Procesadores de Lenguajes 2

Construcción de editores de modelos con EMF

Curso 2013-2014

Iván Ruiz Rube
Departamento de Ingeniería Informática
Escuela Superior de Ingeniería
Universidad de Cádiz

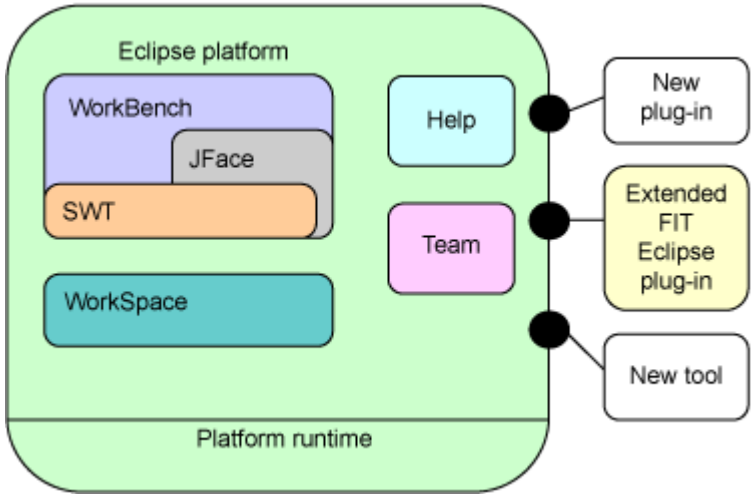
Contenidos

- Introducción
- Generar clases de soporte
- Despliegue de editores como plug-ins
- Despliegue de editores como productos

CONSTRUCCIÓN DE EDITORES DE MODELOS CON EMF

INTRODUCCIÓN

Desarrollo sobre Eclipse



- Eclipse se compone de una base de código (kernel) y un conjunto de extensiones adicionales (plug-ins).
- Estas extensiones de la GUI se desarrollan con JFace (un framework MVC sobre SWT)
- Las herramientas de Eclipse Modeling Project, se encargarán de hacerlo por nosotros.

Eclipse Modeling Framework

- Ya vimos como EMF permite diseñar metamodelos utilizando el lenguaje Ecore.
- Ahora, vamos a generar sencillos editores de modelos reflexivos basados en una estructura de árbol.
- El desarrollo del editor se basará en una sucesiva transformación de modelos.

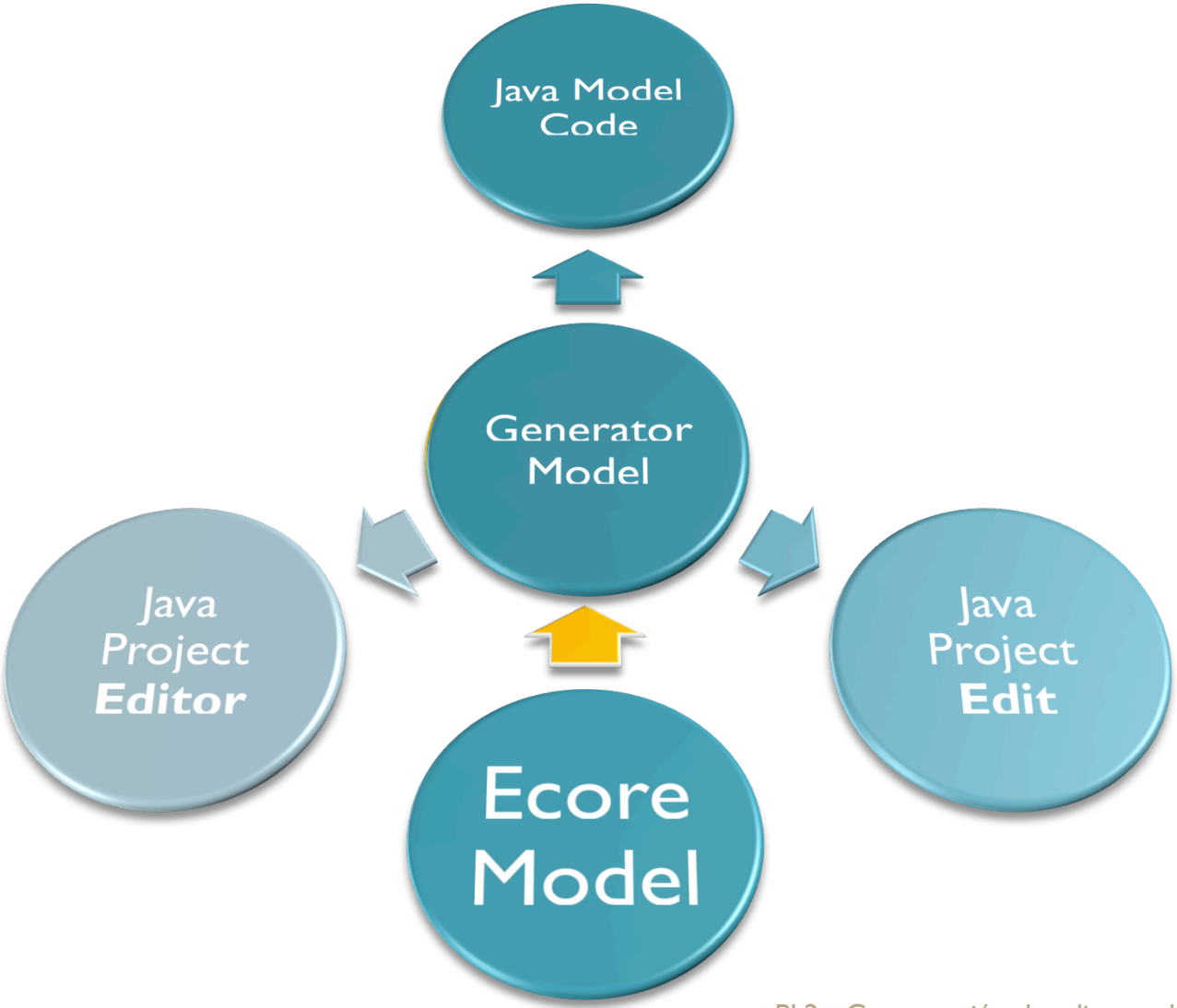
Pasos para crear un editor de modelos

1. Elaborar y validar un metamodelo con Ecore
2. Generar clases Java de soporte al metamodelo:
   - factorías, interfaces, listeners, etc.
3. Desplegar editor
   - Plug-in de Eclipse: Eclipse PDE
   - Aplicación independiente: Eclipse RCP
   - Aplicación web: Eclipse RAP

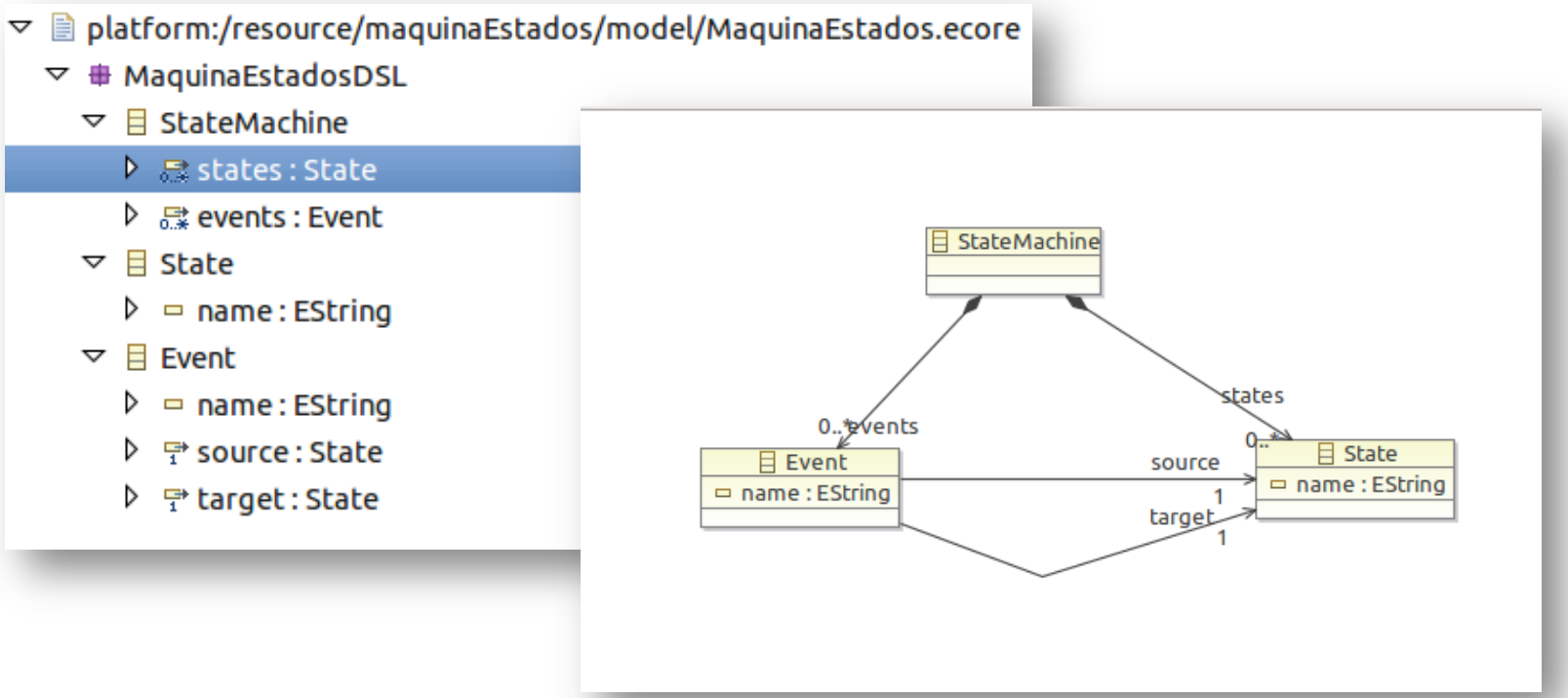
CONSTRUCCIÓN DE EDITORES DE MODELOS CON EMF

