or department of the federal, state or local government of the U.S. or a U.S. public and accredited educational institution, then your indemnification obligations are only applicable to the extent they would not cause you to violate any applicable law (e.g., the Anti-Deficiency Act), and you have any legally required authorization or authorizing statute.

11. THIRD PARTY SOFTWARE

The Software may contain third party software which requires notices and/or additional terms and conditions. Such required third party software notices and/or additional terms and conditions are

located at our Website at http://www.altova.com/legal_3rdparty.html and are made a part of and incorporated by reference into this Agreement. By accepting this Agreement, you are also accepting the additional terms and conditions, if any, set forth therein.

12. JURISDICTION, CHOICE OF LAW, AND VENUE

If you are located in the European Union and are using the Software in the European Union and not in the United States, then this Agreement will be governed by and construed in accordance with the laws of the Republic of Austria (excluding its conflict of laws principles and the U.N. Convention on Contracts for the International Sale of Goods) and you expressly agree that exclusive jurisdiction for any claim or dispute with Altova or relating in any way to your use of the Software resides in the Handelsgericht, Wien (Commercial Court, Vienna) and you further agree and expressly consent to the exercise of personal jurisdiction in the Handelsgericht, Wien (Commercial Court, Vienna) in connection with any such dispute or claim.

If you are located in the United States or are using the Software in the United States then this Agreement will be governed by and construed in accordance with the laws of the Commonwealth of Massachusetts, USA (excluding its conflict of laws principles and the U.N. Convention on Contracts for the International Sale of Goods) and you expressly agree that exclusive jurisdiction for any claim or dispute with Altova or relating in any way to your use of the Software resides in the federal or state courts of the Commonwealth of Massachusetts and you further agree and expressly consent to the exercise of personal jurisdiction in the federal or state courts of the Commonwealth of Massachusetts in connection with any such dispute or claim.

If you are located outside of the European Union or the United States and are not using the Software in the United States, then this Agreement will be governed by and construed in accordance with the laws of the Republic of Austria (excluding its conflict of laws principles and the U.N. Convention on Contracts for the International Sale of Goods) and you expressly agree that exclusive jurisdiction for any claim or dispute with Altova or relating in any way to your use of the Software resides in the Handelsgericht, Wien (Commercial Court, Vienna) and you further agree and expressly consent to the exercise of personal jurisdiction in the Handelsgericht Wien (Commercial Court, Vienna) in connection with any such dispute or claim. This Agreement will not be governed by the conflict of law rules of any jurisdiction or the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded.

13. TRANSLATIONS

Where Altova has provided you with a foreign translation of the English language version, you agree that the translation is provided for your convenience only and that the English language version will control. If there is any contradiction between the English language version and a translation, then the English language version shall take precedence.

14. GENERAL PROVISIONS

This Agreement contains the entire agreement and understanding of the parties with respect to the subject matter hereof, and supersedes all prior written and oral understandings of the parties with respect to the subject matter hereof. Any notice or other communication given under this Agreement shall be in writing and shall have been properly given by either of us to the other if sent by certified or registered mail, return receipt requested, or by overnight courier to the address shown on Altova's Web site for Altova and the address shown in Altova's records for you, or such other address as the parties may designate by notice given in the manner set forth above. This Agreement will bind and inure to the benefit of the parties and our respective heirs, personal and legal representatives, affiliates, successors and permitted assigns. The failure of either of us at

any time to require performance of any provision hereof shall in no manner affect such party's right at a later time to enforce the same or any other term of this Agreement. This Agreement may be amended only by a document in writing signed by both of us. In the event of a breach or threatened breach of this Agreement by either party, the other shall have all applicable equitable as well as legal remedies. Each party is duly authorized and empowered to enter into and perform this Agreement. If, for any reason, any provision of this Agreement is held invalid or otherwise unenforceable, such invalidity or unenforceability shall not affect the remainder of this Agreement, and this Agreement shall continue in full force and effect to the fullest extent allowed by law. The parties knowingly and expressly consent to the foregoing terms and conditions.

Last updated: 2015/09/03

Índice

.

