



**Crear, configurar y ejecutar una transformación de modelo a modelo**



---

## Contenido

### Crear, configurar y ejecutar una transformación de modelo a modelo . . . 1

Introducción: Crear, configurar y ejecutar una transformación de modelo a modelo . . . . . 1

Módulo 1: Crear un proyecto de correlación de transformación de modelo a modelo y perfeccionar la correlación de modelo a modelo. . . . . 3

    Lección 1: Crear un proyecto de correlación de transformación de modelo a modelo. . . . . 3

    Lección 2: Examinar el proyecto de correlación . . . . . 5

    Lección 3: Crear y perfeccionar una declaración de correlación de clase a clase . . . . . 6

    Lección 4: Crear y perfeccionar una declaración de correlación de clase a interfaz . . . . . 10

    Lección 5: Crear y perfeccionar las declaraciones de correlación necesarias para la declaración de correlación de operación a operación . . . . . 12

    Lección 6: Crear y perfeccionar una declaración de correlación de paquete a paquete . . . . . 17

    Lección 7: Crear y perfeccionar una declaración de correlación de modelo a modelo . . . . . 19

Módulo 2: Generar código de transformación y ejecutar la transformación . . . . . 22

    Lección 1: Generar y compilar código fuente de transformación . . . . . 22

    Lección 2: Configurar un entorno de trabajo de tiempo de ejecución . . . . . 23

    Lección 3: Crear una configuración de transformación . . . . . 25

    Lección 4: Ejecutar la transformación . . . . . 26

Resumen: Crear, configurar y ejecutar una transformación de modelo a modelo . . . . . 27



---

# Crear, configurar y ejecutar una transformación de modelo a modelo

En esta guía de aprendizaje se describe cómo crear una transformación de modelo a modelo. Puede crear una transformación de modelo a modelo mediante la creación de un proyecto de correlación de transformación. Una transformación de modelo a modelo transforma un modelo en un modelo con un nivel diferente de abstracción. En esta guía de aprendizaje también se describe cómo crear y perfeccionar los modelos de correlación de un proyecto de correlación y cómo generar y probar el código de transformación de modelo a modelo.

## Objetivos del curso

Debe realizar todas las lecciones de esta guía de aprendizaje en la secuencia especificada. En esta guía de aprendizaje se explica cómo realizar las tareas siguientes:

- Crear un proyecto de correlación de transformación de modelo a modelo.
- Crear y perfeccionar los modelos de correlación en una correlación de transformación de modelo a modelo mediante las herramientas de correlación de transformación.
- Generar y compilar código fuente de transformación de modelo a modelo.
- Configurar y ejecutar la transformación de modelo a modelo en un entorno de trabajo de tiempo de ejecución.

## Tiempo necesario

60 minutos aproximadamente

### Información relacionada

Crear transformaciones de modelo a modelo



Ver la versión PDF

Para ver este archivo, debe tener Adobe Acrobat Reader instalado en el sistema.

---

## Introducción: Crear, configurar y ejecutar una transformación de modelo a modelo

En esta guía de aprendizaje se describe cómo crear, configurar y ejecutar una transformación de modelo a modelo mediante las herramientas de correlación disponibles en los Productos de modelado de IBM Rational. Después de crear un proyecto de correlación de transformación, y tras crear y perfeccionar los modelos de correlación en el proyecto, puede generar el código para una transformación de modelo a modelo. Después de generar el código fuente de transformación, puede configurar y ejecutar la transformación en un entorno de trabajo de tiempo de ejecución.

En esta guía de aprendizaje, se crea una transformación de modelo a modelo que transforma clases en un modelo de origen en interfaces y clases de implementación en un modelo de destino. La clase de implementación generada necesita copias de las operaciones en la clase de origen, y la interfaz generada solo necesita copias de las operaciones públicas de la clase de origen.

Es posible que esta guía de aprendizaje necesite algunos componentes de instalación opcional. Para asegurarse de que ha instalado los componentes opcionales adecuados, consulte la lista de requisitos del sistema.

Una transformación es un tipo de estrategia de implementación de patrón que adopta un elemento de origen, o un conjunto de elementos, y los cambia por un nuevo elemento de destino, o por un conjunto de elementos. Las transformaciones permiten efectuar la transición entre modelos y código, así como entre modelos de diferentes niveles de abstracción. La creación de una transformación es un proceso dirigido por modelo que permite crear transformaciones que contengan información detallada de implementación, o bien crear transformaciones que especifiquen relaciones de correlación entre modelos o metamodelos.

Los proyectos de correlación de transformación de modelo a modelo permiten especificar metamodelos de origen y de destino, así como crear un modelo de correlación que defina las relaciones entre los elementos de los metamodelos. Puede generar incrementalmente código fuente de transformación ampliable que implemente las relaciones. Trabajar con este nivel de abstracción permite centrarse en el dominio del problema en lugar de hacerlo en el dominio de la solución.

El proceso de creación de transformaciones de modelo a modelo consiste en los siguientes pasos de alto nivel:

1. Se crea un proyecto de correlación de transformación de modelo a modelo que contiene un modelo de correlación. Un proyecto de correlación puede contener varios modelos de correlación. Si crea un proyecto de correlación, el servicio de transformación registra una transformación. Cada transformación tiene un proveedor de transformaciones, una transformación denominada MainTransform y una transformación para cada declaración de correlación del proyecto.
2. Se añaden declaraciones de correlación, también denominadas correlaciones, al modelo de correlación. Un modelo de correlación puede contener una o varias declaraciones de correlación.
3. Se añaden reglas de correlación a las declaraciones de correlación de un modelo de correlación.
4. Se genera código fuente de transformación a partir del modelo o modelos de correlación del proyecto de correlación. Las herramientas de creación de transformaciones de modelo a modelo generan una transformación para cada modelo de correlación del proyecto de correlación. Para cada declaración de correlación, las herramientas de creación generan un archivo de origen Java que implementa una transformación. Para cada movimiento o regla de correlación personalizada de una declaración de correlación, se genera una regla en el código fuente de la transformación. Para cada regla de correlación de subcorrelación de una declaración de correlación, se genera un extractor de contenido en el código fuente de la transformación.

## Objetivos del curso

Esta guía de aprendizaje consiste en dos módulos que debe realizar en secuencia. En estos módulos, lleve a cabo las tareas siguientes:

- Cree un proyecto de creación de transformaciones de modelo a modelo que contenga una infraestructura de transformación y un modelo de correlación.
- Cree declaraciones de correlación en el modelo de correlación.
- Perfeccione las declaraciones de correlación efectuando las tareas siguientes:
  - Especifique un objeto de entrada y un objeto de salida para cada declaración de correlación.
  - Cree reglas de correlación definiendo las relaciones entre los atributos de los objetos de entrada y salida en una declaración de correlación.
- Genere y compile el código fuente de transformación.
- Configure un entorno de trabajo de tiempo de ejecución para probar la transformación de modelo a modelo.
- En un entorno de trabajo de tiempo de ejecución, cree y aplique una configuración de transformación que ejecute la transformación de modelo a modelo.

## Tiempo necesario

Esta guía de aprendizaje tiene una duración aproximada de 60 minutos. Si explora otros conceptos relacionados con esta guía de aprendizaje, es posible que tarde más.

## Nivel de habilidad

Avanzado

## Público

Esta guía de aprendizaje va dirigida a desarrolladores.

## Requisitos del sistema

Para completar esta guía de aprendizaje, es necesario tener instalado el componente de creación de transformaciones.

También debe habilitar la posibilidad de modelado.

## Requisitos previos

Para completar esta guía de aprendizaje, debe estar familiarizado con los conceptos siguientes:

- Infraestructura de modelado de Eclipse (EMF)
- Proyectos de plug-ins de Eclipse
- Modelos Ecore
- Entorno de trabajo de Eclipse

---

## Módulo 1: Crear un proyecto de correlación de transformación de modelo a modelo y perfeccionar la correlación de modelo a modelo.

En este módulo, se crea un proyecto de correlación de transformación de modelo a modelo que contiene diversas declaraciones de correlación. En cada declaración de correlación, se crean reglas de correlación que definen cómo se relacionan las características de los elementos de entrada y de salida de la declaración de correlación. También obtendrá información sobre los archivos del proyecto y cómo estos se utilizan para generar código de transformación.

## Objetivos del curso

Las lecciones de este módulo describen módulos de correlación, declaraciones de correlación y reglas de correlación y muestran cómo realizar las tareas siguientes:

- Crear un proyecto de correlación
- Crear y perfeccionar declaraciones de correlación
- Gestionar reglas de correlación

## Tiempo necesario

Este módulo tiene una duración aproximada de 40 minutos.

## Lección 1: Crear un proyecto de correlación de transformación de modelo a modelo.

En esta lección se muestra cómo crear un proyecto de correlación de transformación de modelo a modelo.

Un proyecto de correlación de transformación de modelo a modelo es un plug-in estándar de Eclipse que especifica un proveedor de transformaciones, que es el mecanismo que define las transformaciones. Un proyecto de correlación también contiene un archivo de correlación como mínimo, llamado también un modelo de correlación. Al crear un proyecto de correlación, un modelo de correlación se crea automáticamente en el proyecto.

### Obtener más información sobre proyectos de correlación de transformación:

Los proyectos de correlación de transformación de modelo a modelo, también denominados proyectos de correlación, son plug-ins de Eclipse que amplían el punto de ampliación llamado `com.ibm.xtools.transform.core.transformationProviders`. La creación de transformaciones de modelo a modelo en proyectos de correlación de transformación permite centrarse en la especificación de la relación entre elementos de los modelos, o metamodelos, de origen y de destino seleccionados, en lugar de crear código que represente los detalles de implementación de la transformación.

Un proyecto de correlación puede contener más de un archivo de correlación, también denominado modelo de correlación. Puede generar código fuente de transformación iterativamente al modificar el modelo de correlación. Al generar el código fuente de transformación, una transformación visible externamente, llamada `MainTransform`, se registra automáticamente y se genera el código fuente Java de una transformación para cada declaración de correlación en el modelo de correlación.

Puede especificar uno o varios metamodelos de origen y de destino al crear un proyecto de correlación. Puede especificar metamodelos, que tienen la extensión `.core`, o bien perfiles UML, que tienen la extensión `.epx` o `.uml`. Si especifica metamodelos de origen y de destino al crear un proyecto, las dependencias necesarias se añaden automáticamente al archivo de manifiesto del plug-in. Si añade los metamodelos mediante los mandatos del área de editor después de crear el proyecto de correlación, debe añadir las nuevas dependencias necesarias al archivo de manifiesto del plug-in.