GENERAR CLASES DE SOPORTE

Transformación de modelos

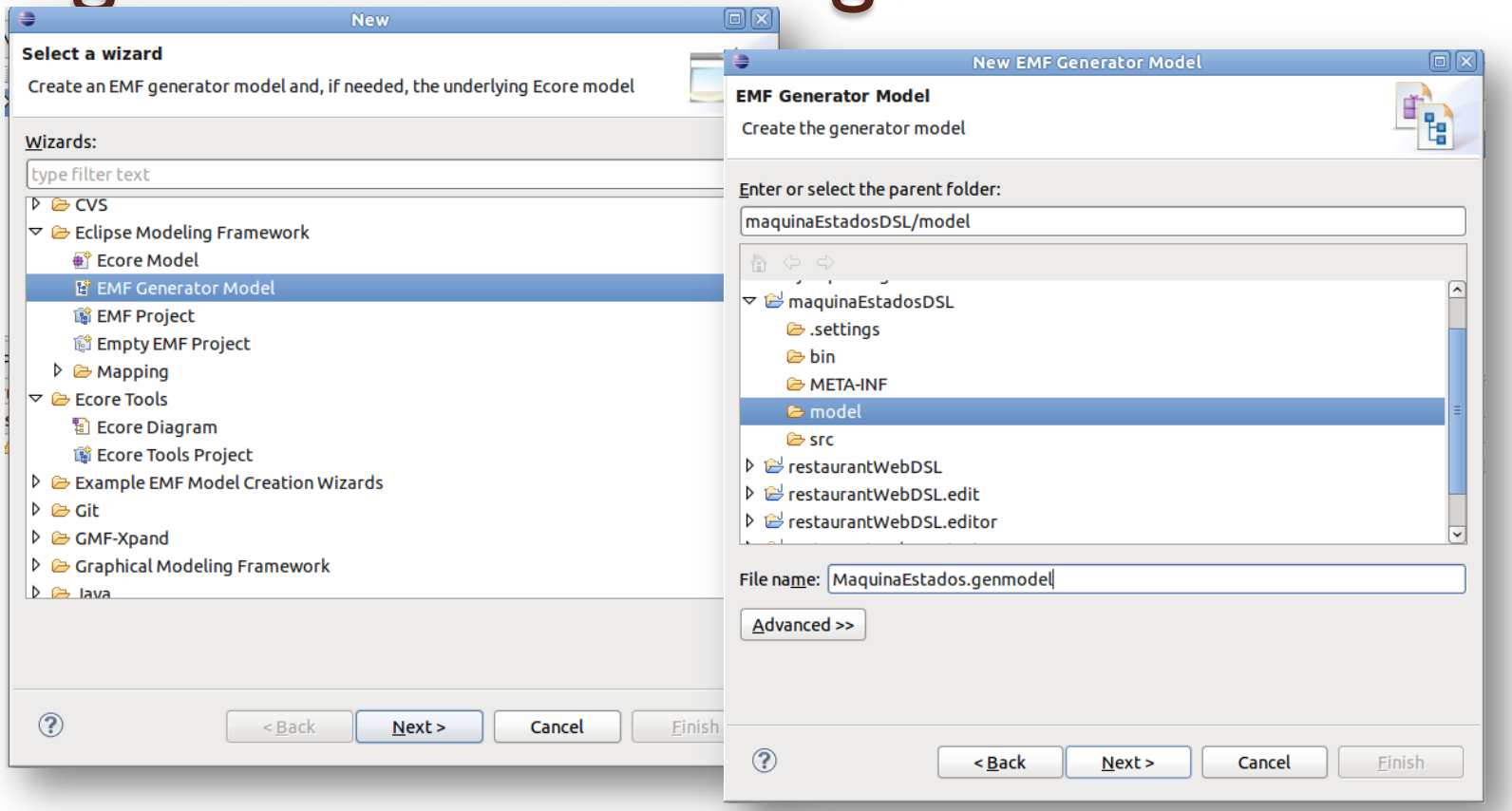


Desarrollo del metamodelo



Diseñamos nuestro metamodelo Ecore con el editor de metamodelos de EMF

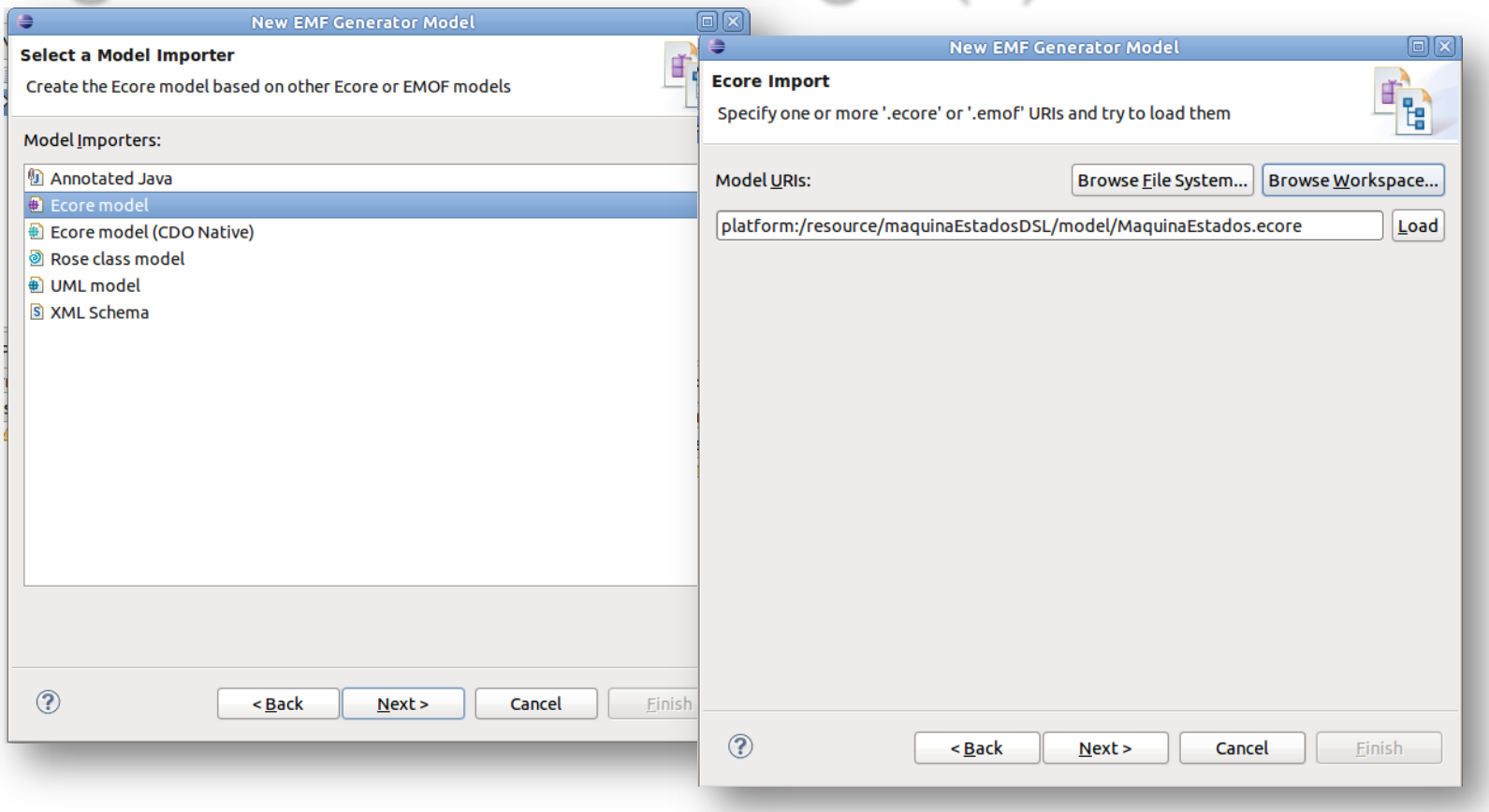
Creación de un modelo de generación de código