.NET Framework,
incluir archivo, 148

1

1.4,
Java, 60

5

5.0,
Java, 60

A

Abrir,
diagrama, 74
paquetes en vista jerárquica, 69
URL, 393
Abrir proyecto,
control de código fuente, 346
Absolutos,
vínculos absolutos y relativos, 176
Abstracta,
clase, 24
Acceso directo,
asignar/eliminar, 419
mostrar en información rápida, 421
tecla, 419
teclado, 419
Acelerar,
rendimiento, 165
Activar/desactivar,
modo compacto, 313
Actividad, 207
agregar a estado, 207
agregar operación, 207
iconos, 373
Actividades,
agregar diagrama de actividades a transición, 207
diagrama de, 189
Actor,
definido por el usuario, 14
personalizar, 14
Actualizar,
archivo de proyectos, 135
Actualizar estado,
control de código fuente, 364
Advertencia,
mensajes, 88
revisión de la sintaxis, 53
Agregar, 356
a Favoritos, 76
al control de código fuente, 356
diagrama a un paquete, 14
insertar / eliminar en Estructura del modelo, 69
mover, eliminar elementos en un diagrama, 89
paquete a un proyecto, 14
proyecto al control de código fuente, 356
proyecto nuevo, 118
vínculo al modelo, 100
Ajustar,
a la cuadrícula, 89
Ajuste,
líneas de ajuste mientras se arrastran objetos, 421
Alinear,
elementos, 92
elementos mientras se arrastran, 14
Anotación,
documentación, 86
esquema XML, 313
Aplicación,
externa (argumentos), 416
Árbol de diagramas,
panel, 74
Archivo, 393
abrir desde URL, 393
combinar archivos de proyecto, 150
ejemplo del tutorial, 8
nuevo / cargar / guardar con archivo de procesamiento por lotes, 114
ump, 118
Archivos,

Archivos,
de muestra, 106
Archivos binarios,
importar C# y Java, 131
Archivos de proyecto,
Borland - MS Visual Studio .Net, 402
Archivos locales,
vínculos absolutos o relativos, 176
Argumentos,
herramientas externas, 416
Arrastrar y colocar,
botón secundario del mouse, 98
crear asociaciones, 172
Artefacto,
agregar al nodo, 48
manifestación, 48
Asignar,
acceso directo a un comando, 419
estereotipo, 300
Asociación, 24, 89, 168
agregado/compuesto, 24
calificador, 168
caso de uso, 14
clase memberEnd, 168
crear mediante arrastrar/colocar, 172
definir el tipo, 168
entre clases, 24
grosor de la línea, 89
rol, 168
ver automáticamente, 168
ver durante la ingeniería de código, 60
ver propiedad como, 89
ver propiedad con tipo, 163
ver relaciones, 171
vínculos entre objetos, 36
Atributo,
estereotipo, 300
mostrar / ocultar, 275
mostrar / ocultar valores etiquetados, 300
seleccionar en Estructura del modelo, 78
ventana de finalización automática, 421
Attribute,
coloring, 281
Automática,
respuesta automática a mensajes, 249
Automáticamente,
agregar operación, 207
ver automáticamente las asociaciones, 168
Automático,
hipervínculo, 100
Ayuda, 429

B

Ball and socket,
notación de forma esférica, 275
Bank,
archivos de muestra, 106
Barra de herramientas, 416
activar/desactivar, 416
agregar comando, 414
crear nueva, 416
mostrar iconos de tamaño grande, 421
restaurar comandos, 416
Barras de herramientas,
restaurar, 414
Base,
clase, 30
BD,
correspondencia entre el modelo y la BD, 147
Binarios,
compatibilidad con binarios confusos, 131
Borland,
archivo de proyecto BSDJ, 402
BSDJ,
proyecto Borland, 402
Buscar, 69, 399
elementos de modelado, 69, 399
elementos no utilizados, 69
pestañas de búsqueda, 68
y reemplazar, 399

C

C#,
código, 436
configurar importación, 126
correspondencia entre el modelo y el código, 144
importar archivo binario, 131
C++,
código, 436
Calificador,

Calificador,
asociación, 168
CallBehavior,
insertar, 190
CallOperation,
insertar, 190
Cambiar de nombre,
clasificador, 137
Cambiar de proveedor,
control de código fuente, 364
Cambiar nombres de clases,
efecto en el nombre del archivo de código, 139
Cambio de estado,
definir en una línea de tiempo, 267
Carpeta,
carpeta de ejemplos, 8
Carpetas,
obtener carpetas en control de código fuente, 351
Casilla de activación,
especificación de ejecución, 243
Caso de uso,
agregar, 14
asociación, 14
compartimento, 14
multilínea, 14
Casos de uso,
diagrama de, 231
iconos, 388
Catálogo,
archivo de catálogo, 421
Centrar,
elementos, 92
Cerrar,
todos los diagramas excepto los que están activos, 89
Ciclo de vida,
diagrama de, 266
iconos, 387
Clase, 121, 275
crear en espacio de nombres, 121
en diagramas de componentes, 42
expandir y contraer compartimentos, 275
generar en espacio de nombres, 121
iconos, 375
interfaz con ball and socket (notación de forma esférica), 275
operación de clase (invalidar), 275
varias instancias en el diagrama, 275
Clase base,
insertar clases derivadas, 98
invalidar, 275
Clases,
abstractas y concretas, 24
agregar, 24
agregar operaciones, 24
agregar propiedades, 24
asociaciones, 24
base, 30
cambios de nombre (sincronización), 139
derivadas, 30
diagrama de, 275
diagramas, 24
habilitar ventana de finalización automática, 421
insertar clases derivadas, 98
sincronización, 135
Clasificador,
cambiar de nombre, 137
nuevo, 137
restringir, 160
Class,
syntax coloring, 281
Código, 139
agregar código a diagrama de secuencia, 263
código Java y nombres de archivos de clases, 139
condiciones para la generación, 141
directorio de destino, 53
generar código a partir de diagramas de secuencia, 259
generar diagrama de secuencia a partir de código, 254
generar varios diagramas de secuencia a partir de código, 257
ingeniería de ida y vuelta, 53
predeterminado, 421
refactorizar, 139
requisitos, 53
sincronización, 135
Código C#,
correspondencia con elementos de UModel, 144
Código de la BD,
correspondencia con elementos de UModel, 147
Código de refactorización,
nombres de clases (sincronizar), 139
Código fuente,
importar, 126
Código Java,
correspondencia con elementos de UModel, 143
Código VB.NET,
correspondencia con elementos de UModel, 146
Código XML Schema,
correspondencia con elementos de UModel, 145