Para crear un proyecto de correlación de transformación de modelo a modelo:

1. Abra la perspectiva Desarrollo de plug-ins: Pulse **Ventana** → **Abrir perspectiva** → **Otras**. En la ventana Abrir perspectiva, pulse **Desarrollo de plug-ins** y pulse **Aceptar**.
2. Pulse **Archivo** → **Nuevo** → **Proyecto**.
3. En el asistente Nuevo proyecto, en la página Seleccionar un asistente, pulse **Creación de transformaciones** y pulse **Proyecto de correlación de transformación de modelo a modelo**.
4. Pulse **Siguiente**.
5. En la página Proyecto de plug-in, en el campo **Nombre de proyecto**, especifique `Generalize_Classes`. Acepte los valores predeterminados para el resto de campos de la página.
6. Pulse **Siguiente**.
7. En la página Contenido de plug-in, revise los valores y pulse **Siguiente**.
8. En la página Plantillas, seleccione **Plug-in con correlación de transformación** en la lista **Plantillas disponibles**.
9. Pulse **Siguiente**.
10. En el campo **Nombre de correlación** de la página Nueva correlación de transformación, si el valor todavía no existe, especifique `Generalize_Classes`. En este campo se especifica el nombre del modelo de correlación, que se encuentra en la carpeta `modelo` y tiene la extensión `.mapping`.
11. En el campo **Nombre de paquete**, si el nombre todavía no existe, especifique `Generalize_classes`. En una lección posterior, al generar el código fuente de transformación, se crea el código fuente Java para las transformaciones en una carpeta denominada `generalize_classes.transforms`.
12. Para especificar el modelo o modelos de entrada y salida, efectúe los pasos siguientes:
  - a. En la página Nueva correlación de transformación, junto al área de modelos de entrada, pulse **Añadir modelo**.

- b. En el recuadro de diálogo Cargar recurso, pulse el botón adecuado para examinar el modelo o modelos. Para esta guía de aprendizaje, pulse **Examinar paquetes registrados**, seleccione la versión más reciente del modelo UML2 que tenga un convenio de denominación de <http://www.eclipse.org/uml2/2.x.y/UML> y pulse **Aceptar**. En este paso se especifica que la transformación acepta un metamodelo UMLecore como origen de la transformación.
- c. En la página Nueva correlación de transformación, junto al área de modelos de salida, pulse **Añadir modelo**.
- d. Repita el paso 12b para especificar que la salida de la transformación sea un modelo del tipo UMLecore.metamodel. En este paso se especifica que la salida de la transformación sea un metamodelo UMLecore.
- e. Pulse **Aceptar**.

Si especifica metamodelos de entrada y de salida al crear un proyecto, las dependencias necesarias se añaden automáticamente al archivo de manifiesto del plug-in. Si añade los metamodelos mediante los mandatos del área de editor después de crear el proyecto de correlación, debe añadir las nuevas dependencias necesarias al archivo de manifiesto del plug-in.

### 13. Pulse **Finalizar**.

Se crea un proyecto de correlación en el espacio de trabajo. En la próxima lección, se examinará la estructura del proyecto de correlación.

## Lección 2: Examinar el proyecto de correlación

Después de crear el proyecto de correlación, puede utilizar la vista Explorador de paquetes para examinar la estructura del proyecto.

Para examinar el contenido del proyecto de correlación:

1. En la vista Explorador de paquetes, expanda el proyecto de correlación **Generalize Classes** y observe los archivos generados.
2. Vaya a la carpeta modelo. Observe cómo esta carpeta contiene un modelo de correlación con el mismo nombre que el proyecto de correlación de transformación; el archivo también tiene la extensión .mapping. Un proyecto de correlación puede contener varios modelos de correlación. Más adelante, en esta guía de aprendizaje, añadirá declaraciones de correlación a este modelo de correlación.

### Obtener más información sobre modelos de correlación:

Los modelos de correlación, también denominados archivos de correlación, son instancias de metamodelos de infraestructura de modelado de Eclipse (EMF), también denominados modelos Ecore, que contienen referencias a los metamodelos que se correlacionan. Al crear un proyecto de correlación, las herramientas de creación crean un modelo de correlación en el proyecto mediante el modelo o modelos de entrada y salida que haya especificado. Un modelo de correlación tiene la extensión .mapping.

Los modelos de correlación se almacenan y se serializan como archivos XML. La vista Problemas muestra información detallada de errores sobre modelos de correlación. En esta vista, efectúe una doble pulsación en un elemento para abrir el modelo de correlación en un editor de texto y ver la línea que contiene el error. Este método de resolución de problemas suele ser más sencillo que la resolución de problemas mediante la visualización del modelo de correlación en el área del editor.

3. En la vista Explorador de paquetes, expanda la carpeta src. La carpeta generalize\_classes.transforms contiene el código fuente para las transformaciones generadas. En este momento, solo existe la transformación predeterminada, denominada MainTransform. Más adelante, en esta guía de aprendizaje, creará declaraciones de correlación y regenerará el código fuente de la transformación. Para cada declaración de correlación del modelo de correlación, la

infraestructura de creación de transformaciones genera una clase Java con el nombre *nTransform*, donde *n* representa el nombre de la declaración de correlación. Colectivamente, estas clases Java componen el código de transformación.

4. En la vista Explorador de paquetes, en la carpeta *modelo*, efectúe una doble pulsación en el modelo de correlación. El editor de correlaciones de transformación, que permite crear declaraciones de correlación y perfeccionar las reglas de correlación en cada declaración de correlación. Efectúe estas tareas en las siguientes lecciones de este módulo.

## Lección 3: Crear y perfeccionar una declaración de correlación de clase a clase

En esta lección se muestra cómo crear una declaración de correlación que especifique una clase UML como el objeto de entrada y el de salida. Esta declaración de correlación de clase a clase contiene reglas de correlación que, al ejecutar la transformación generada, crean una copia de la clase y de sus operaciones en el modelo de origen y coloca la copia en el modelo de destino. También puede crear una declaración de correlación de operación a operación, que invoca la subcorrelación en la declaración de correlación de clase a clase.

Una declaración de correlación, también denominada correlación, especifica cómo crear o actualizar un objeto de salida para un objeto de salida determinada. Las declaraciones de correlación permiten especificar cómo los atributos de un objeto de entrada se corresponden con los atributos de un objeto de salida. Cada declaración de correlación especifica un tipo de entrada y un tipo de salida, que se seleccionan a partir de los metamodelos que se han añadido al modelo de correlación.

Para cada declaración de correlación del modelo de correlación, la infraestructura de creación de transformaciones genera un archivo de origen Java con el nombre *nTransform.java*, donde *n* representa el nombre de la declaración de correlación. Colectivamente, estos archivos Java componen el código de transformación. Además de generar el código de implementación para la transformación, la infraestructura de creación de transformaciones también genera código para registrar la transformación en el servicio de transformación. Después de crear una declaración de correlación, puede añadir reglas de correlación y generar el código fuente o implementaciones incrementalmente para las reglas de correlación. No debe definir todas las reglas de correlación para poder generar el código fuente.

### Obtener más información sobre declaraciones de correlación:

Las declaraciones de correlación suelen seguir el convenio de denominación *x2y*, donde *x* representa el tipo de objeto de entrada e *y* representa el tipo de objeto de salida. Por ejemplo, una declaración de correlación denominada *Package2EPackage* especifica una declaración de correlación que tiene un paquete como objeto de entrada, y *EPackage* como objeto de salida.

Una correlación entre elementos establece la correspondencia entre sus atributos, lo que permite el intercambio de datos entre ellos. La mayoría de correlaciones proporcionan la capacidad de seguir manipulando los datos entre el origen y el destino. Por ejemplo, podría especificar cálculos o efectuar otras modificaciones en los datos mediante la creación de código personalizado, lo que permite asignar valores al destino.

Para crear una declaración de correlación de clase a clase en el modelo de correlación:

1. Si todavía no está abierto, en la carpeta *modelo* de la vista Explorador de paquetes, efectúe una doble pulsación en el archivo *.mapping*.
2. En la sección Raíz de correlación del editor de correlaciones de transformación, efectúe una pulsación con el botón secundario del ratón en **Generalize\_Classes** y, a continuación, pulse **Crear correlación**.
3. En la ventana Nueva correlación, en el campo **Nombre de correlación**, especifique **Class2Class** y, a continuación, pulse **Aceptar**. La correlación se visualiza en la vista Esquema y se abre en el área del editor bajo la sección Raíz de correlación.

## Añadir objetos de entrada y salida en la declaración de correlación de clase a clase

Después de crear la declaración de correlación, debe añadir un objeto de entrada y un objeto de salida a este. En esta lección, especifique una clase UML como el objeto de entrada y salida.

**Nota:** Al crear transformaciones de modelo a modelo en añadir un tipo de objeto que no está disponible en el metamodelo de entrada o salida utilizado por la correlación, puede añadir los perfiles UML o metamodelos Ecore adecuados al ámbito de la correlación., en la ventana Añadir entrada, pulse **Añadir modelo** y especifique el modelo adecuado.

Para añadir un objeto de entrada y un objeto de salida en la declaración de correlación de clase a clase:

1. Pulse el icono **Añadir un objeto de entrada**, que es el icono situado más a la izquierda en la barra de herramientas para la correlación que se edita:



2. En la ventana Añadir elemento, en el panel Elemento, seleccione un objeto de metamodelo. En el panel Elemento se visualizan los elementos del metamodelo o metamodelos especificados como origen o destino para el modelo de correlación. Para esta guía de aprendizaje, en el panel Elemento, expanda **uml**, pulse **Clase** y pulse **Aceptar**.
3. Pulse el icono **Añadir un objeto de salida**, que es el segundo icono de la izquierda en la barra de herramientas para la correlación que se edita:



4. En la ventana Añadir salida, en el panel Elemento, expanda **uml**, pulse **Clase** y pulse **Aceptar**.
5. Pulse **Archivo** → **Guardar**.

Puede definir reglas de correlación entre los atributos de los objetos de entrada y salida de la clase.

## Definir reglas de correlación entre atributos de los objetos de entrada y salida de la clase

Después de añadir objetos de entrada y salida en la declaración de correlación, puede definir las reglas de correlación entre los atributos. Las reglas de correlación, también llamadas correlaciones, especifican cómo asignar un valor a un atributo de un objeto de salida, en función de los valores de los atributos de un objeto de entrada.