`File → New → EMF Generator Model`

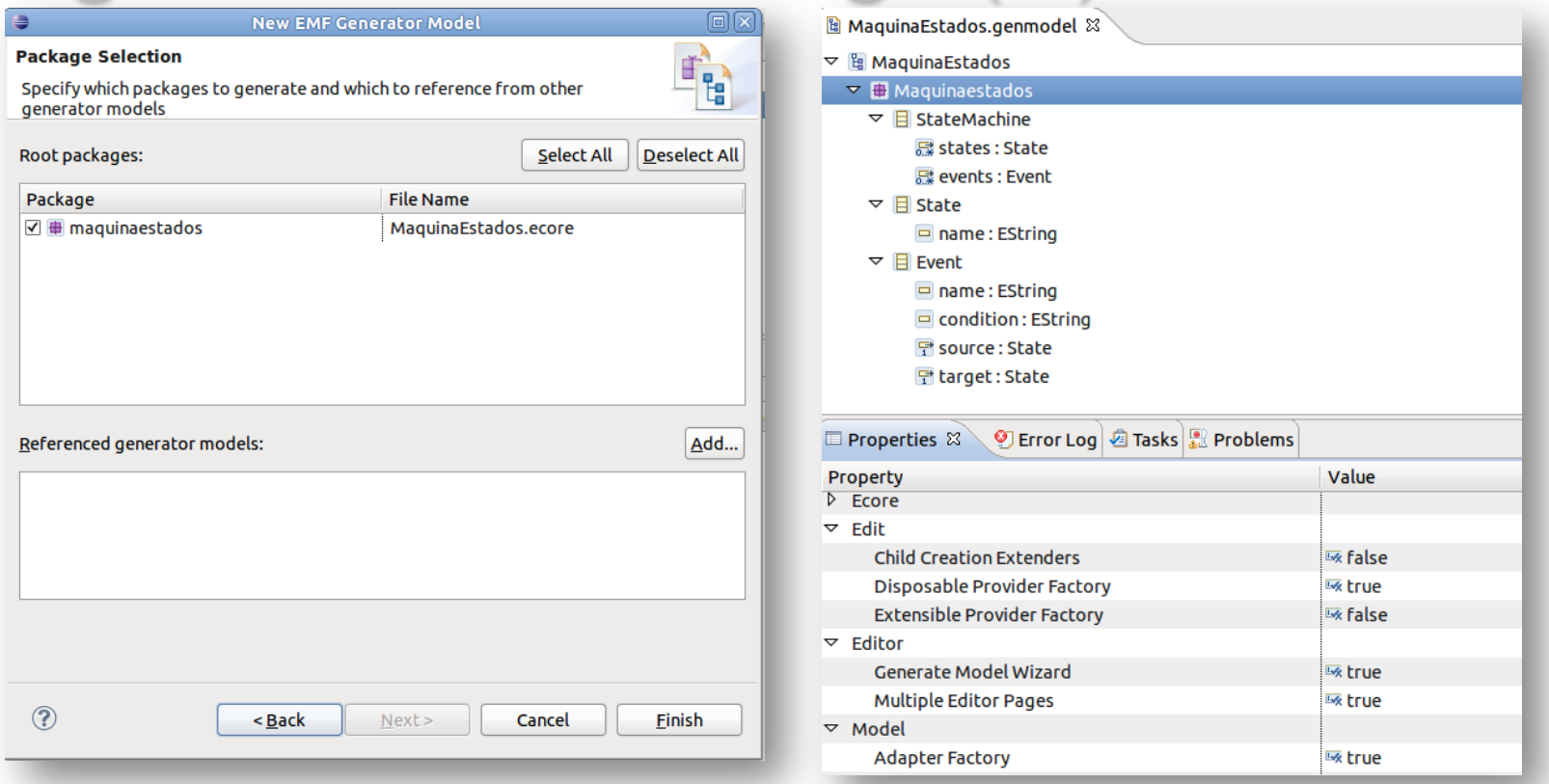
El modelo de generación de código lo llamaremos igual que el modelo Ecore, pero con extensión ".genmodel"

Creación de un modelo de generación de código (II)



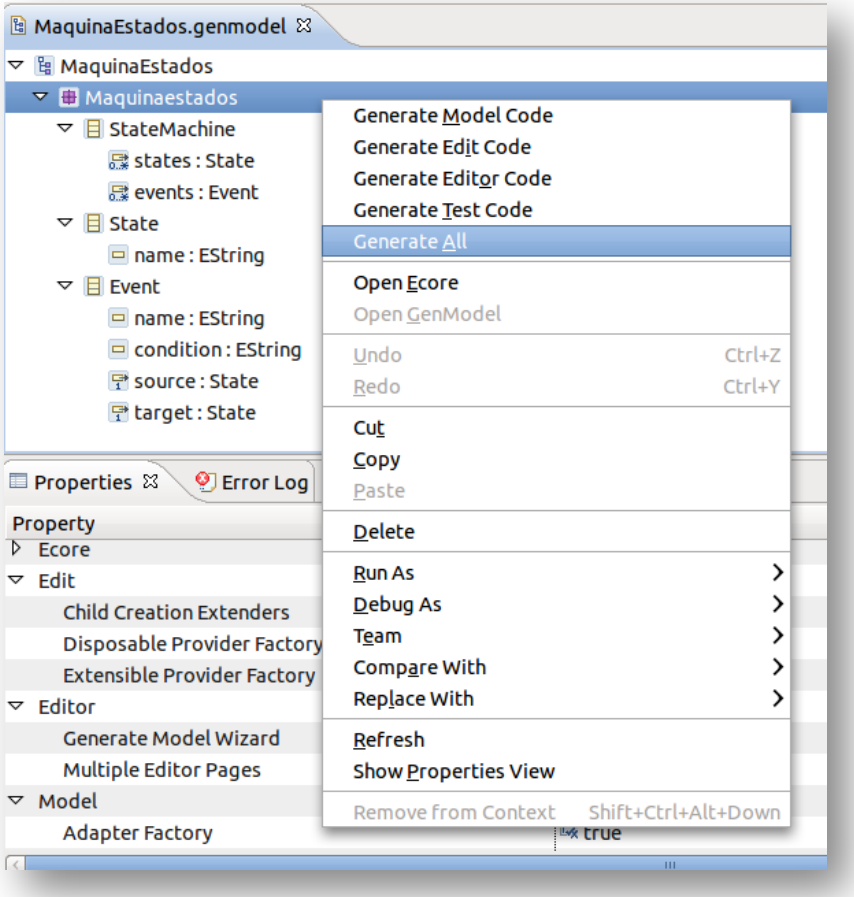
El modelo de generación de código lo crearemos a partir del modelo Ecore diseñado previamente.

Creación de un modelo de generación de código (III)



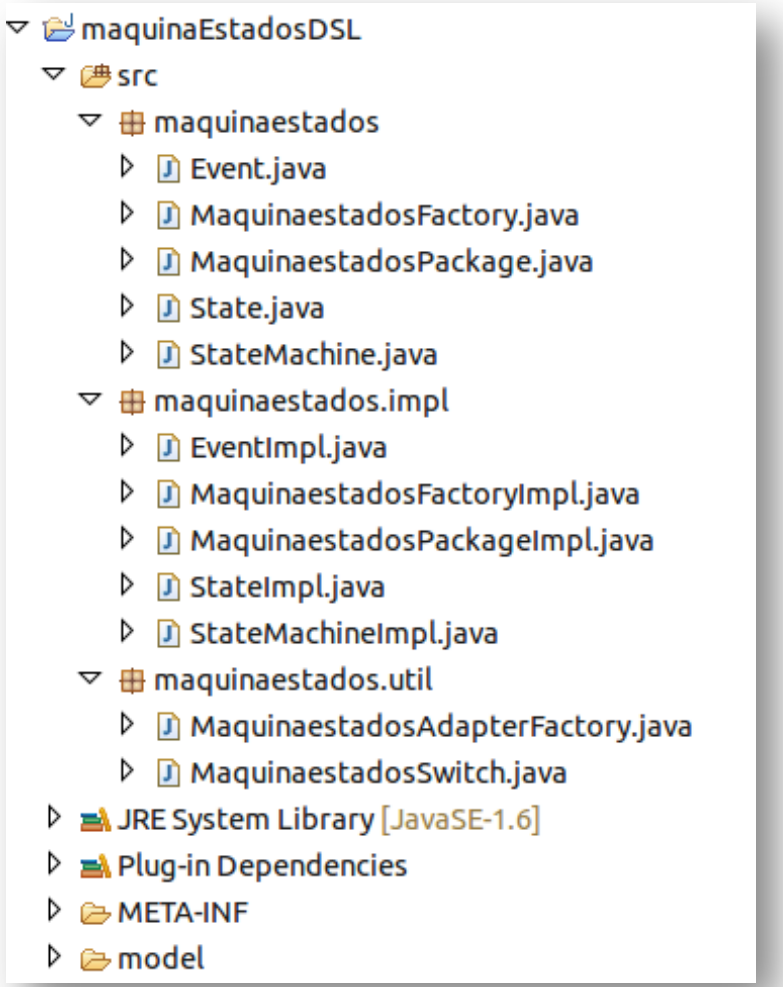
Una vez seleccionado el EPackage del modelo Ecore, automáticamente (via M2M) se construye un nuevo modelo similar al anterior, aunque con información adicional.