Colaboración,
diagrama de estructura de un compuesto, 290
Color,
syntax coloring - enable/disable, 281
Color de fondo,
transparente, 307
Comando,
agregar a barra de herramientas / menú, 414
eliminar de menú, 420
menú contextual, 420
procesamiento de la línea de comandos, 110
restaurar menú, 420
Comandos,
línea de comandos: nuevo / cargar / guardar, 114
procesamiento de la línea de comandos, 114
CombinaciónDePaquete, 295
Combinar,
código con el modelo, 53, 402
modelo con el código, 402
omitir directorio, 421
proyectos, 150
Comentarios,
documentación, 86
ver comentarios del código fuente en el diagrama, 89
Compacto,
modo (activar/desactivar), 313
Comparar archivos de código fuente, 362
Compartimento,
expandir uno / varios, 275
Compartir,
desde el control de código fuente, 359
paquete y diagrama, 156
Compatibilidad,
actualizar proyectos, 135
Componentes, 42
diagrama, 42
diagrama de, 292
iconos, 378
insertar clase, 42
realización, 42
Comportamiento,
diagramas de, 189
Composición,
crear asociación, 24
Comunicación,
diagrama de, 232
iconos, 376
Concreta,
clase, 24
Condiciones,
para la generación de código, 141
Configuración,
control de código fuente, 421
Configurar,
sincronización, 135
Confusos,
compatibilidad con binarios confusos, 131
Contraer,
compartimentos de clase, 275
Contrato de licencia para el usuario final, 440
Control de código fuente, 342
abrir proyecto, 346
actualizar estado, 364
agregar al control de código fuente, 356
anular desprotección, 355
cambiar de proveedor, 364
comandos, 346
desproteger, 352
ejecutar interfaz nativa, 364
habilitar / deshabilitar, 349
mostrar diferencias, 362
mostrar historial, 360
obtener archivo, 350
obtener la versión más reciente, 349
opciones / configuración, 421
propiedades, 363
proteger, 354
quitar del, 358
Control de versiones,
comandos, 346
subproyectos y trabajo en equipo, 332
Controlador,
crear relaciones, 172
Convergencia,
crear en diagrama de actividades, 192
Copiar,
pegar en Estructura del modelo, Árbol de diagramas, 94
Correo electrónico,
enviar proyecto, 393
Correspondencia,
entre elementos C# y elementos del modelo, 144
entre elementos de la BD y elementos del modelo, 147
entre elementos de VB.NET y elementos del modelo, 146
entre elementos de XML Schema y elementos del modelo, 145
entre elementos Java y elementos del modelo, 143

CPU,
carga - acelerar actualización de estado en segundo plano, 342
CR/LF,
en el archivo ump al guardarlo, 118
Crear,
esquema XML, 323
métodos getter / setter, 275
un proyecto desde cero (ingeniería de código), 121
CSPROJ - CSDPROJ,
MS Visual Studio .Net, 402
Ctrl+Barra espaciadora,
finalización automática a petición, 92
Cuadrícula,
ajustar a la cuadrícula, 89
líneas de ajuste, 421
líneas de ajuste mientras se arrastran objetos, 14
CVS, 342

D

Definidas por el usuario,
plantillas SPL, 135
Definido por el usuario,
actor, 14
Definidos,
símbolos definidos (importar código), 126
Definidos por el usuario,
estilos de estereotipo, 306
Dependencia,
incluir, 14
uso, 42
ver relaciones, 171
Derivada,
clase, 30
Derivadas,
insertar clases derivadas, 98
Descargar proyecto de control de código fuente, 346
Deshabilitar el control de código fuente, 349
Deshacer desprotección, 355
Desproteger, 352
Desviar,
hipervínculos, 100
Diagram,
de perfil, 298
Diagrama, 275, 421
abrir, 74
agregar actividad a transición, 207
agregar código a diagrama de secuencia, 263
compartir paquete y diagrama, 156
de actividades, 189
de casos de uso, 231
de ciclo de vida, 266
de clases, 275
de componentes, 292
de comunicación, 232
de esquema XML, 312
de esquema XML (importación), 313
de estructura de un compuesto, 290
de implementación, 292
de máquina de estados, 205
de objetos, 293
de paquetes, 293
de perfil y estereotipos, 300
de secuencia, 240
elementos de restricción, 69
estilos, 80
generar código a partir de diagramas de secuencia, 259
generar diagrama de dependencias entre paquetes, 293
global de interacción, 236
guardar como PNG, 393
guardar elementos como mapa de bits, 399
guardar los diagramas abiertos con el proyecto, 421
iconos, 372
omitir elementos de archivos incluidos, 421
varias instancias de la clase, 275
Diagrama de actividades,
crear rama / convergencia, 192
elementos, 195
insertar elementos, 190
Diagrama de ciclo de vida,
cambiar de un tipo a otro, 267
ciclo de vida, 267
evento/estímulo, 271
insertar elementos, 267
línea de vida, 267
marca de graduación, 270
mensaje, 273
restricciónDeDuración, 271
restricciónDeTiempo, 272
valor general de la línea de vida, 267
Diagrama de comunicación,
generar a partir de un diagrama de secuencia, 233
Diagrama de paquetes,