En la declaración de correlación de clase a clase, efectúe los pasos siguientes:

- Cree una regla de correlación de movimiento que cree una clase en el modelo de destino.  
La regla de correlación se crea entre el atributo nombre de los objetos de entrada y de destino. La clase de destino tiene el mismo nombre que la clase en el modelo de entrada; también podría entenderse como la creación de una copia de la clase. En una lección posterior, se añadirán operaciones de correlación en la regla de correlación.
- Cree una regla de correlación de subcorrelación que, para cada operación de la clase, cree una operación correspondiente en la clase del modelo de destino.  
La regla de correlación de subcorrelación se crea entre el atributo `ownedOperation` de los objetos de entrada y de destino. Para cada operación de la recopilación `ownedOperation`, la operación generada del modelo de destino tiene el mismo nombre y visibilidad que la operación del modelo de entrada.

**Obtener más información sobre reglas de correlación:**

Las reglas de correlación, también llamadas correlaciones, especifican cómo asignar un valor a un atributo de un objeto de salida, en función de los valores de los atributos de un objeto de entrada. Puede crear diversas reglas de correlación entre el objeto de entrada y el de salida y, como se muestra en una lección posterior de esta guía de aprendizaje, también puede crear diversas reglas de correlación entre un atributo de un objeto de entrada y de un objeto de salida. Puede crear los siguientes tipos de reglas de correlación:

### **Movimiento**

Un movimiento, también denominado regla de correlación simple, es el tipo más básico de regla de correlación. Los atributos de entrada y salida han de ser tipos de datos compatibles. Ambos o ninguno de los atributos de entrada y salida tienen diversos valores. Por ejemplo, seleccione esta opción si un atributo del objeto de salida es una serie y un atributo del objeto de entrada puede convertirse en una serie sin código personalizado. Este tipo de regla de correlación soporta una correlación entre un elemento o atributo de origen y un elemento o atributo de destino. El código fuente de transformación que se genera para una regla de correlación de movimiento implementa una regla que copia el valor de un atributo de entrada en un atributo de salida.

### **Subcorrelación**

Una subcorrelación es una invocación de una correlación desde dentro de otra correlación. La subcorrelación que se invoca puede estar definida, aunque no es necesario, en el mismo archivo de correlación que la correlación que la invoca. Una subcorrelación permite correlacionar un tipo complejo del modelo de entrada con un tipo complejo del modelo de salida. La subcorrelación que se crea puede invocar una correlación que exista en cualquier archivo de correlación. La definición de subcorrelaciones en diferentes archivos de correlación fomenta la reutilización de correlaciones; no obstante, la creación de diversos archivos de correlación podría requerir un mayor mantenimiento del proyecto. Las subcorrelaciones también pueden incluir otras subcorrelaciones, lo que resulta en una estructura jerárquica.

Las reglas de correlación de subcorrelación soportan los siguientes tipos de correlaciones:

- Correlaciones de 1 a 1 entre objetos de entrada y objetos de salida, o bien entre los atributos de los objetos de entrada y salida.
- Correlaciones de 1 a  $m$  o de  $m$  a 1 entre atributos de los objetos de entrada y salida.
- Correlaciones de  $m$  a  $n$  entre atributos de los objetos de entrada y salida.

Si la regla de subcorrelación especifica una correlación de 1 a  $m$ , la transformación generada añade un objeto de un atributo singleton a una lista; para una correlación de  $m$  a 1, la transformación extrae un objeto de una lista y lo inserta en un atributo singleton.

También puede crear reglas de correlación de subcorrelación entre los objetos de entrada y los objetos de salida en una declaración de correlación.

Para cada subcorrelación de una declaración de correlación, se genera un extractor denominado `getCaracterísticaEntradaToCaracterísticaSalida_UsingCorrelación_Extractor` en la transformación contenedora, donde *CaracterísticaEntrada* representa el nombre del atributo de entrada, *CaracterísticaSalida* representa el nombre del atributo de salida y *Correlación* representa el nombre de la declaración de correlación.

### **Personalizado**

Este tipo de regla de correlación: permite especificar código personalizado que calcula el valor de una propiedad de salida. Por ejemplo, seleccione este tipo de correlación para establecer el valor de una propiedad del objeto de salida como igual a la concatenación de varias propiedades de un objeto de entrada.

Puede especificar refinamientos semánticos mediante la API OCL (Object Constraint Language) proporcionada por Eclipse.

Las reglas de correlación personalizadas soportan los siguientes tipos de correlaciones:

- Correlaciones de 1 a  $n$  entre objetos de entrada y de salida, o bien entre atributos de los objetos de entrada y salida.

- Correlaciones de  $m$  a  $n$  entre objetos de entrada y de salida, o bien entre atributos de los objetos de entrada y salida.
- Correlaciones de  $m$  a 1, donde  $m$  a 1 representa una de las correlaciones siguientes:
  - Una correlación de un único atributo de entrada, cuya multiplicidad se establece en  $m$ , con un único atributo de salida, cuya multiplicidad se establece en 1.
  - Una correlación de diversos atributos de entrada con un único atributo de salida.
  - Una correlación de diversos objetos de entrada con un único objeto de salida.

### Correlaciones heredadas

Puede crear esta regla de correlación sólo entre un objeto de entrada y un objeto de salida.

Una declaración de correlación hereda las reglas de correlación que se definen en la declaración de correlación heredada. Puede alterar temporalmente las reglas de correlación mediante la definición de una regla de correlación que tenga las características siguientes:

- La propiedad del objeto de entrada y la propiedad del objeto de salida son las mismas que la regla de correlación heredada.
- Las reglas de correlación heredadas y que efectúan la alteración temporal son correlaciones de subcorrelaciones que tienen extractores coincidentes, o bien las reglas de correlación heredadas y que efectúan la alteración temporal son correlaciones de movimiento o correlaciones personalizadas.

Si crea una regla de correlación de correlaciones heredadas, debe especificar la declaración de correlación que contenga las reglas de correlación que desea heredar. No puede especificar más de una correlación heredada en una declaración de correlación.

Una regla de correlación heredada, así como la regla o extractor generado a partir de esta correlación, mantiene la misma posición relativa en el orden de proceso que la regla de correlación alterada temporalmente y su regla o extractor generado.

Para cada movimiento o regla de correlación personalizada de una declaración de correlación, se añade una regla al código fuente de la transformación generada. Para cada regla de correlación de subcorrelación, se genera un extractor de contenido en el código fuente de la transformación. Al crear una regla de correlación, su tipo viene determinado por los atributos de entrada y salida que haya seleccionado. Por ejemplo, si los atributos de entrada y salida son tipos primitivos compatibles, como series o enteros, se especifica una regla de correlación de movimiento. Si los atributos de entrada y salida son tipos complejos, se especifica una regla de subcorrelación. Se especifica una regla de correlación personalizada si los tipos de regla de correlación de movimiento y de subcorrelación no son adecuadas.

Para definir las reglas de correlación en la declaración de correlación Class2Class:

1. Cree una regla de correlación de movimiento entre el atributo nombre del objeto de entrada y salida:
  - a. En el área del editor, en el objeto de entrada de clase, pulse el atributo nombre.
  - b. Arrastre el manejador del atributo nombre al atributo nombre en el objeto de salida de clase.
2. Cree una regla de correlación de subcorrelación entre el atributo ownedOperation del objeto de entrada y de salida.
  - a. En el área del editor, en el objeto de entrada de clase, pulse el atributo ownedOperation.
  - b. Arrastre el manejador del atributo ownedOperation al atributo ownedOperation en el objeto de salida de clase. Dado que el atributo ownedOperation es una recopilación, se crea una regla de correlación de subcorrelación por omisión.
3. Pulse **Archivo** → **Guardar**.

Ahora puede crear la declaración de correlación de operación a operación.

## Crear una declaración de correlación de operación a operación

En este punto de la lección, en el área del editor, la subcorrelación tiene un adorno con una X dentro de un círculo rojo, que indica un error. Coloque el puntero del ratón encima de este adorno para ver el mensaje de error. El mensaje de error indica que debe seleccionar una declaración de correlación para la regla de correlación de subcorrelación que desee invocar. Para resolver este error, cree una declaración de correlación de operación a operación.

Para crear una declaración de correlación de operación a operación:

1. Si la vista Propiedades no está abierta, en el área del editor, efectúe una pulsación con el botón secundario del ratón en el elemento de subcorrelación creado en el paso 2 de la sección anterior y pulse **Mostrar en Propiedades**.
2. En la pestaña **Detalles** de la vista Propiedades, junto al campo **Correlación**, pulse **Nueva**.
3. En el campo **Nombre de campo** de la ventana Nueva correlación, especifique `Operation2Operation`.
4. Si el campo **Entrada** no contiene un valor de Operación, pulse **Examinar** y, en la ventana Añadir entrada, en el panel Elemento, pulse **Operación**.
5. Si el campo **Salida** no contiene un valor de Operación, pulse **Examinar** y, en la ventana Añadir salida, en el panel Elemento, pulse **Operación**.
6. Pulse **Aceptar**. Se elimina el adorno de error y se visualiza la declaración de correlación de operación a operación en la vista Esquema. En una lección posterior, se crearán reglas de correlación en esta declaración de correlación.
7. Pulse **Archivo** → **Guardar**.

En la próxima lección, se definirá una declaración de correlación de clase a interfaz.

## Lección 4: Crear y perfeccionar una declaración de correlación de clase a interfaz

En esta lección se muestra cómo crear una declaración de correlación de clase a interfaz en el modelo de correlación. Esta declaración de correlación contiene reglas de correlación que crean una interfaz cuyo nombre se deriva del nombre de la clase en el modelo de origen y que solo copian los métodos públicos de la clase en el modelo de origen.

Para crear una declaración de correlación de clase a interfaz en el modelo de correlación:

1. Si el archivo `Generalize_Classes.mapping` no está abierto en el editor de correlaciones, en la vista Explorador de paquetes, efectúe una doble pulsación en el archivo.
2. En la sección Raíz de correlación del editor de correlaciones de transformación, efectúe una pulsación con el botón secundario del ratón en **Generalize\_Classes** y, a continuación, pulse **Crear correlación**.
3. En la ventana Nueva correlación, en el campo **Nombre de correlación**, especifique `Class2Interface` y, a continuación, pulse **Aceptar**. La correlación se visualiza en la vista Esquema y se abre en el área del editor bajo la sección Raíz de correlación.