Generación del tree-editor



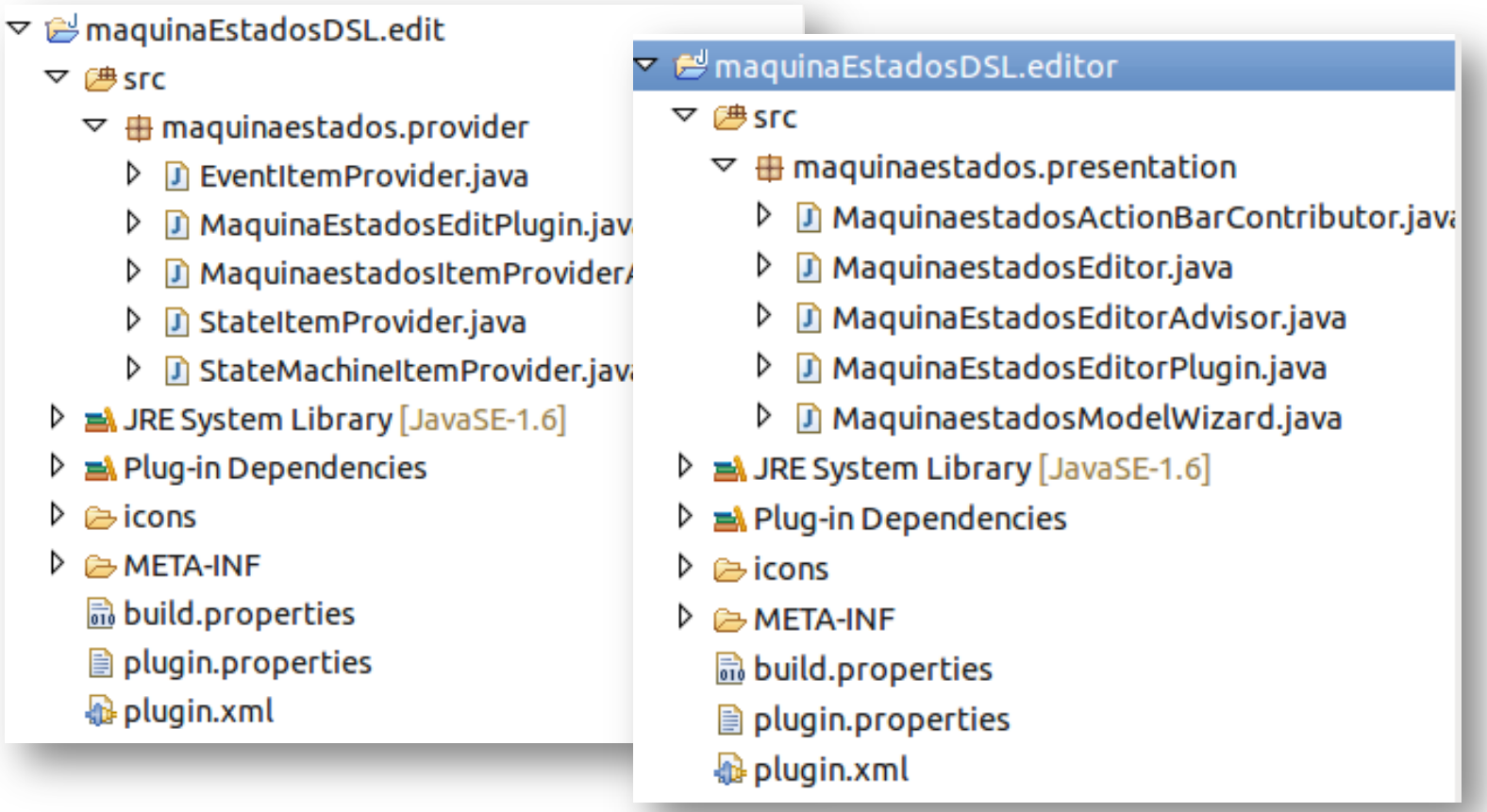
Partiendo de nuestro modelo de generación de código, generaremos el código fuente necesario para implementar nuestro editor de modelos.

Java Model Code



- Implementación en Java de los elementos del metamodelo.
- Se generan interfaces, clases de implementación, factorías y otras clases de soporte necesarias para construir editores.
- También se genera un proyecto *Tests*, con casos de prueba en Junit.

Edit / Editor

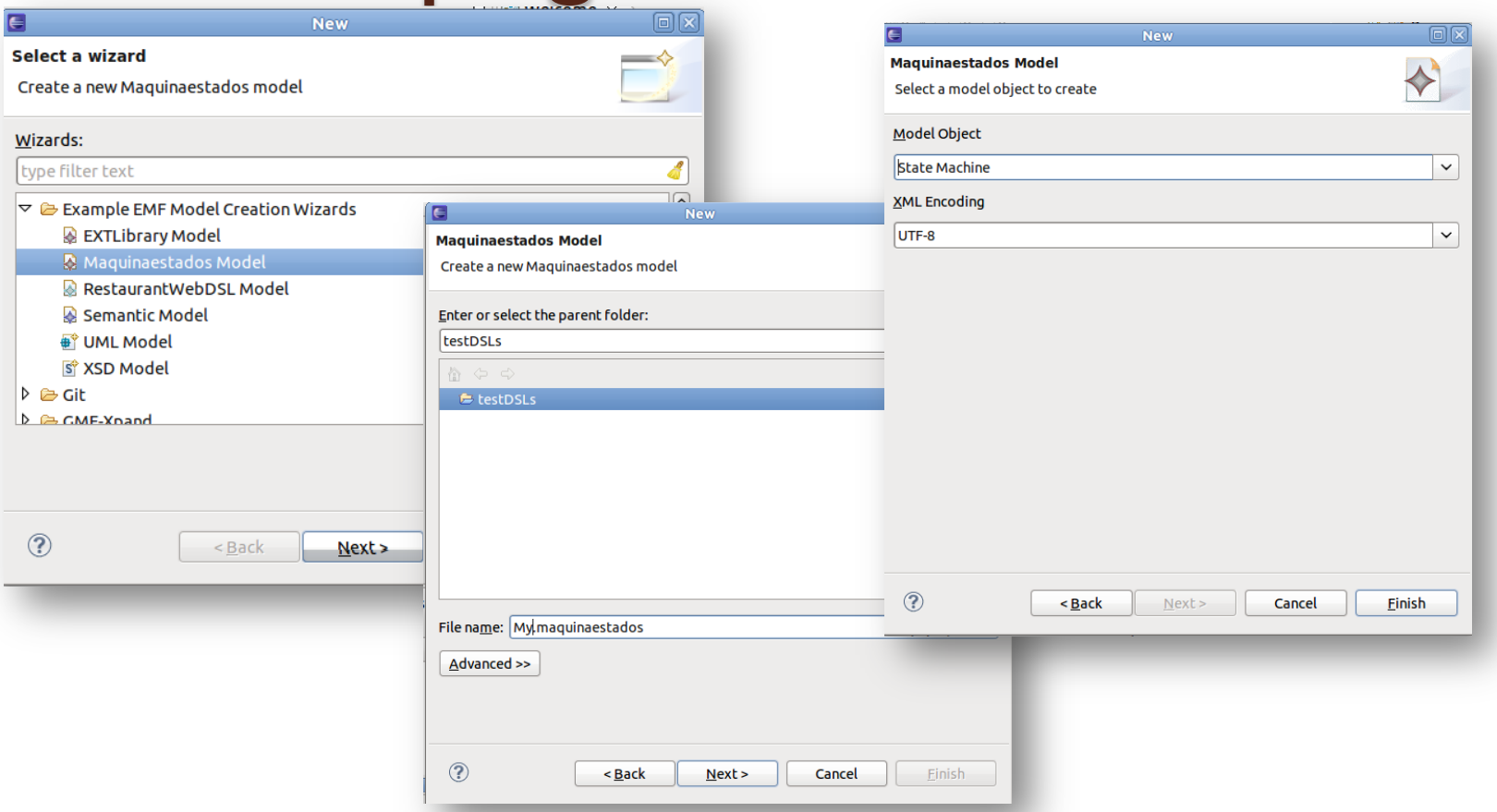


El proyecto *Edit* incluye las clases necesarias para la visualización y edición de los modelos. *Editor* proporciona los elementos requeridos para la interfaz de usuario de nuestro editor de modelos.