Diagrama de paquetes,
insertar elementos, 295
Diagrama de secuencia, 254
agregar código a, 263
fragmento combinado, 244
generar a partir de código, 254
generar a partir de un diagrama de comunicación, 233
generar código a partir de, 259
generar diagrama de secuencia a partir de métodos getter/setter, 257
generar varios diagramas de secuencia a partir de código, 257
insertar elementos, 241
invariante de estado, 249
línea de vida, 243
mensajes, 249
nombres de operación que se deben omitir, 254
puerta, 248
uso de interacción, 248
Diagrama global de interacción,
insertar elementos, 237
Diagramas, 188
ajustar el tamaño, 89
cerrar todos excepto los que están activos, 89
de comportamiento, 189
de estructura, 275
hipervínculo, 100
panel, 89
pegar, 94
peso de las líneas, 89
propiedades, 89
ver comentarios del código fuente, 89
Directa,
ingeniería, 141
Directorio,
cambiar ubicación del proyecto, 118
carpeta de ejemplos, 8
importar, 60
importar código desde, 126
omitir durante la combinación, 421
para generación de código, 53
usar espacio de nombres en la ruta de acceso, 53
Directorio de trabajo,
control de código fuente, 346
Diseño, 411
Disparador,
definir disparador de transición, 207
Distribución,
de productos de software de Altova, 440, 441, 444
División,
evitar división entre las páginas, 393
Documentación, 86, 176
anotación, 86
generar proyecto UML, 176
pestaña, 86, 87
vínculos relativos, 176
Documento,
vincular, 100
Dos bandas,
fusión del proyecto, 150
Dot,
propiedad, 172

E

Edición, 399
Ejecutar interfaz nativa, 364
Ejemplos,
archivos de ejemplo, 106
carpeta del tutorial, 8
Elemento,
agregar a Favoritos, 76
agregar restricción a, 69
asociaciones durante la importación, 60
estilos, 80
generar documentación, 176
guardar elemento seleccionado como mapa de bits, 399
propiedades, 78
restricción, 69
ver jerarquía, 83
vincular, 100
Elementos,
alinear, 92
cortar, copiar, pegar, 94
insertar, 98
insertar en diagrama de máquina de estados, 206
omitir elementos de archivos incluidos, 421
relaciones, 168
Elementos no utilizados,
lista, 69
Elementos que faltan,
lista, 69
Elimina,
acceso directo, 419
Eliminar, 414

Eliminar, 414
barra de herramientas, 416
comando de menú contextual, 420
comando de una barra de herramientas, 414
de Favoritos, 76
icono de una barra de herramientas, 414
relaciones entre clases, 168
En cascada,
estilos, 80
End User License Agreement, 445
Enlace,
plantilla, 162
Enumeración,
valor predeterminado, 304
y estereotipos, 304
Enviar por correo electrónico,
proyecto, 393
Error,
mensajes, 88
revisión de la sintaxis, 53
Espaciar,
en horizontal, 92
Espacio de nombres,
crear una clase en, 121
raíz de espacio de nombres Java, 141
usar para la generación de código, 53
Especializar,
generalizar, 30
Especificación de ejecución,
línea de vida, 243
Esquema,
crear esquema XML, 323
generador de código, 436
tipo de datos (definir), 320
XML, 312
XML (importación), 313
Esquema XML,
anotación, 313
crear/generar, 323
diagrama de, 312
iconos, 389
insertar elementos, 320
modelo de contenido, 320
Estado, 207
agregar actividad, 207
definir transición entre, 207
insertar estado simple, 207
ortogonal, 213
submáquina, 213
Estado compuesto, 213
agregar región, 213
Estado de submáquina,
agregar punto de entrada/salida, 213
Estereotipo,
agregar a diagrama de perfil, 300
asignar, 300
atributos (definir), 300
estilos definidos por el usuario, 306
icono personalizado, 307
memberEnd, 300
perfiles, 300
valor etiquetado predeterminado, 304
y enumeración, 304
Estereotipos,
definición, 298
Estilos,
en cascada, prioridad, 80
estereotipo definido por el usuario, 306
pestaña, 80
varias selecciones, 80
Estructura,
diagramas de, 275
Estructura de un compuesto,
diagrama de, 290
iconos, 377
insertar elementos, 290
Estructura del modelo,
abrir paquetes, 69
crear hipervínculo a un elemento de, 100
panel, 69
seleccionar atributo en, 78
Etiquetado,
valor (predeterminado), 304
Etiquetados,
valores, 300
valores (definición), 298
valores (ocultar/mostrar atributos), 300
Etiquetas,
identificadores ID y UUID, 328
Etiquetas de texto,
ver / ocultar, 171
Evento/estímulo,
diagrama de ciclo de vida, 271
Excepción,
agregar excepción generada, 275
operación Java, 126