**Nota:** Para ver los detalles de una declaración de correlación que no esté abierto en el área del editor de la vista Esquema, expanda el nombre de la declaración de correlación. Para abrir una declaración de correlación diferente, en la vista Esquema, pulse el nombre de la declaración de correlación.

### Añadir objetos de entrada y salida en la declaración de correlación de clase a interfaz

Después de crear la declaración de correlación de clase a interfaz, especifique una clase UML como objeto de entrada y una interfaz UML como objeto de salida.

Para añadir un objeto de entrada y un objeto de salida en la declaración de correlación de clase a interfaz:

1. Pulse el icono **Añadir un objeto de entrada**, que es el icono situado más a la izquierda en la barra de herramientas para la correlación que se edita.
2. En la ventana Añadir entrada, en el panel Elemento, expanda **uml**, pulse **Clase** y pulse **Aceptar**.
3. Pulse el icono **Añadir un objeto de salida**, que es segundo icono de la izquierda en la barra de herramientas para la correlación que se edita.
4. En la ventana Añadir salida, en el panel Elemento, expanda **uml**, pulse **Interfaz** y pulse **Aceptar**.
5. Pulse **Archivo** → **Guardar**.

Puede definir reglas de correlación entre los atributos del objeto de entrada de la clase y el objeto de salida de la interfaz.

## Definir reglas de correlación entre los atributos de los objetos de entrada y salida

Después de añadir un objeto de entrada y salida en la declaración de correlación de clase a interfaz, puede definir las reglas de correlación entre los atributos de los objetos de entrada y salida.

Para esta lección, puede crear una regla de correlación personalizada que cree una interfaz denominada *INombreClase*. Cree también una regla de correlación de subcorrelación que invoque la subcorrelación de operación a operación creada en “Lección 3: Crear y perfeccionar una declaración de correlación de clase a clase” en la página 6 y añada un filtro de entrada en la regla de correlación de subcorrelación para especificar que solo las operaciones públicas se transformen en la interfaz de destino.

Para definir las reglas de correlación entre los atributos de los objetos de entrada y salida:

1. Cree una regla de correlación personalizada entre el atributo name del objeto de entrada y salida:
  - a. En el área del editor, en el objeto de entrada de clase, pulse el atributo nombre.
  - b. Arrastre el manejador del atributo name al atributo name en el objeto de salida de la interfaz. Se crea una regla de correlación de movimiento.
  - c. En el elemento **Movimiento** que conecta los atributos name, pulse la tecla de flecha abajo y, a continuación, pulse **Personalizar**.
  - d. Efectúe una pulsación con el botón secundario del ratón en el elemento **Personalizar**; a continuación, pulse **Mostrar en Propiedades**.
  - e. En la vista Propiedades, en la pestaña **Detalles**, pulse **Incorporado** y, en el área de texto bajo la signatura de método, escriba el código siguiente:  

```
Interface_tgt.setName("I"+Class_src.getName());
```

**Nota:** Para invocar la funcionalidad del asistente para contenido a medida que escribe el código, pulse **Control+Barra espaciadora**.
  - f. Pulse **Aplicar**.
2. Cree una regla de correlación de subcorrelación entre el atributo ownedOperation del objeto de entrada y de salida.
  - a. En el área del editor, en el objeto de entrada de clase, pulse el atributo ownedOperation.
  - b. Arrastre el manejador del atributo ownedOperation al atributo ownedOperation en el objeto de salida de la interfaz. Dado que el atributo ownedOperation es una recopilación, se crea una regla de correlación de subcorrelación por omisión.
  - c. Si la vista Propiedades no está abierta, efectúe una pulsación con el botón secundario del ratón en el nuevo elemento **Subcorrelación**; a continuación, pulse **Mostrar en Propiedades**.
  - d. Si todavía no está seleccionado, en la pestaña **Detalles** de la vista Propiedades, seleccione **Operation2Operation** en la lista **Correlación**.
3. En la regla de correlación de subcorrelación creada en el paso 2, añada un filtro de entrada que sólo transforme las operaciones públicas en la interfaz de destino:
  - a. En la vista Propiedades, en la pestaña **Filtro de entrada**, pulse **Filtrar elementos de entrada** y pulse **Incorporado**.

- b. En el área de texto bajo la opción **Código**, bajo la signature de método, escriba el código siguiente:
- ```
if (ownedOperation_src.getVisibility().equals(VisibilityKind.PUBLIC_LITERAL)) { return true; } return false;
```

**Nota:** El código especificado para un filtro de entrada o salida debe devolver un valor booleano. Puede especificar sólo el cuerpo del método; la infraestructura de creación de transformaciones define la signature de método. Para ver una lista de nombres válidos de variables, pulse **Control+Barra espaciadora** en el área de texto bajo el botón **Incorporado**.

- c. Pulse **Aplicar**.

4. Pulse **Archivo → Guardar**.

En la lección siguiente, se crean varias declaraciones de correlación en el modelo `Generalize_Classes.mapping`; estas declaraciones de correlación definen cómo se transforman los parámetros y los tipos primitivos al ejecutar la transformación generada.

## Lección 5: Crear y perfeccionar las declaraciones de correlación necesarias para la declaración de correlación de operación a operación

En esta lección se muestra cómo crear las declaraciones de correlación necesarias para la declaración de correlación de operación a operación para transformar elementos en una operación. Por ejemplo, cree una declaración de correlación que transforme los parámetros del modelo de origen en parámetros del modelo de destino; cree también una declaración de correlación que defina cómo transformar tipos primitivos del modelo de origen en tipos primitivos del modelo de destino. En esta lección, cree también las reglas de correlación para la declaración de correlación de operación a operación que invocan las declaraciones de correlación creadas en esta lección.

En la tabla siguiente se listan las declaraciones de correlación y las reglas de correlación creadas en esta lección:

| Declaración de correlación | Tipo de objeto de entrada y salida | Atributos correlacionados | Tipo de regla de correlación | Descripción de regla de correlación                                                                                                |
|----------------------------|------------------------------------|---------------------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Primitive2Primitive        | Tipo primitivo UML                 | nombre                    | movimiento                   | Crea un tipo primitivo en el elemento cuya subcorrelación invoque la declaración de correlación <code>Primitive2Primitive</code> . |

| Declaración de correlación | Tipo de objeto de entrada y salida | Atributos correlacionados | Tipo de regla de correlación | Descripción de regla de correlación                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------------|------------------------------------|---------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter2Parameter        | Parámetro UML                      | nombre, visibilidad       | movimiento                   | Crea un parámetro en el elemento cuya subcorrelación invoque la declaración de correlación Parameter2Parameter; el parámetro generado tiene el mismo nombre y visibilidad que el parámetro del elemento de entrada de origen.                                                                                                                                                                                      |
|                            |                                    | tipo                      | subcorrelación               | Para cada tipo en el elemento cuya subcorrelación invoque esta declaración de correlación, estos son los resultados: <ul style="list-style-type: none"> <li>• Si el parámetro es de tipo primitivo, esta regla invoca la declaración de correlación Primitive2Primitive.</li> <li>• Si el parámetro es de tipo clase, esta regla invoca las declaraciones de correlación Class2Class y Class2Interface.</li> </ul> |
| Operation2Operation        | Operación UML                      | nombre, visibilidad       | movimiento                   | Crea una operación en el elemento cuya subcorrelación invoque la declaración de correlación Operation2Operation; la operación generada tiene el mismo nombre y visibilidad que la operación del elemento de entrada de origen.                                                                                                                                                                                     |
|                            |                                    | ownedParameter            | subcorrelación               | Para cada parámetro de la operación, esta subcorrelación invoca la declaración de correlación Parameter2Parameter.                                                                                                                                                                                                                                                                                                 |

En esta guía de aprendizaje, la declaración de correlación `Parameter2Parameter` invoca la declaración de correlación `Primitive2Primitive` para crear parámetros de tipo primitivo.

Para crear una declaración de correlación de tipo primitivo a tipo primitivo en el modelo de correlación:

1. Si no está abierto, en la carpeta `modelo` de la vista Explorador de paquetes, efectúe una doble pulsación en el archivo `Generalize_Classes.mapping`.
2. En la sección Raíz de correlación del editor de correlaciones de transformación, efectúe una pulsación con el botón secundario del ratón en **Generalize\_Classes** y, a continuación, pulse **Crear correlación**.
3. En la ventana Nueva correlación, en el campo **Nombre de correlación**, especifique `Primitive2Primitive` como nombre de la correlación y, a continuación, pulse **Aceptar**. La correlación se visualiza en la vista Esquema y se abre bajo la Raíz de correlación en el área del editor.

## Añadir objetos de entrada y salida en la declaración de correlación `Primitive2Primitive`

Después de crear la declaración de correlación `Primitive2Primitive`, debe añadir un objeto de entrada y un objeto de salida a este. En esta lección, especifique un tipo primitivo UML como el objeto de entrada y salida.

Para añadir un objeto de entrada y un objeto de salida en la declaración de correlación `Primitive2Primitive`:

1. Pulse el icono **Añadir un objeto de entrada**, que es el icono situado más a la izquierda en la barra de herramientas para la correlación que se edita.
2. En la ventana Añadir entrada, en el panel Elemento, expanda **uml**, pulse **TipoPrimitivo** y pulse **Aceptar**.
3. Pulse el icono **Añadir un objeto de salida**, que es segundo icono de la izquierda en la barra de herramientas para la correlación que se edita.
4. En la ventana Añadir salida, en el panel Elemento, expanda **uml**, pulse **TipoPrimitivo** y pulse **Aceptar**.
5. Pulse **Archivo** → **Guardar**.

Ahora puede definir una regla de correlación entre el atributo nombre de los objetos de entrada y de destino `TipoPrimitivo`.

## Definir reglas de correlación entre atributos de los objetos de entrada y salida `TipoPrimitivo`

Para esta lección, cree una regla de correlación de movimiento que cree un tipo primitivo en el modelo de destino. El tipo primitivo generado tiene el mismo nombre que el tipo primitivo en el modelo de entrada; también podría entenderse como la creación de una copia del tipo primitivo.

Para definir una regla de correlación de movimiento que defina una relación entre el atributo nombre de los objetos de entrada y salida `TipoPrimitivo`:

1. En el área del editor, en el objeto de entrada `TipoPrimitivo`, pulse el atributo nombre.
2. Arrastre el manejador del atributo nombre al atributo nombre en el objeto de salida.
3. Pulse **Archivo** → **Guardar**.