CONSTRUCCIÓN DE EDITORES DE MODELOS CON EMF

DESPLIEGUE DE EDITORES COMO PLUG-INS

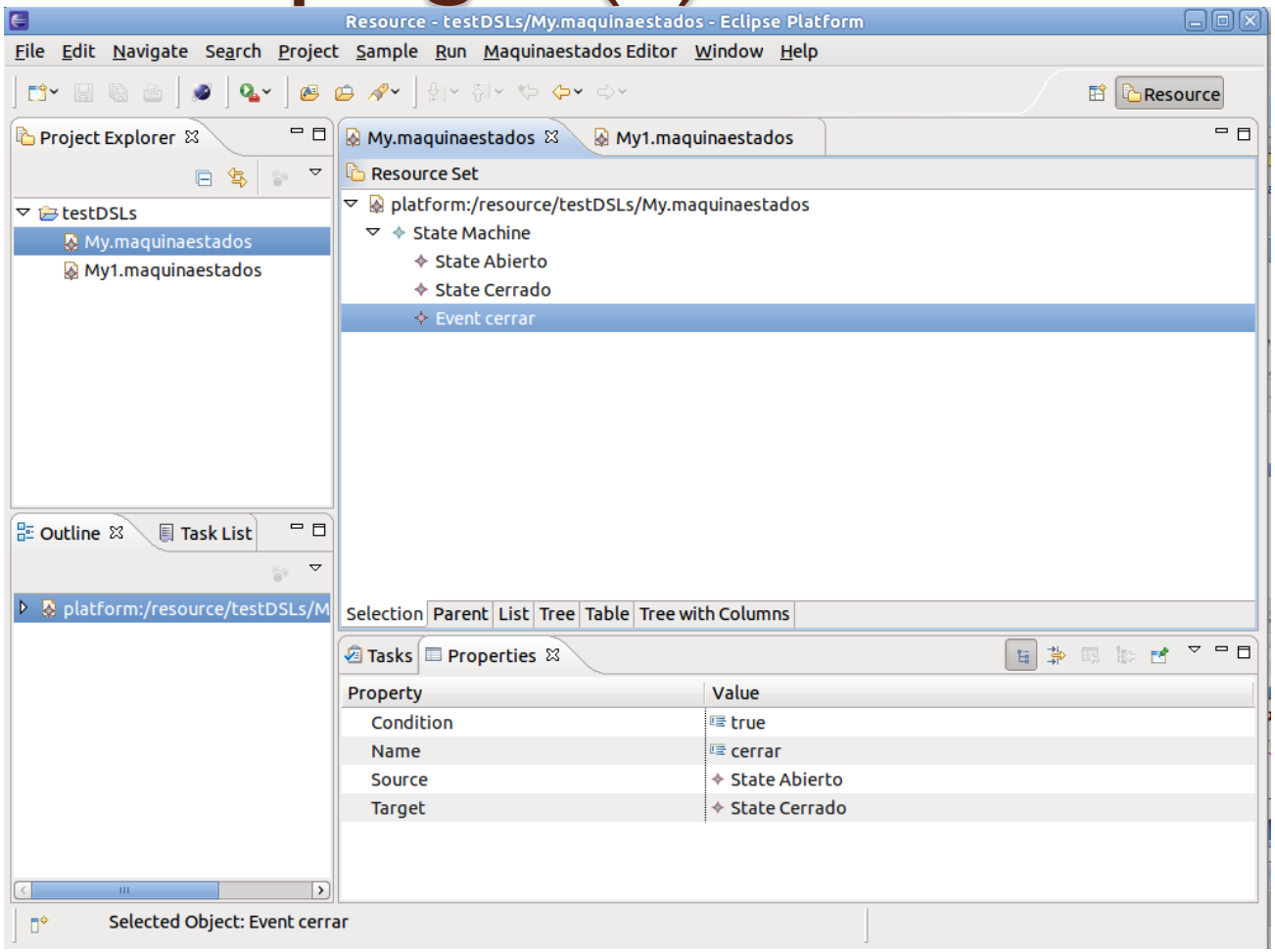
Test del plugin



`[Project Editor] Run as → Eclipse Application`

Al ejecutar el proyecto Editor, se abre una nueva instancia de Eclipse incluyendo el plugin tree-editor de modelos.

Test del plugin (II)



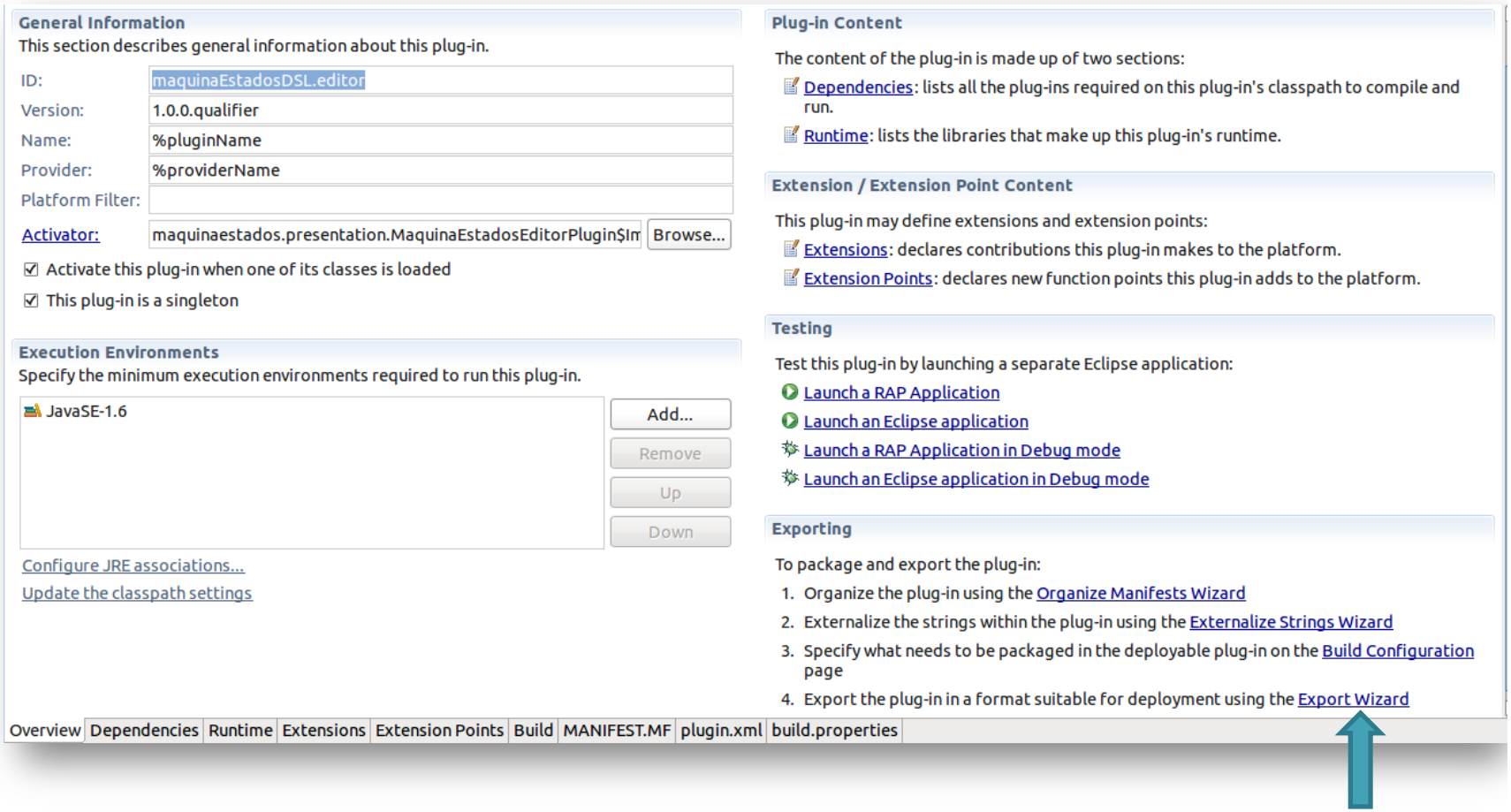
Nuestro nuevo editor de modelos se integra perfectamente en el entorno Eclipse.