Excepciones generadas, 126
agregar, 275
Expandir,
contraer paquetes, 69
todos los compartimentos de clase, 275
Exportar,
como XMI, 328
Extensión,
XMI, 328

F

Favoritos,
panel, 76
Finalización automática, 92
a petición (Ctrl+Barra espaciadora), 92
función, 24
modo único / modo múltiple, 92
ventana (tipos), 92
ventana para edición de clases, 421
Firma,
plantilla, 160, 161
Flujo de trabajo,
proyecto, 118
Formato,
estereotipo definido por el usuario, 306
ventana de finalización automática, 92
Fragmento combinado, 244
Fuente,
comentarios del código fuente (ver en el diagrama), 89
Fusión,
a 2 bandas, 150
a 3 bandas, 151
a 3 bandas manual, 153

G

Generación de código, 53
usar un espacio de nombres como directorio, 53
Generador de código, 436
Generalizar,
especializar, 30
Generar,
código a partir de un esquema, 436
diagrama de secuencia a partir de código, 254
diagrama de secuencia a partir de diagrama de comunicación, 233
documentación del proyecto UML, 176
esquema XML, 323
RealizacionesDeComponente automáticamente, 137
respuesta a mensajes automáticamente, 249
varios diagramas de secuencia a partir de código, 257
Get,
métodos getter / setter, 275
Getter / Setter,
generar diagrama de secuencia a partir de métodos getter/setter, 257
Guardar,
archivos de subproyectos, 334
diagrama como imagen, 393
elementos como mapas de bits, 399

H

Habilitar,
líneas de ajuste mientras se arrastran objetos, 421
Habilitar el control de código fuente, 349
Herramientas, 414
agregar al menú Herramientas, 416
opciones, 421
Hipervínculo, 100
automático, 100
Historial,
mostrar, 360

I

Icono,
actividad, 373
agregar a barra de herramientas / menú, 414
caso de uso, 388
ciclo de vida, 387
clase, 375
componentes, 378
comunicación, 376
esquema XML, 389
estructura de un compuesto, 377
icono de estereotipo personalizado, 307
implementación, 379

Icono,
interacción global, 380
máquina de estados, 386
mostrar iconos de tamaño grande, 421
objeto, 381
paquete, 382
secuencia, 385
Iconos,
visibilidad, 275
Iconos de los diagramas de UModel, 372
ID,
identificadores ID y UUID, 328
Ida y vuelta,
código - modelo - código, 60
ingeniería, 53
modelo - código - modelo, 53
Igualar,
alto / ancho / tamaño, 92
Implementación,
diagrama, 48
diagrama de, 292
iconos, 379
ImportaciónDePaquete, 295
Importar, 60, 328
archivo XMI, 328
archivos binarios, 131
asociación de elementos, 60
código fuente, 126
directorio, 60
en relación al archivo UMP, 126
esquema XML, 313
proyecto, 126
proyecto C#, 126
proyecto de origen, 60
XMI generado con UModel, 328
Imprimir,
vista previa, 393
Incluir, 148, 156
.NET Framework, 148
cambiar estado, 156
compartir paquete y diagrama, 156
dependencia, 14
proyecto de UModel, 148
subproyectos en el proyecto principal, 334
Información legal, 440
Información rápida, 421
mostrar accesos directos en, 421
ver, 421
Información sobre derechos de autor, 440
Ingeniería de código, 60
crear un proyecto desde cero, 121
generar RealizacionesDeComponente, 137
importar directorio, 60
mover archivo de proyecto a una ubicación nueva, 118
ver asociaciones, 60
Iniciar,
con el proyecto anterior, 421
UModel, 10
Insertar, 98, 190
acción (CallBehavior), 190
acción (CallOperation), 190
con..., 98
elementos, 98
elementos en diagrama de paquetes, 295
elementos en diagrama global de interacción, 237
elementos en diagramas de ciclo de vida, 267
elementos en estructura de un compuesto, 290
estado simple, 207
Instalación,
carpeta de ejemplos, 8
Instalador,
multiusuario, 8
Instancia,
diagrama, 36
objeto, 36
varias clases, 275
Integrar,
subproyectos en el proyecto principal, 334
Inteligente,
finalización automática, 24
Interacción,
diagrama global de, 236
Interacción global,
iconos, 380
Interfaz, 275
ball and socket (notación de forma esférica), 275
implementar, 275
Interfaz del usuario, 68
Introducción, 6
Invalidar,
clase base, 275
operaciones de clase, 275
plantillas SPL predeterminadas, 135
Invariante de estado, 249
Ir a,
línea de vida, 243