Ahora puede crear una declaración de correlación de parámetro a parámetro.

## Crear una declaración de correlación de parámetro a parámetro

En esta sección se muestra cómo crear una declaración de correlación que especifique un parámetro UML como el objeto de entrada y el de salida. Esta declaración de correlación de parámetro a parámetro contiene reglas de correlación que, al ejecutar la transformación generada, crean un parámetro en el

modelo de destino cuyo nombre, visibilidad y tipo son los mismos que los del parámetro en el elemento cuya subcorrelación haya invocado esta declaración de correlación. En esta guía de aprendizaje, la declaración de correlación `Operation2Operation` contiene una regla de correlación de subcorrelación que invoca esta declaración de correlación.

Para crear una declaración de correlación de parámetro a parámetro:

1. Si no está abierto, en la carpeta `modelo` de la vista Explorador de paquetes, efectúe una doble pulsación en el archivo `Generalize_Classes.mapping`.
2. En la sección Raíz de correlación del editor de correlaciones de transformación, efectúe una pulsación con el botón secundario del ratón en **Generalize\_Classes** y, a continuación, pulse **Crear correlación**.
3. En la ventana Nueva correlación, en el campo **Nombre de correlación**, especifique `Parameter2Parameter` y, a continuación, pulse **Aceptar**. La correlación se visualiza en la vista Esquema y se abre en el área del editor bajo Raíz de correlación.

## **Añadir objetos de entrada y salida en la declaración de correlación `Parameter2Parameter`**

Después de crear la declaración de correlación, debe añadir un objeto de entrada y un objeto de salida a este. En esta lección, especifique un parámetro UML como el objeto de entrada y salida.

Para añadir un objeto de entrada y un objeto de salida en la declaración de correlación `Parameter2Parameter`:

1. Pulse el icono **Añadir un objeto de entrada**, que es el icono situado más a la izquierda en la barra de herramientas para la correlación que se edita.
2. En la ventana Añadir entrada, en el panel Elemento, expanda **uml**, pulse **Parámetro** y pulse **Aceptar**.
3. Pulse el icono **Añadir un objeto de salida**, que es segundo icono de la izquierda en la barra de herramientas para la correlación que se edita.
4. En la ventana Añadir salida, en el panel Elemento, expanda **uml**, pulse **Parámetro** y pulse **Aceptar**.
5. Pulse **Archivo** → **Guardar**.

Puede definir reglas de correlación entre los atributos de los objetos de entrada y salida del parámetro.

## **Definir reglas de correlación en la declaración de correlación `Parameter2Parameter`**

Después de añadir objetos de entrada y salida en la declaración de correlación, puede definir las reglas de correlación entre los atributos.

En esta sección, cree reglas de subcorrelación y una regla de correlación de movimiento que, al ejecutar la transformación generada, creen un parámetro en el modelo de destino cuyo nombre, visibilidad y tipo son los mismos que los del parámetro en un elemento del modelo de origen. En esta guía de aprendizaje, la declaración de correlación `Operation2Operation` invoca esta declaración de correlación.

Para definir las reglas de correlación en la declaración de correlación `Parameter2Parameter`:

1. Cree una regla de correlación de movimiento entre el atributo nombre del objeto de entrada y salida:
  - a. En el área del editor, en el objeto de entrada de parámetro, pulse el atributo nombre.
  - b. Arrastre el manejador del atributo nombre al atributo nombre en el objeto de salida.
2. Repita los pasos 1a y 1b, pero sustituya nombre por visibilidad.
3. Cree una regla de subcorrelación entre el atributo tipo del objeto de entrada y salida:
  - a. En el área del editor, en el objeto de entrada de parámetro, pulse el atributo tipo.
  - b. Arrastre el manejador del atributo tipo al atributo tipo en el objeto de salida. Dado que este atributo es un tipo complejo, se crea una subcorrelación por omisión.

- c. Si la vista Propiedades no está abierta, en el área del editor, efectúe una pulsación con el botón secundario del ratón en el nuevo elemento **Subcorrelación**; a continuación, pulse **Mostrar en Propiedades**.
- d. En la pestaña **Detalles** de la vista Propiedades, seleccione **Primitive2Primitive** en la lista **Correlación**.

Esta regla de correlación genera código que invoca la transformación Primitive2Primitive cuando se encuentra un parámetro de primitivo en el modelo de origen. Si el parámetro del modelo de origen no es de tipo primitivo, el parámetro no se transforma en el modelo de destino.

4. Repita los pasos 3a, 3b y 3c y, a continuación, realice el paso siguiente:
  - En la pestaña **Detalles** de la vista Propiedades, seleccione **Class2Class** en la lista **Correlación**.

Esta regla de correlación de subcorrelación genera código que crea un parámetro de tipo clase en un elemento del modelo de destino, si el parámetro de origen es de tipo clase.

5. Repita los pasos 3a, 3b y 3c y, a continuación, realice el paso siguiente:
  - En la pestaña **Detalles** de la vista Propiedades, seleccione **Class2Interface** en la lista **Correlación**.

Esta regla de correlación de subcorrelación crea un parámetro de tipo interfaz en un elemento del modelo de destino, si el parámetro de origen es de tipo clase.

6. Pulse **Archivo** → **Guardar**.

Puede definir las reglas de correlación en la declaración de correlación de operación a operación creada en “Lección 3: Crear y perfeccionar una declaración de correlación de clase a clase” en la página 6.

## Definir reglas de correlación en la declaración de correlación Operation2Operation

En “Lección 3: Crear y perfeccionar una declaración de correlación de clase a clase” en la página 6 se ha mostrado cómo crear una declaración de correlación de operación a operación. En esta sección, ahora está listo para crear reglas de correlación en esta declaración de correlación. Al ejecutar la transformación generada, estas reglas de correlación crean una operación en el modelo de destino cuyo nombre, visibilidad y parámetros son los mismos que los de la operación en el modelo de origen. En esta guía de aprendizaje, las declaraciones de correlación Class2Class y Class2Interface invocan esta declaración de correlación.

Para definir las reglas de correlación en la declaración de correlación Operation2Operation:

1. Si no está abierto, en la carpeta modelo de la vista Explorador de paquetes, efectúe una doble pulsación en el archivo Generalize\_Classes.mapping.
2. En la vista Esquema, pulse la declaración de correlación **Operation2Operation**.
3. Cree una regla de correlación de movimiento entre el atributo nombre del objeto de entrada y salida:
  - a. En el área del editor, en el objeto de entrada de la operación, pulse el atributo nombre.
  - b. Arrastre el manejador del atributo nombre al atributo nombre en el objeto de salida.
4. Repita los pasos 3a y 3b, pero sustituya nombre por visibilidad.
5. Cree una regla de subcorrelación entre el atributo ownedParameter del objeto de entrada y salida:
  - a. En el área del editor, en el objeto de entrada de la operación, pulse el atributo ownedParameter.
  - b. Arrastre el manejador del atributo ownedParameter al atributo ownedParameter en el objeto de salida. Dado que este atributo es una recopilación, se crea una subcorrelación por omisión.
  - c. Si la vista Propiedades no está abierta, en el área del editor, efectúe una pulsación con el botón secundario del ratón en el nuevo elemento **Subcorrelación**; a continuación, pulse **Mostrar en Propiedades**.
  - d. En la pestaña **Detalles** de la vista Propiedades, seleccione **Parameter2Parameter** en la lista **Correlación**.
6. Pulse **Archivo** → **Guardar**.

En la próxima lección, se creará una declaración de correlación de paquete a paquete.

## Lección 6: Crear y perfeccionar una declaración de correlación de paquete a paquete

En esta lección se muestra cómo crear una declaración de correlación de paquete a paquete y diversas reglas de correlación. Las reglas de correlación especifican cómo la transformación generada procesa los paquetes anidados o los elementos de clase en el paquete contenido en el modelo de origen.

Los objetos de entrada y salida del paquete contienen una característica llamada `packagedElement`. Esta característica es una recopilación que contiene diferentes tipos de objetos UML válidos. Las reglas de correlación creadas en esta lección definen cómo la transformación procesa los elementos de la recopilación que sean del tipo paquete o clase.

En esta lección se crean las reglas de correlación siguientes:

- Una regla de correlación de movimiento que crea un paquete en el proyecto de destino; este paquete tiene el mismo nombre que el paquete del modelo de origen.
- Una regla de correlación de subcorrelación que invoca la transformación `Class2ClassTransform` si el elemento es del tipo clase.
- Una regla de correlación de subcorrelación que invoca la transformación `Class2InterfaceTransform` si el elemento es del tipo clase.
- Una regla de correlación de subcorrelación que invoca la transformación `Package2PackageTransform` si el elemento es del tipo paquete.

En “Lección 3: Crear y perfeccionar una declaración de correlación de clase a clase” en la página 6 se ha mostrado que para cada regla de correlación de movimiento de una declaración de correlación, se añade al código fuente de transformación generado una regla que copia un valor de atributo desde el modelo de origen al modelo de destino. Para cada regla de correlación de subcorrelación, se genera en el código fuente de transformación un extractor que extrae elementos de la recopilación especificada. La regla de subcorrelación se aplica a un objeto si el objeto de entrada actual es una instancia del tipo de entrada que se define en la declaración de correlación.

Al ejecutar la transformación generada, si el modelo de origen contiene un paquete, se invoca la transformación `Package2PackageTransform` y se crea un paquete con el nombre de guardado en el modelo de destino. La transformación `Package2PackageTransform` cruza la recopilación de la característica `packagedElement`. La transformación invoca la transformación `Package2PackageTransform` para cada elemento de la recopilación que sea del tipo paquete, es decir, que el modelo de origen contenga paquetes anidados. Para cada elemento de la recopilación que sea del tipo clase, la transformación invoca las reglas par transformar la clase en una clase e interfaz correspondiente en el modelo de salida de destino.

Para crear una declaración de correlación de paquete a paquete en el modelo de correlación:

1. Si el archivo `Generalize_Classes.mapping` no está abierto en el editor de correlaciones de transformación, en la vista Explorador de paquetes, efectúe una doble pulsación en el archivo.
2. En la sección Raíz de correlación del editor de correlaciones de transformación, efectúe una pulsación con el botón secundario del ratón en **Generalize\_Classes** y, a continuación, pulse **Crear correlación**.
3. En la ventana Nueva correlación, en el campo **Nombre de correlación**, especifique `Package2Package` y, a continuación, pulse **Aceptar**. La correlación se visualiza en la vista Esquema y se abre en el área del editor bajo Raíz de correlación.