Personalizar mensajes del plugin

```
pluginName = MaquinaEstados Editor
providerName = www.example.org
_UI_MaquinaestadosEditor_menu = &Maquinaestados Editor
_UI_CreateChild_menu_item = &New Child
_UI_CreateSibling_menu_item = N&ew Sibling
_UI_ShowPropertiesView_menu_item = Show &Properties View
_UI_OpenEditorError_label = Open Editor
_UI_Wizard_category = Example EMF Model Creation Wizards
_UI_CreateModelError_message = Problems encountered in file "{0}"
_UI_MaquinaestadosModelWizard_label = Maquinaestados Model
_UI_MaquinaestadosModelWizard_description = Create a new model
_UI_MaquinaestadosEditor_label = Maquinaestados Model Editor
```

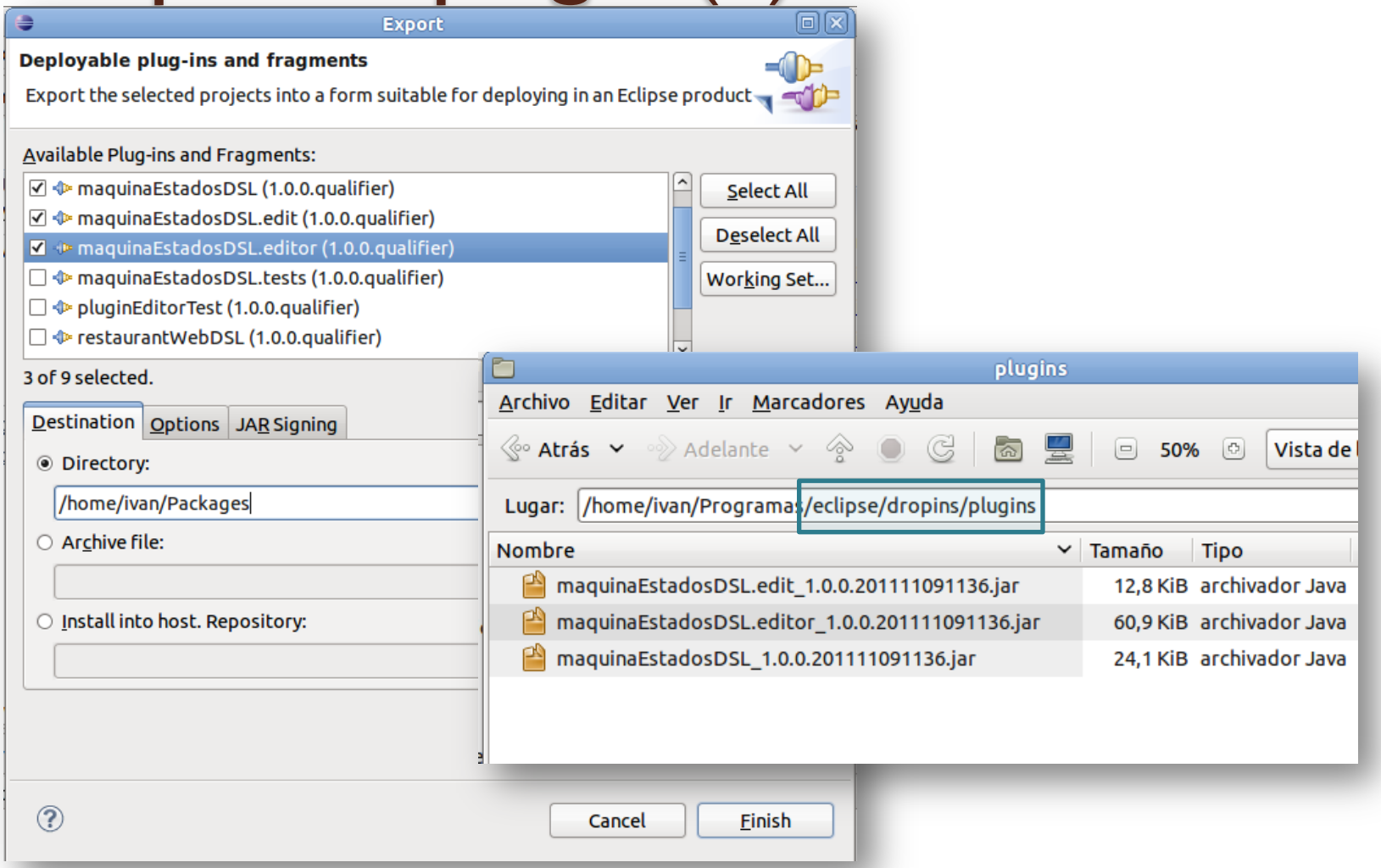
[plugin.properties]

Exportar plugin



[plugin.xml]

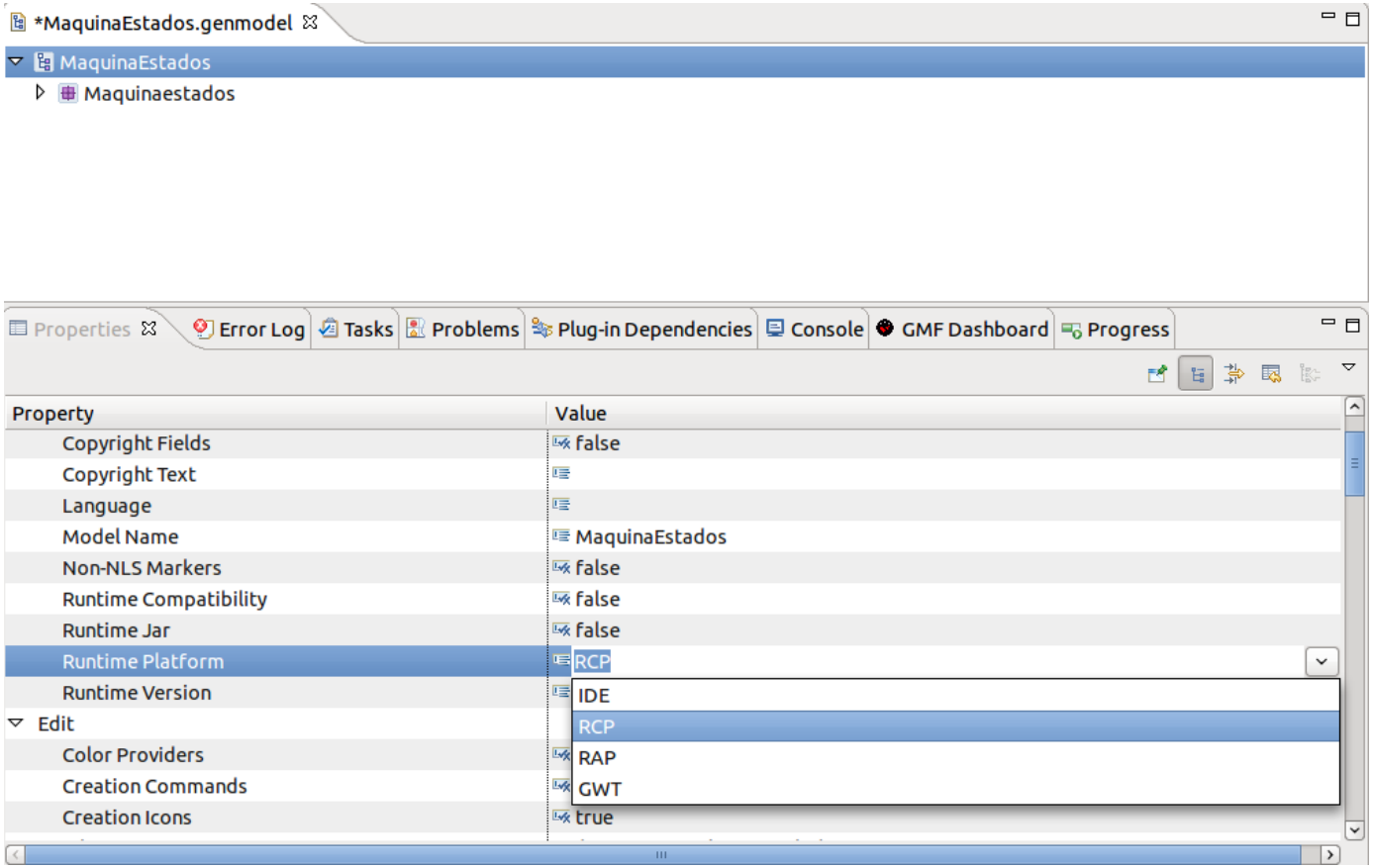
Exportar plugin (II)