J

Java,
código, 436
código y nombres de archivos de clases, 139
correspondencia entre el modelo y el código, 143
excepción, 126
importar archivo binario, 131
raíz de espacio de nombres, 141
versiones compatibles, 60
JavaDocs, 86
Jerarquía,
ver todas las relaciones, 83
Jerarquías (diagrama),
niveles en la documentación, 176

L

Licencia,
información sobre, 440
License, 445
Limitar,
elementos de restricción, 69
Línea,
ortogonal, 42
peso/grosor en los diagramas, 89
Línea de tiempo,
definir cambios de estado, 267
Línea de vida,
atributos, 243
ir a, 243
propiedad de tipo, 243
valor general, 267
Línea nueva,
en línea de vida, 233
operandoDeInteracción, 244
Líneas,
de ajuste, 421
formato, 36
Líneas de ajuste, 14
Lista,
elementos no utilizados, 69
Llamada,
mensaje, 249
Llamada del mensaje,
operación ir a, 249

M

Manifestación,
artefacto, 48
Mapa de bits,
guardar elementos como, 399
Máquina de estados,
diagrama de, 205
elementos, 224
estados compuestos, regiones, 213
estados, actividades, transiciones, 207
iconos, 386
insertar elementos, 206
Marca de graduación,
diagrama de ciclo de vida, 270
Marco del diagrama,
ver título del diagrama UML, 92
Medición de licencias,
en los productos de Altova, 442
Mejorar,
rendimiento, 165
Member end,
estereotipo, 300
MemberEnd,
asociación, 168
Mensaje, 249
crear objeto, 249
diagrama de ciclo de vida, 273
flechas, 249
insertar, 249
llamada, 249
mover, 249
numeración, 249
operación ir a, 249
Mensajes,
panel, 88
Menú, 420
agregar / eliminar comando, 414
agregar menú a, 416
archivo, 393
ayuda, 429
diseño, 411

Menú, 420
edición, 399
eliminar comandos de, 420
herramientas, 414
personalizar, 420
predeterminado/XMLSPY, 420
proyecto, 402
ventanas, 428
vista, 413
Menú contextual,
comandos, 420
Metadatos,
salida XMI, 328
Método,
agregar excepción agregada, 275
generar diagrama de secuencia a partir de métodos, 254
generar diagrama de secuencia a partir de métodos getter/setter, 257
generar varios diagramas de secuencia a partir de métodos, 257
Métodos,
getter / setter, 275
MisDocumentos,
archivos de ejemplo, 8
Modelado,
mejorar el rendimiento, 165
Modelo,
agregar vínculo, 100
cambiar el nombre de las clases (efecto en Java), 139
Modelo con el código,
ver asociaciones, 60
Modelo de contenido,
del esquema XML, 320
Modo múltiple,
finalización automática, 92
Modo único,
finalización automática, 92
Mostrar,
ocultar slot, 275
ocultar valores etiquetados / atributos, 300
valores etiquetados, 313
Mostrar / ocultar,
atributos, operaciones, 275
Mostrar diferencias, 362
Mostrar historial, 360
Mouse,
copiar, pegar, 94
Mover,
proyecto, 118
Mover las flechas de los mensajes, 249
MS Visual Source Safe, 342
MS Visual Studio .Net,
archivo de proyecto CSPROJ - CSDPROJ, 402
Muestra,
archivos de muestra, 106
Multilínea,
caso de uso, 14
operandoDeInteracción, 244
texto del actor, 14
Multiusuario,
carpeta de ejemplos, 8

N

Navegar,
hipervínculo, 100
Nodo, 48
agregar, 48
agregar artefacto, 48
estilos, 80
Nombre,
de región (ver/ocultar), 213
Nota,
vincular, 100
Nuevo,
clasificador, 137
Numeración,
mensajes, 249

O

Objeto,
crear mensaje, 249
iconos, 381
vínculos (asociaciones), 36
Objetos,
diagrama, 36
diagrama de, 293
Obtener archivo,
control de código fuente, 350
Obtener carpetas,
control de código fuente, 351