### Añadir objetos de entrada y salida en la declaración de correlación de paquete a paquete

Después de crear la declaración de correlación, debe añadir un objeto de entrada y un objeto de salida a este. En esta lección, especifique un paquete UML como el objeto de entrada y salida.

Para añadir un objeto de entrada y un objeto de salida en la declaración de correlación de paquete a paquete:

1. Pulse el icono **Añadir un objeto de entrada**, que es el icono situado más a la izquierda en la barra de herramientas para la correlación que se edita.
2. En la ventana Añadir entrada, en el panel Elemento, expanda **uml**, pulse **Paquete** y pulse **Aceptar**.
3. Pulse el icono **Añadir un objeto de salida**, que es segundo icono de la izquierda en la barra de herramientas para la correlación que se edita.
4. En la ventana Añadir salida, en el panel Elemento, expanda **uml**, pulse **Paquete** y pulse **Aceptar**.
5. Pulse **Archivo** → **Guardar**.

Puede definir las reglas de correlación entre las características de los objetos de entrada y salida.

## Definir reglas de correlación entre atributos de los objetos de entrada y salida del paquete

Después de añadir los objetos de entrada y salida del paquete en la declaración de correlación, puede crear las reglas de correlación entre los atributos de los objetos de entrada y salida.

**Nota:** Al generar el código fuente de transformación, se genera una transformación para cada declaración de correlación. Cada transformación generada contiene un método `getAccept_Condition()`. Por omisión, este método devuelve `true` si el elemento de entrada actual es una instancia del objeto de entrada que se define en la declaración de correlación. Si el método devuelve `false`, la transformación no procesa el elemento de entrada actual. En esta lección, puede crear filtros de entrada que también verifiquen si el objeto de entrada es una instancia del tipo de entrada. Aunque estos filtros de entrada sean redundantes, también es cierto que son inofensivos y solo demuestran cómo crear filtros de entrada. Los filtros de entrada en reglas de correlación de subcorrelación han de escribirse en Java y devolver un valor booleano.

Para crear las reglas de correlación entre los atributos de los objetos de entrada y salida del paquete:

1. Cree una regla de correlación de movimiento entre los atributos nombre; para ello, arrastre el manejador del atributo nombre del objeto de entrada al atributo nombre del objeto de salida.
2. Cree una regla de correlación de subcorrelación entre el atributo `packagedElement` del objeto de entrada y de salida, y especifique que esta subcorrelación invoca la transformación `Class2ClassTransform` si el elemento es de clase tipo:

- a. Arrastre el manejador del atributo `packagedElement` al objeto de entrada del atributo `packagedElement` en el objeto de salida. Dado que el atributo `packagedElement` es una recopilación, el editor de correlaciones crea una regla de correlación de subcorrelación por omisión.
- b. En el área del editor, efectúe una pulsación con el botón secundario del ratón en el nuevo elemento **Subcorrelación**; a continuación, pulse **Mostrar en Propiedades**.
- c. En la pestaña **Detalles** de la vista Propiedades, seleccione **Class2Class** en la lista **Correlación**.

Al ejecutar la transformación generada, para cada elemento del paquete de entrada, la transformación invoca la transformación `Class2ClassTransform`, pero solo si el elemento es de clase tipo.

3. Opcional: En la regla de subcorrelación creada en el paso 2, cree un filtro de entrada que verifique si el objeto actual es una instancia de una clase:
  - a. En el área del editor, efectúe una pulsación con el botón secundario del ratón en el elemento **Subcorrelación** creado en el paso 2.
  - b. En la pestaña **Filtro de entrada** de la vista Propiedades, marque el recuadro de selección **Filtrar elementos de entrada**.
  - c. Pulse **Incorporado**.
  - d. En el área de texto bajo el campo **Código**, bajo la signature de método, especifique el código siguiente: `return packagedElement_src instanceof org.eclipse.uml2.uml.Class;`

- e. Pulse **Aplicar**.
4. Cree una regla de correlación de subcorrelación entre la característica `packagedElement` del objeto de entrada y de salida, y especifique que el elemento actual de la recopilación `packagedElement` es de clase tipo; la transformación denominada `Package2PackageTransform` transforma la clase en una interfaz del modelo de destino.
  - a. Repita el paso 2a para crear otra subcorrelación entre las características `packagedElement`.
  - b. En el área del editor, efectúe una pulsación con el botón secundario del ratón en el nuevo elemento **Subcorrelación**; a continuación, pulse **Mostrar en Propiedades**.
  - c. En la pestaña **Detalles** de la vista Propiedades, seleccione **Class2Interface** en la lista **Correlación**.
5. Opcional: En la regla de subcorrelación creada en el paso 4, cree un filtro de entrada que verifique si el objeto actual es una instancia de una interfaz:
  - a. En el área del editor, efectúe una pulsación con el botón secundario del ratón en el elemento **Subcorrelación** creado en 4; a continuación, pulse **Mostrar en Propiedades**.
  - b. Repita los pasos 3b, 3c, 3d y 3e para especificar el mismo filtro de entrada.
6. Cree una regla de correlación de subcorrelación que procese los paquetes anidados. Si el elemento actual de la recopilación `packagedElement` es un paquete, la transformación invoca la transformación `Package2PackageTransform`.
  - a. Repita el paso 2a para crear otra subcorrelación entre las características `packagedElement`.
  - b. En el área del editor, efectúe una pulsación con el botón secundario del ratón en el nuevo elemento **Subcorrelación**; a continuación, pulse **Mostrar en Propiedades**.
  - c. En la pestaña **Detalles** de la vista Propiedades, seleccione **Package2Package** en la lista **Correlación**.
7. Opcional: En la regla de subcorrelación creada en el paso 6, cree un filtro de entrada que verifique si el elemento de entrada actual es una instancia de un paquete:
  - a. En el área del editor, efectúe una pulsación con el botón secundario del ratón en el elemento **Subcorrelación** creado en 6; a continuación, pulse **Mostrar en Propiedades**.
  - b. Repita los pasos 3b y 3c para especificar que esta regla tiene un filtro de entrada.
  - c. En el área de texto bajo el campo **Código**, bajo la signature de método, especifique el código siguiente: `return packagedElement_src instanceof org.eclipse.uml2.uml.Package;`
  - d. Pulse **Aplicar**.
8. Pulse **Archivo** → **Guardar**.

La declaración de correlación contiene una regla de correlación de movimiento entre los atributos nombre y diversas reglas de subcorrelación entre los atributos `packagedElement`.

## Lección 7: Crear y perfeccionar una declaración de correlación de modelo a modelo

En esta lección se muestra cómo crear una declaración de correlación de modelo a modelo en el modelo de correlación. Esta declaración de correlación contiene una regla de correlación personalizada que, al ejecutar la transformación generada, crea un modelo de salida de destino cuyo nombre se deriva del modelo de entrada de origen. En esta lección también se muestra cómo cambiar el orden en el que la transformación procesa las declaraciones de correlación en el modelo de correlación.

Para renombrar el modelo de destino que genera la transformación, puede crear una regla de correlación personalizada entre los objetos de entrada y salida, en lugar de crear una regla de correlación entre las características de los objetos de entrada y salida.

Para crear una declaración de correlación de modelo a modelo:

1. Si el modelo de correlación no está abierto en el editor de correlaciones de transformación, en la vista Explorador de paquetes, efectúe una doble pulsación en el archivo `.mapping` de la carpeta `modelo`.

2. En la sección Raíz de correlación del editor de correlaciones de transformación, efectúe una pulsación con el botón secundario del ratón en **Generalize\_Classes** y, a continuación, pulse **Crear correlación**.
3. En el campo **Nombre de correlación** de la ventana Nueva correlación, escriba `Model2Model` y pulse **Aceptar**. La correlación se visualiza en la vista Esquema y se abre bajo la Raíz de correlación en el área del editor.

## Añadir objetos de entrada y salida en la declaración de correlación de modelo a modelo

Después de crear la declaración de correlación de modelo a modelo, debe añadir un objeto de entrada y de salida a este. En esta lección, especifique un modelo como el objeto de entrada y salida.

Para añadir un objeto de entrada y un objeto de salida en la declaración de correlación de modelo a modelo:

1. Pulse el icono **Añadir un objeto de entrada**, que es el icono situado más a la izquierda en la barra de herramientas para la correlación que se edita.
2. En la ventana Añadir entrada, en el panel Elemento, expanda **uml**, pulse **Modelo** y pulse **Aceptar**.
3. Pulse el icono **Añadir un objeto de salida**, que es segundo icono de la izquierda en la barra de herramientas para la correlación que se edita.
4. En la ventana Añadir salida, en el panel Elemento, expanda **uml**, pulse **Modelo** y pulse **Aceptar**.
5. Pulse **Archivo** → **Guardar**.

Puede definir las reglas de correlación entre los objetos de entrada y salida.

## Definir las reglas de correlación personalizada y de subcorrelación

En esta lección se crean las reglas de correlación siguientes:

- Una regla de correlación personalizada que renombra el modelo de destino que genera la transformación.
- Una regla de correlación de subcorrelación invoca la transformación `Package2PackageTransform` para transformar los paquetes del modelo.

Para crear las reglas de correlación en la declaración de correlación:

1. Cree una regla de correlación personalizada que renombre el modelo de destino:
  - a. En el área del editor, pulse el compartimento superior del objeto de entrada del modelo. Se resalta todo el objeto de entrada del modelo.
  - b. Arrastre el manejador del objeto de entrada del modelo al objeto de salida del modelo.
  - c. En el nuevo elemento **Subcorrelación**, pulse la tecla de flecha abajo y pulse **Personalizar**.
  - d. Efectúe una pulsación con el botón secundario del ratón en el elemento **Personalizar**; a continuación, pulse **Mostrar en Propiedades**.
  - e. En la pestaña **Detalles** de la vista Propiedades, pulse **Incorporado**.
  - f. En el área de texto situada bajo el campo **Código**, bajo la signature de método, especifique el código siguiente, que implementa la regla de correlación personalizada al ejecutar la transformación: `Model_tgt.setName(Model_src.getName()+"TgtModel");`
  - g. Pulse **Aplicar**.