CONSTRUCCIÓN DE EDITORES DE MODELOS CON EMF

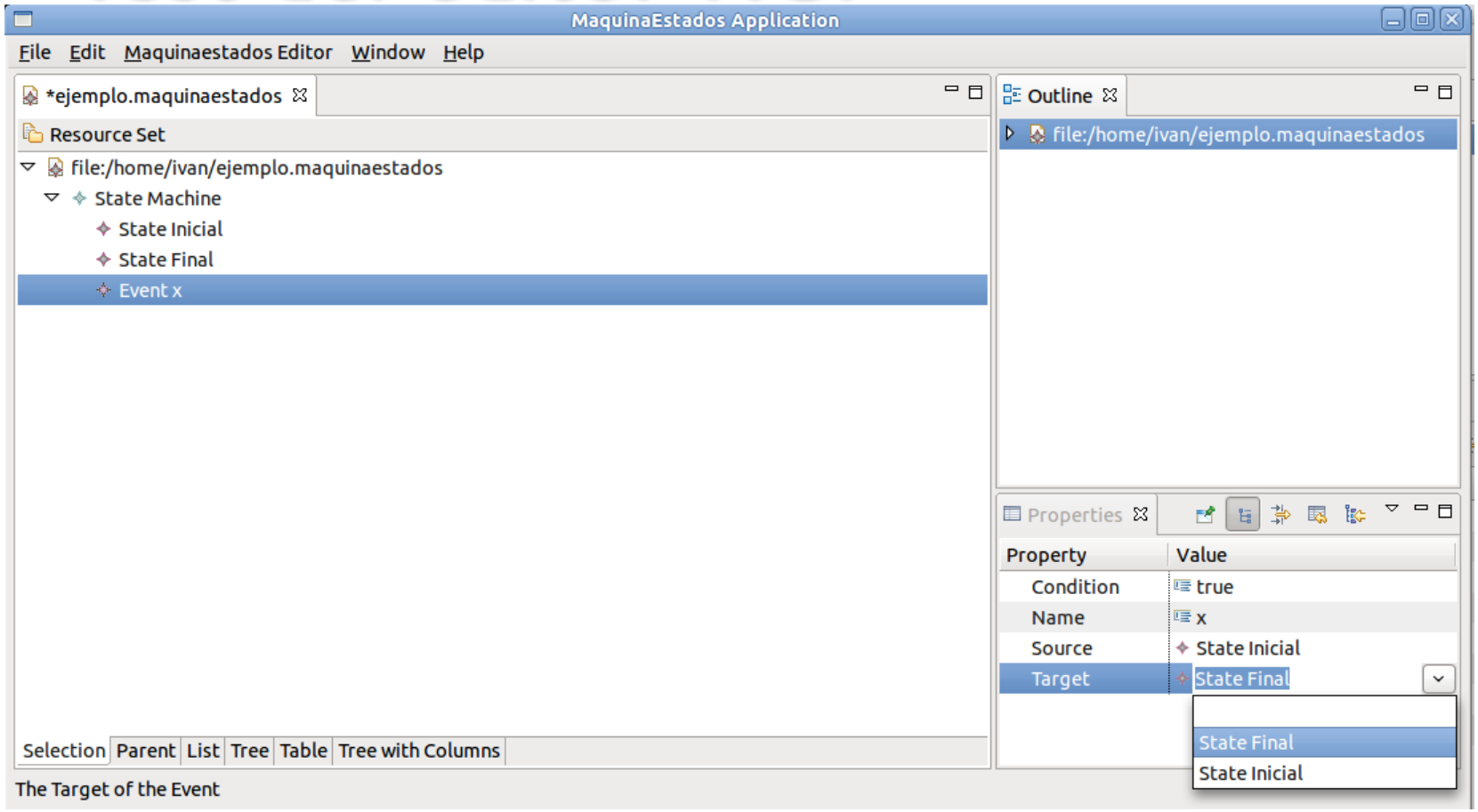
DESPLIEGUE DE EDITORES COMO PRODUCTOS

Generación del editor RCP



Eliminamos los proyectos generados automáticamente (*edit, editor y tests*), cambiamos "Runtime Platform" del *genmodel* a "RCP" y volvemos a generar el código fuente.

Test del editor RCP



`[Project *Editor*] Run as → Eclipse Application`

Nuestro nuevo editor de máquinas de estado aparece como una nueva aplicación de escritorio independiente del IDE de Eclipse.

Personalizar mensajes del RCP

_UI_Application_title = MaquinaEstados Application

_UI_Menu_File_label = &File

_UI_Menu_New_label = &New

_UI_Menu_Open_label = &Open...

_UI_Menu_Open_description = Opens a model object file

_UI_Menu_OpenURI_label = &Open URI...

_UI_Menu_Edit_label = &Edit

_UI_Menu_Window_label = &Window

_UI_Menu_Help_label = &Help

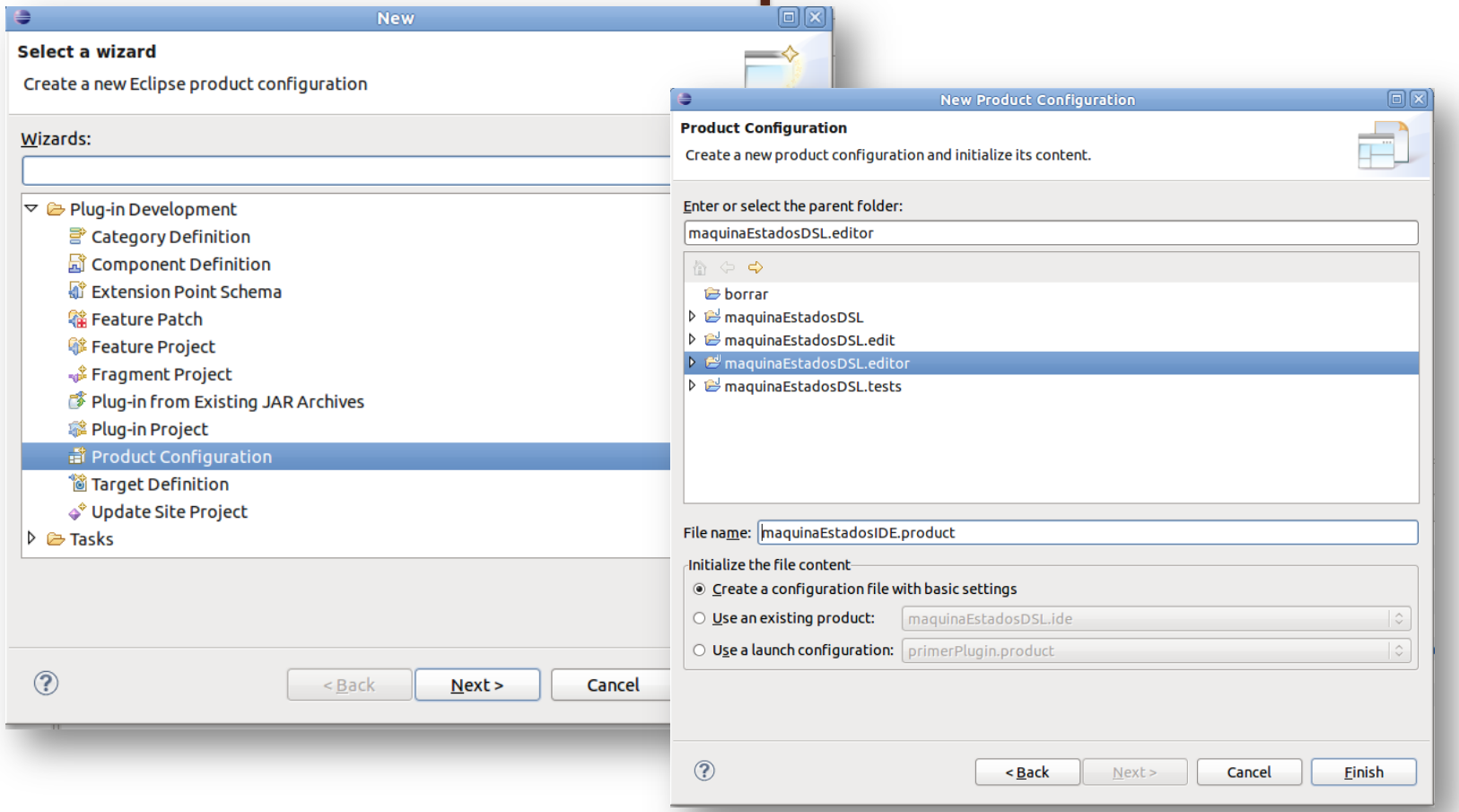
_UI_Menu_About_label = MaquinaEstados &About...

_UI_About_title = MaquinaEstados Application

_UI_About_text = MaquinaEstados Application about box goes here.

[plugin.properties]