Obtener la versión más reciente, 349
Ocultar,
etiquetas de texto, 171
mostrar slot, 275
valores etiquetados, atributos, 300
Omitir, 421
directorios, 421
elementos en la lista, 421
nombres de operación, 254
Opciones, 421
control de código fuente, 421
herramientas, 421
proyecto, 164
Operación, 275
agregar automáticamente en actividad, 207
excepción, 126
invalidar, 275
ir a (en la llamada del mensaje), 249
mostrar / ocultar, 275
omitir en la generación de diagramas de secuencia, 254
plantilla, 163
ventana de finalización automática, 421
volver a utilizar, 30
Operaciones,
agregar, 24
Operador,
interacción, 244
Operador de interacción,
definir, 244
Operando,
interacción, 244
Operando de interacción,
multilínea, 244
Operation,
coloring, 281
Ordenar,
diagrama, 74
elementos en Estructura del modelo, 69
Ortogonal,
estado, 213
línea, 42
OwnedEnd,
asociación, 168

P

Página,
evitar división entre las páginas, 393
Panel,
Árbol de diagramas, 74
Estructura del modelo, 69
Favoritos, 76
Mensajes, 88
Vista general, 86
Paquete,
compartir, 156
crear un paquete de espacio de nombres, 121
expandir / contraer, 69
iconos, 382
perfil, 300
Paquetes,
diagrama de, 293
generar diagrama de dependencias entre, 293
Parámetro,
de procesamiento por lotes, 110
plantilla, 163
Parcial,
generar documentación parcial, 176
Pegar,
elementos en diagramas, 94
Perfil, 298
aplicación, 298
diagrama, 298
diagrama de, 298, 300
estereotipos, 300
Período de evaluación,
de los productos de software de Altova, 440, 441, 444
Personalizado,
icono de estereotipo, 307
Personalizar, 414
actor, 14
comandos de las barras de herramientas / menús, 414
menú, 420
menú contextual, 420
Peso,
grosor de la línea, 89
Pestañas de búsqueda, 68
Plantilla,
enlace, 162

Plantilla,
firma, 160, 161
operación/parámetro, 163
Plantillas,
SPL definidas por el usuario, 135
PNG,
guardar diagrama, 393
Poner en fila,
elementos, 92
Por lotes, 114
modo de procesamiento por lotes, 114
nuevo / cargar / guardar, 114
procesamiento, 110, 114
Predeterminadas,
plantillas SPL, 135
Predeterminado,
código de proyecto, 421
menú, 420
valor etiquetado, 300
Pretty print,
preparar proyecto para pretty print al guardarlo, 118
salida XMI, 328
Procesamiento por lotes,
modo de procesamiento por lotes, 114
Property,
coloring, 281
Propiedad,
con tipo (ver), 163
de tipo línea de vida, 243
dot, 172
ver como asociación, 171
volver a utilizar, 30
Propiedades, 78
agregar, 24
control de código fuente, 363
Proteger, 354
Proveedor,
de control de código fuente, 342
seleccionar, 346
Proyecto, 402, 421
abrir último proyecto al iniciar UModel, 421
actualizar archivo de proyecto, 135
agregarlo al control de código fuente, 356
código predeterminado, 421
combinar, 150
crear, 118
crear paquete de espacio de nombres, 121
crear subproyectos, 334
enviar por correo electrónico, 393
estilos, 80
flujo de trabajo, 118
fusión a 2 bandas, 150
fusión a 3 bandas, 151
fusión a 3 bandas manual, 153
generar documentación, 176
guardar los diagramas abiertos, 421
guardar para pretty print, 118
importar, 126
incluir proyecto de UModel, 148
insertar paquete, 118
mover, 118
opciones, 164
quitarlo del control de código fuente, 358
revisión de la sintaxis, 402
Proyecto local, 346
Puerta,
diagrama de secuencia, 248
Punto de entrada,
agregar a submáquina, 213
Punto de salida,
agregar a submáquina, 213
PVCS Version Manager, 342

Q

Quitar,
de Favoritos, 76
del control de código fuente, 358

R

Raíz,
catálogo, 421
espacio de nombres Java, 141
sincronizar raíz / paquetes / clases, 135
Rama,
crear en diagrama de actividades, 192
Realización,
componente, 42
Realizaciones,
generar RealizacionesDeComponente, 137
RealizacionesDeComponente,

RealizacionesDeComponente,
generación automática, 137
Rechazar cambios, 355
Recuperar archivo,
control de código fuente, 349
Recursos,
acelerar actualización de estado en segundo plano, 342
Referencia, 392
mostrar clase a la que se hace referencia, 89
Región,
agregar a estado compuesto, 213
nombre de región (ver/ocultar), 213
Relación,
ver todas (pestaña Jerarquía), 83
Relaciones,
crear relaciones usando los controladores, 172
elemento, 168
ver relaciones del modelo, 171
Relativos,
importar/guardar archivos relativos, 126
vínculos relativos en la documentación, 176
Rendimiento,
mejorar, 165
Repositorios, 342
Requisitos,
ingeniería directa, 141
Respuesta,
mensaje (generar automáticamente la respuesta), 249
Restaurar,
acceso directo, 419
barras de herramientas y ventanas, 414
comandos de la barra de herramientas, 416
comandos de menú, 420
Restricción,
agregar al diagrama, 69
asignar a varios elementos, 69
elemento, 69
revisar sintaxis, 402
RestricciónDeDuración,
diagrama de ciclo de vida, 271
RestricciónDeTiempo,
diagrama de ciclo de vida, 272
Restringir,
clasificadores, 160
Reutilizar,
subproyectos, 334
Revisar,
sintaxis del proyecto, 402
Rol,
asociación, 168
Ruta de acceso,
cambiar ubicación del proyecto, 118
carpeta de ejemplos, 8
importar rutas relativas al archivo UMP, 126
usar espacio de nombres en el código, 53