**Nota:** Para renombrar el modelo de destino, también puede crear una regla de correlación personalizada entre el atributo name del objeto de entrada y del objeto de salida, y especificar el mismo código en la pestaña **Detalles**, en lugar de crear una regla de correlación entre los objetos de entrada y salida.

2. Cree una regla de correlación de subcorrelación que invoque la correlación Package2Package creada en “Lección 6: Crear y perfeccionar una declaración de correlación de paquete a paquete” en la página 17:
  - a. Arrastre el manejador de la característica packagedElement al atributo packagedElement en el objeto de salida del modelo.
  - b. Efectúe una pulsación con el botón secundario del ratón en el nuevo elemento **Subcorrelación**; a continuación, pulse **Mostrar en Propiedades**.
  - c. En la pestaña **Detalles** de la vista Propiedades, seleccione **Package2Package** en la lista **Correlación**.
3. Pulse **Archivo** → **Guardar**.

Ha creado todas las declaraciones de correlación y reglas de correlación necesarias para esta guía de aprendizaje. Ahora puede especificar el orden de proceso de las declaraciones de correlación.

### Cambiar el orden de proceso de las declaraciones de correlación

Puede cambiar el orden de proceso de las declaraciones de correlación en un modelo de correlación. En la vista Esquema se listan las declaraciones de correlación en el orden en que se procesan al ejecutar el código de transformación generado. Al cambiar el orden de proceso, puede especificar instrucciones de proceso para los objetos de entrada que una declaración de correlación menos específica podría procesar y consumir.

Para esta guía de aprendizaje, especifique que la transformación generada procesa las declaraciones de correlación en el orden siguiente:

- Model2Model
- Package2Package
- Class2Class
- Class2Interface
- Operation2Operation
- Parameter2Parameter
- Primitive2Primitive

**Nota:** En esta guía de aprendizaje, el orden de las declaraciones de correlación Model2Model y Package2Package es lo más importante. Al ejecutar la transformación generada, dado que un modelo es un paquete, la transformación Package2Package acepta un modelo o un paquete. Si la transformación Package2Package es la primera transformación que se lista, esta procesa y consume el objeto de entrada del modelo. En este escenario, la transformación Model2Model no procesa el objeto de entrada del modelo; por tanto, la regla personalizada creada en la transformación Model2Model no se ejecuta.

Para cambiar el orden de proceso de las declaraciones de correlación:

1. Si la vista Esquema no es visible, pulse **Ventana** → **Mostrar vista** → **Esquema**.
2. En la vista Esquema, efectúe una pulsación con el botón secundario del ratón en la declaración de correlación Model2Model; pulse **Orden de ejecución** y pulse **Subir**. Repita este paso hasta que la declaración de correlación Model2Model esté al principio de la lista.
3. Repita el paso 2 para cada declaración de correlación hasta que todas estén en el mismo orden que en la lista de más arriba.
4. Pulse **Archivo** → **Guardar**.

Ahora puede generar el código fuente de transformación.

---

## Módulo 2: Generar código de transformación y ejecutar la transformación

En este módulo, se genera el código para la transformación de modelo a modelo y se prueba la transformación al ejecutarla en una instancia aparte del entorno de trabajo de Eclipse.

### Objetivos del curso

Las lecciones de este módulo muestran cómo realizar las tareas siguientes:

- Genere y compile el código fuente de transformación.
- Configurar un entorno de trabajo de tiempo de ejecución.
- Crear una configuración de transformación.
- Ejecutar el código fuente de transformación en un entorno de trabajo de tiempo de ejecución.

### Tiempo necesario

Este módulo tiene una duración aproximada de 20 minutos.

### Requisitos previos

Debe saber cómo realizar las tareas siguientes:

- Crear modelos UML en proyectos.
- Efectuar el modelado básico UML.

## Lección 1: Generar y compilar código fuente de transformación

En esta lección se muestra cómo generar y compilar el código fuente de transformación.

Para generar el código fuente de transformación, en el proyecto Generalize Classes, examine el contenido de la carpeta src. Estos paquetes y archivos se han generado al crear el proyecto; los otros archivos se han creado al editar el archivo Generalize\_Classes.mapping.

Puede añadir reglas de correlación y generar implementaciones incrementalmente para las reglas de correlación en una declaración de correlación. No debe definir todas las reglas de correlación para poder generar el código fuente de transformación. Por ejemplo, después de completar esta guía de aprendizaje, podría añadir reglas de correlación que creasen una relación de realización de la clase de implementación con la interfaz.

Al generar el código fuente de transformación, para cada declaración de correlación del modelo de correlación, la infraestructura de creación de transformaciones genera un archivo de origen Java con el nombre *n*Transform.java, donde *n* representa el nombre de una declaración de correlación. Colectivamente, estos archivos Java componen el código de transformación. Además de generar el código de implementación para la transformación, la infraestructura de creación de transformaciones también genera código para registrar la transformación en el servicio de transformación.

Para generar y compilar el código fuente de transformación:

1. Si todavía no está abierta, abra la perspectiva Desarrollo de plug-ins: Pulse **Ventana** → **Abrir perspectiva** → **Otras** → **Desarrollo de plug-ins** y, a continuación, pulse **Aceptar**.
2. Si el archivo Generalize\_Classes.mapping no está abierto en el editor de correlaciones, en la vista Explorador de paquetes, efectúe una doble pulsación en el archivo.
3. En el área del editor de correlaciones, efectúe una pulsación con el botón secundario del ratón en la sección Raíz de correlación y, a continuación, pulse **Generar código fuente de transformación**.

**Consejo:** Para generar el código fuente de transformación, lleve a cabo estos pasos: en la vista Explorador de paquetes, efectúe una pulsación con el botón secundario del ratón en el archivo .mapping y, a continuación, pulse **Transformar** → **Generar código fuente de transformación**. En la vista Explorador de paquetes, examine los paquetes y archivos de la carpeta src. Las transformaciones generadas se muestran en la carpeta src/generalize\_classes.transforms.

4. Para compilar el código fuente generado, en la vista Explorador de paquetes, pulse el proyecto **Generalize Classes** y, a continuación, pulse **Proyecto** → **Construir proyecto**. Por omisión, los proyectos de Eclipse se construyen automáticamente al guardar cambios en un proyecto. Si no ha cambiado las preferencias de construcción para el espacio de trabajo o proyecto, no es necesario que efectúe este paso.

**Consejo:** Para cambiar las preferencias de construcción de proyectos, pulse **Ventana** → **Preferencias**, expanda **General** y pulse **Espacio de trabajo**. Después de cambiar las preferencias, pulse **Aceptar**.

5. Examine el contenido de la carpeta src para ver si hay errores.
6. Si se produce un error de compilación en el código de transformación Class2Interface, es posible que necesite importar el paquete VisibilityKind:
  - a. En la vista Explorador de paquetes, en el archivo src/generalize\_classes.transforms/Class2InterfaceTransform.java, efectúe una doble pulsación en el método que tiene el adorno de error.
  - b. En el margen izquierdo del área del editor, efectúe una pulsación con el botón secundario del ratón en el adorno de error y, a continuación, pulse **Arreglo rápido** y efectúe una doble pulsación en **Importar "VisibilityKind" (org.eclipse.uml2.uml)**.
  - c. Pulse **Archivo** → **Guardar**.
7. En la vista Explorador de paquetes, pulse el proyecto **Generalize Classes** y, a continuación, pulse **Proyecto** → **Construir proyecto**.

Revise los archivos en la carpeta src/generalize\_classes.transforms del proyecto. Se genera una transformación Java para cada declaración de correlación en el modelo de correlación. En el archivo MainTransform.java, el método MainTransform añade una instancia de cada transformación generada en el orden especificado en la vista Esquema.

Ahora puede configurar un entorno de trabajo de tiempo de ejecución.

## Lección 2: Configurar un entorno de trabajo de tiempo de ejecución

En esta lección se muestra cómo configurar un entorno de trabajo de tiempo de ejecución y cómo crear modelos UML de origen y de destino en este entorno de trabajo.

Puede crear e invocar un entorno de trabajo de tiempo de ejecución para probar y depurar transformaciones, lo que significa que no es necesario empaquetar el plug-in de transformación antes de efectuar pruebas.

Después de crear y abrir el entorno de trabajo de tiempo de ejecución, debe crear modelos de origen y de destino para probar el código de transformación. La transformación transforma los elementos creados en el modelo de origen y genera la salida en el modelo de destino.

Para configurar un entorno de trabajo de tiempo de ejecución:

1. Abra la perspectiva Desarrollo de plug-ins: Pulse **Ventana** → **Abrir perspectiva** → **Desarrollo de plug-ins**.
2. Pulse **Ejecutar** → **Abrir diálogo Ejecutar**.
3. En la ventana Ejecutar, en el panel izquierdo, pulse **Aplicación Eclipse** y pulse el icono **Nueva configuración de lanzamiento**, que es el primer icono de la izquierda en la barra de herramientas.
4. Pulse la pestaña **Configuración**.

5. Pulse **Utilizar un archivo config.ini existente como plantilla** y acepte el valor predeterminado en el campo **Ubicación**. Estos valores especifican que la instancia de tiempo de ejecución es una instancia del producto que se ejecuta, en lugar de una instancia predeterminada de Eclipse. Una instancia de tiempo de ejecución predeterminada de Eclipse no proporciona suficiente funcionalidad para probar una transformación.
6. Pulse **Aplicar**.
7. Pulse **Ejecutar**.

**Nota:** Si los proyectos del espacio de trabajo contienen errores, se visualiza un recuadro de diálogo en el que se listan los proyectos. Para continuar abriendo el entorno de trabajo de tiempo de ejecución, pulse **Continuar**.

Se abre el nuevo entorno de trabajo de tiempo de ejecución.

**Nota:** Según los recursos disponibles en el sistema, es posible que el entorno de trabajo de tiempo de ejecución tarde algunos minutos en abrirse.

## Crear un proyecto de prueba en el entorno de trabajo de tiempo de ejecución

Después de configurar y abrir el entorno de trabajo de tiempo de ejecución, debe crear un proyecto que contenga los elementos siguientes:

- Un modelo UML de origen para la transformación que se va a transformar; este modelo contiene un paquete con una clase.
- Un modelo de destino vacío donde la transformación genera la salida.

Para crear el proyecto que contiene los modelos de origen y de destino:

1. En el entorno de trabajo de tiempo de ejecución, abra la perspectiva Modelado: Pulse **Ventana** → **Abrir perspectiva** → **Modelado**.
2. Cree un proyecto de modelado UML denominado TransformationTest y un modelo UML denominado SourceModel:
  - a. Pulse **Archivo** → **Nuevo** → **Proyecto**, expanda **Modelado**, pulse **Proyecto UML** y, a continuación, pulse **Siguiente**.
  - b. En el campo **Nombre de proyecto** de la página Crear proyecto de modelo, especifique TransformationTest. Acepte los valores predeterminados para el resto de campos y pulse **Siguiente**.
  - c. En la página Crear modelo, si todavía no está seleccionado, pulse **General** en el panel Categorías.
  - d. En el panel Plantillas, pulse **Modelo en blanco**.
  - e. En el campo **Nombre de archivo**, especifique SourceModel.
  - f. Pulse **Finalizar**.
  - g. Si se solicita que cambie a la perspectiva Modelado, pulse **Sí**.
3. En el proyecto TransformationTest, cree un modelo UML llamado TargetModel:
  - a. En la vista Explorador de proyectos, pulse el botón derecho del ratón sobre el proyecto TransformationTest y, a continuación, pulse **Nuevo** → **Modelo UML**.
  - b. En el asistente Modelo UML, en la página Crear modelo, acepte los valores predeterminados y pulse **Siguiente**.
  - c. En la segunda página Crear modelo, si todavía no está seleccionado, pulse **General** en el panel Categorías.
  - d. En el panel Plantillas, pulse **Modelo en blanco**.
  - e. En el campo **Nombre de archivo**, especifique TargetModel.
  - f. Pulse **Finalizar**.

El proyecto TransformationTarget ahora contiene el modelo de origen que transforma la transformación generada y el modelo de destino donde la transformación genera la salida.

4. En el modelo SourceModel, cree un paquete llamado BusinessClasses que contenga una clase denominada Employee que tenga tres operaciones privadas y una operación pública:
  - a. En la vista Explorador de proyectos, efectúe una pulsación con el botón secundario del ratón en el modelo **SourceModel** y, a continuación, pulse **Añadir UML → Paquete**.
  - b. Denomine al paquete BusinessClasses.
  - c. Efectúe una pulsación con el botón secundario del ratón en el paquete **BusinessClasses** y, a continuación, pulse **Añadir UML → Clase**.
  - d. Denomine a la clase Employee.
  - e. Efectúe una pulsación con el botón secundario del ratón en la clase **Employee** y, a continuación, pulse **Añadir UML → Operación**.
  - f. Denomine a la operación readEmail.
  - g. En la pestaña **General** de la vista Propiedades, en el área **Visibilidad**, pulse **Privada**.
  - h. Repita el paso 4e y denomine a la operación answerPhone.
  - i. Repita el paso 4g para especificar que la operación answerPhone es privada.
  - j. Repita el paso 4e y denomine a la operación performWork.
  - k. Repita el paso 4g para especificar que la operación performWork es privada.
  - l. Repita el paso 4e y denomine a la operación reportToManager(name:String).
5. Pulse **Archivo → Guardar**.

Ahora puede crear una configuración de transformación, que se utiliza para ejecutar la transformación.

### Lección 3: Crear una configuración de transformación

En esta lección se muestra cómo crear una configuración de transformación. Una configuración de transformación especifica la información que necesita una transformación para generar la salida esperada.

Puede aplicar una configuración de transformación para transformar el contenido especificado en un contenido diferente en el modelo de destino. Si el origen especificado es un modelo, la transformación interpreta los elementos del modelo de origen y genera una salida en el destino especificado. En la mayor parte de los casos, el modelo de origen no sufre cambios. La transformación genera un nuevo conjunto de archivos o modifica un conjunto existente de archivos.

Para simplificar el trabajo con el archivo de configuración de transformación, debe guardar el archivo de configuración en el proyecto que contiene el origen de la transformación. En esta lección, se guarda el archivo de configuración de transformación en el entorno de trabajo de tiempo de ejecución, en el proyecto TransformationTest. Los archivos de configuración de transformación tienen la extensión .tc.

Para crear una configuración de transformación:

1. En el entorno de trabajo de tiempo de ejecución, pulse **Archivo → Nuevo → Configuración de transformación**. Si **Configuración de transformación** no es un elemento del menú, pulse **Archivo → Nuevo → Otros → Configuración de transformación**.
2. En la página Nombre y transformación del asistente Nueva configuración de transformación, especifique los valores siguientes:
  - a. En el campo **Nombre**, escriba FirstConfiguration.
  - b. En la lista **Transformación directa**, expanda **Generalize\_Classes** y seleccione **Transformación Generalize\_Classes**.
  - c. En el campo **Destino de archivo de configuración**, si todavía no está especificado, escriba /TransformationTest.
  - d. Pulse **Siguiente**.
3. En la página Origen y destino, especifique el modelo de origen que transforma la transformación y el modelo de destino donde la transformación genera la salida:

- a. En el panel Origen seleccionado, pulse **TransformationTest** → **Modelos** → **SourceModel**
- b. En el panel Destino seleccionado, pulse **TransformationTest** → **Modelos** → **TargetModel**
4. Pulse **Siguiente** y acepte los valores predeterminados del resto de páginas del asistente.
5. Pulse **Finalizar**.

Se crea un archivo de configuración de transformación en el entorno de trabajo de tiempo de ejecución, en el proyecto TransformationTest y se visualiza el archivo de configuración de transformación en el editor de configuraciones de transformación. Ahora puede ejecutar la transformación.

## Lección 4: Ejecutar la transformación

En esta lección se muestra cómo ejecutar la transformación en el entorno de trabajo de tiempo de ejecución. Al aplicar una configuración de transformación, se crea una instancia de la transformación y esta se ejecuta con las propiedades especificadas en la configuración de transformación.

Al ejecutar la transformación, se crea un modelo temporal y se compara este modelo con el modelo de destino especificado en la configuración de transformación. Las diferencias entre los dos modelos se visualizan en la ventana Fusionar.

Para obtener más información sobre cómo ejecutar transformaciones y especificar estrategias de fusión, consulte los temas relacionados más abajo.

Para aplicar la configuración de transformación que invoca la transformación Generalize\_Classes:

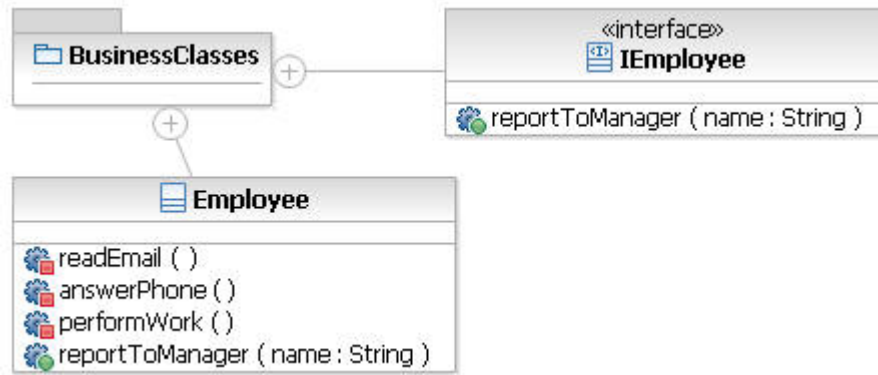
1. Si todavía no está abierto, en la vista Explorador de proyectos del entorno de trabajo de tiempo de ejecución, en el proyecto TransformationTest, efectúe una doble pulsación en **FirstConfiguration.tc** y, a continuación, pulse **Ejecutar** en la página principal del editor de configuraciones de transformación.

**Consejo:** En el proyectos TransformationTest, también puede efectuar una pulsación con el botón secundario del ratón en **FirstConfiguration.tc** y, a continuación, pulsar **Transformar** → **Transformación Generalize Classes**.

2. Mientras se ejecuta la transformación, basada en las opciones de fusión predeterminadas especificadas por el proveedor de transformaciones, es posible que se le solicite que acepte los cambios en los archivos del modelo de destino. Pulse **Aceptar**.
3. Como respuesta a los mensajes que se visualizan, pulse **Aceptar**.
4. En la ventana Fusionar, revise las actualizaciones que se proponen para el modelo de destino, marque o quite la marca de los recuadros de selección para aceptar o rechazar los cambios propuestos y pulse **Aceptar**.
5. Como respuesta a los mensajes que se visualizan, pulse **Aceptar**.

**Consejo:** Para volver a ejecutar esta transformación, pulse **Modelado** → **Transformar** → **Ejecutar última**. La transformación utiliza los mismos elementos de origen empleados en la última ejecución de la transformación.

Ahora puede explorar la salida de la transformación en el modelo TargetModel. En la imagen de más abajo se muestra la representación visual de los elementos en el modelo de salida SourceModelTgtModel generado. Este modelo contiene un paquete denominado BusinessClasses, que contiene una clase llamada Employee y una interfaz llamada IEmployee.



Información relacionada:

Añadir soporte de fusión para modelos generados por proyectos de correlación de transformación de modelo a modelo

Ejecutar y volver a ejecutar transformaciones

---

## Resumen: Crear, configurar y ejecutar una transformación de modelo a modelo

En esta guía de aprendizaje se muestra cómo crear, configurar y ejecutar una transformación simple de correlación de modelo a modelo.

### Lecciones aprendidas

En esta guía de aprendizaje se ha explicado los conceptos de las transformaciones de correlación de modelo a modelo y se ha mostrado cómo realizar las tareas siguientes:

- Crear un proyecto de creación de transformaciones de modelo a modelo que contenga una infraestructura de transformación y un modelo de correlación.
- Crear declaraciones de correlación en el modelo de correlación.
- Perfeccione las declaraciones de correlación efectuando las tareas siguientes:
  - Especificar un objeto de entrada de origen y un objeto de salida de destino para cada declaración de correlación.
  - Crear reglas de correlación definiendo las relaciones entre los atributos de los objetos de entrada y salida en una declaración de correlación.
- Generar y compilar el código fuente de transformación.
- Configurar un entorno de trabajo de tiempo de ejecución para probar la transformación de modelo a modelo.
- En un entorno de trabajo de tiempo de ejecución, cree y aplique una configuración de transformación que ejecute la transformación de modelo a modelo.

### Recursos adicionales

Para obtener información sobre los temas de esta guía de aprendizaje, tenga en cuenta las fuentes siguientes:

- Ayuda en línea para la creación de transformaciones y extensiones de transformación.