S

Salida,
archivo XMI de salida, 328
Salto de línea,
en el texto del actor, 14
SC,
syntax coloring, 281
Secuencia,
diagrama de, 240
iconos, 385
Segundo plano,
actualización de estado - aumentar el intervalo, 342
Seleccionar atributo,
en Estructura del modelo, 78
Set,
métodos getter / setter, 275
Setter / Getter,
generar diagrama de secuencia a partir de métodos setter/getter, 257
Show,
tagged values, 317
Símbolos,
definidos (importar código), 126
iconos de visibilidad, 275
Sincronización,
cambiar nombres de las clases, 139
opciones de configuración, 135
Sincronizar,
código de combinación a partir del modelo, 53
combinar modelo con el código, 60
en la ubicación nueva, 118
nombre de las clases y el nombre del archivo de código, 139
raíz / paquetes / clases, 135
Sintaxis,
archivo de procesamiento por lotes, 110
errores y advertencias, 53
revisar, 53

Sintaxis,
revisar sintaxis del proyecto, 402
revisión de la sintaxis: mensajes, 88
Slot,
mostrar / ocultar, 275
Sobrescribir,
código con el modelo, 402
modelo con el código, 402
Socket,
Ball and socket (notación de forma esférica), 275
Software product license, 445
SPL,
plantillas SPL definidas por el usuario, 135
StarTeam, 342
Subclase,
insertar en un diagrama, 98
Subproyecto,
abrir / editar, 334
control de versiones, 332
crear, 334
guardar archivos de subproyectos, 334
opciones para trabajo en equipo, 332
Syntax coloring, 281

T

Tagged values,
show, 317
Tamaño,
panel de diagramas, 89
Tecla de acceso rápido, 419
Tipo,
propiedad (ver), 163
propiedad de tipo línea de vida, 243
Tipo de datos,
definir en esquema, 320
Tipos,
finalización automática, 92
Tipos de datos,
agregar (finalización automática), 92
Tipos de filtros,
ventana de finalización automática, 92
Título del diagrama,
ver, 92
Todos,
expandir / contraer, 275
Trabajo en equipo,
subproyectos, 332
Transición, 207
agregar diagrama de actividades a, 207
definir disparador, 207
definir entre estados, 207
Transparente,
color de fondo, 307
Tres bandas,
fusión del proyecto, 151
fusión manual del proyecto, 153
Tutorial, 8
archivos de ejemplo, 8
carpeta de ejemplos, 8
objetivos, 8

U

Ubicación,
mover proyecto, 118
UML,
compartir diagrama UML, 156
diagramas, 188
iconos de visibilidad, 275
plantillas, 160
ver título del diagrama, 92
UModel,
correspondencia con código C#, 144
correspondencia con código Java, 143
correspondencia con el código de la BD, 147
correspondencia con el código VB.NET, 146
correspondencia con el código XML Schema, 145
importar XMI generado, 328
iniciar, 10
UMP,
cambiar ubicación del proyecto, 118
extensión de archivo, 118
URL,
abrir archivo desde, 393
Uso,
dependencia, 42
Uso de interacción, 248
Usuario,
carpeta de ejemplos multiusuario, 8
UUID,
identificadores únicos universales, 328

V

Valor general de la línea de vida,
diagrama de ciclo de vida, 267
Valores,
etiquetados, 298, 300
etiquetados (mostrar), 313
Valores etiquetados,
mostrar, 313
value,
tagged, show, 317
Variables,
argumentos de herramientas externas, 416
Varios elementos,
ver estilos, 80
VB.NET,
correspondencia entre el modelo y el código, 146
Velocidad,
acelerar actualización de estado en segundo plano, 342
Ventanas, 428
restaurar, 414
Ver, 83
etiquetas de texto, 171
o ajustar a la cuadrícula, 89
propiedad como asociación, 89, 163
relaciones del modelo, 171
todas las relaciones (pestaña Jerarquía), 83
u ocultar nombre de región, 213
varias instancias del elemento, 275
vista en forma de diagrama, 83
Vincular,
crear hipervínculo, 100
elemento de Estructura del modelo, 100
Vínculos,
relativos a la documentación, 176
Visibilidad,
iconos (seleccionar), 275
Vista, 413
Vista en forma de diagrama,
un solo conjunto de relaciones, 83
Vista general,
panel, 86

W

Web,
hipervínculo, 100

X

XMI, 328
extensiones, 328
pretty print, 328
XML Schema,
correspondencia entre el modelo y el esquema, 145

Z

Zoom,
ajustar el tamaño, 89