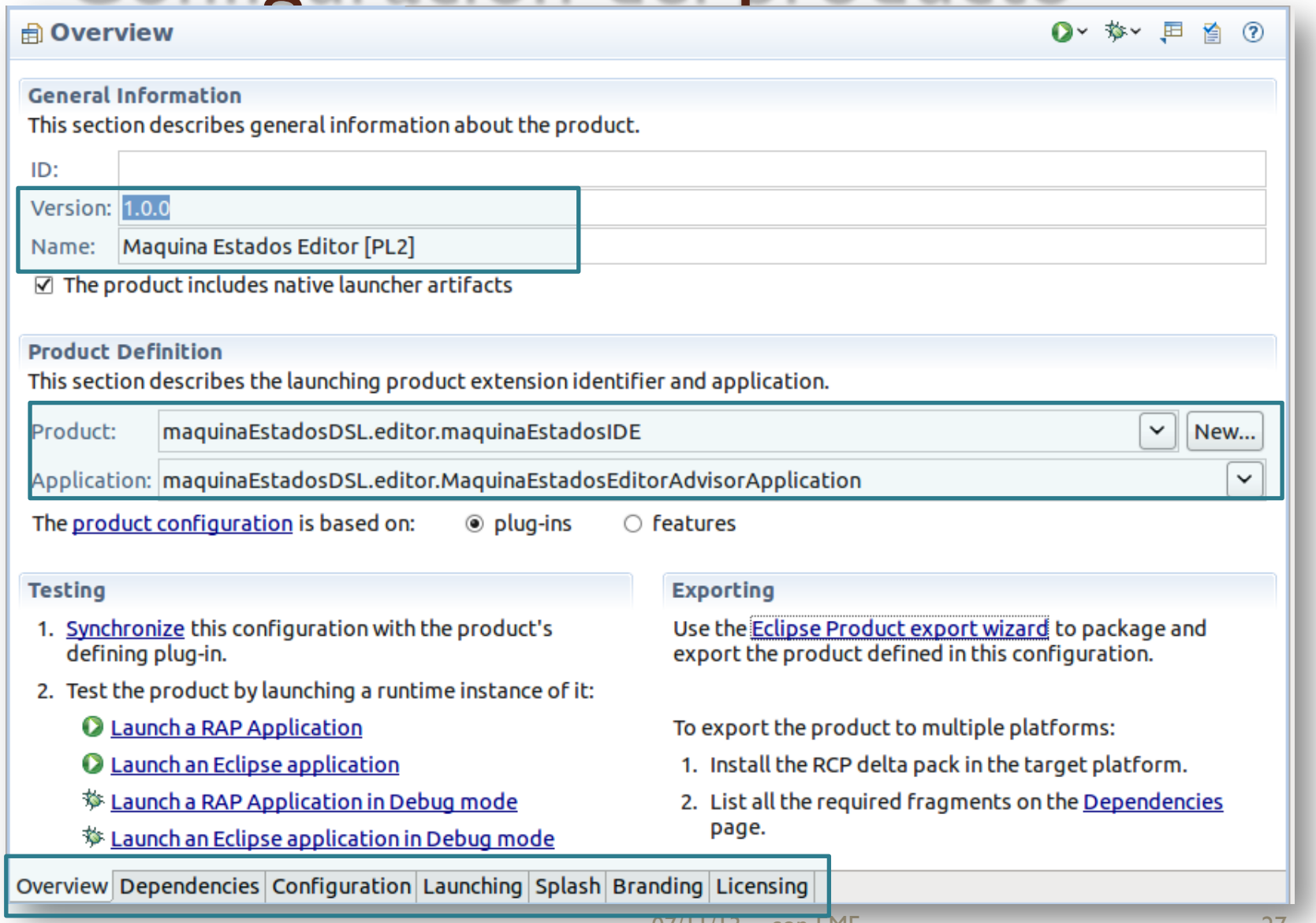
Creación de un producto



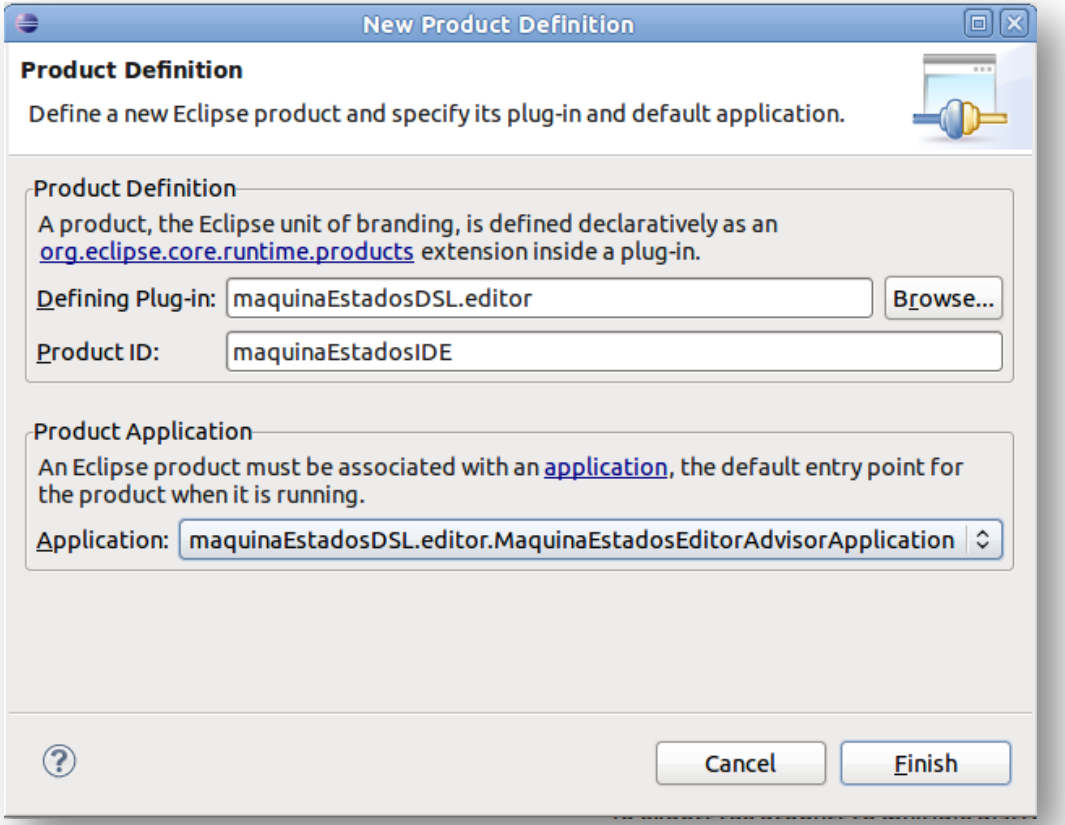
`File → New → Product Configuration`

Los productos definen todos los recursos que formaran parte del despliegue de nuestra aplicación RCP

Configuración del producto

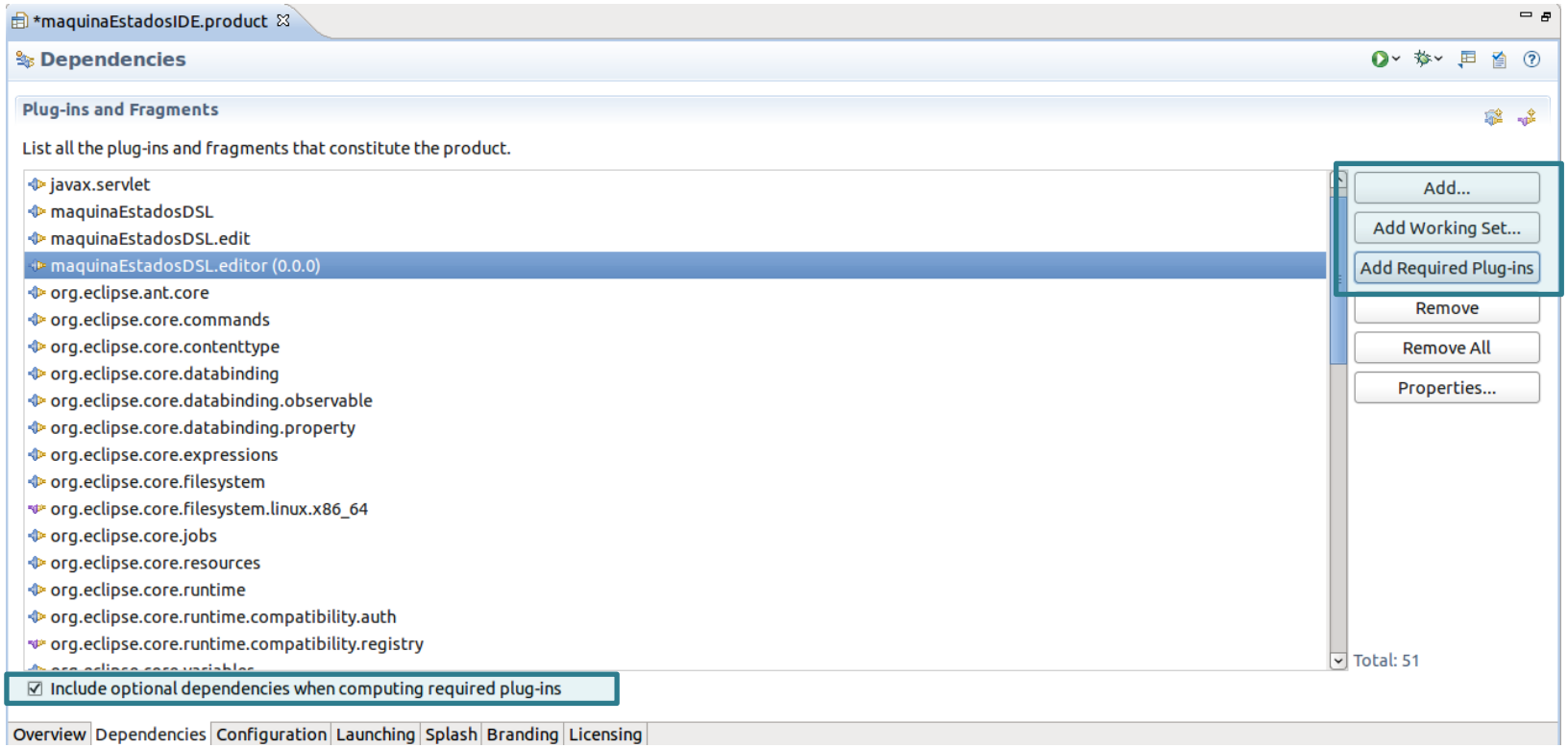


Definición del producto



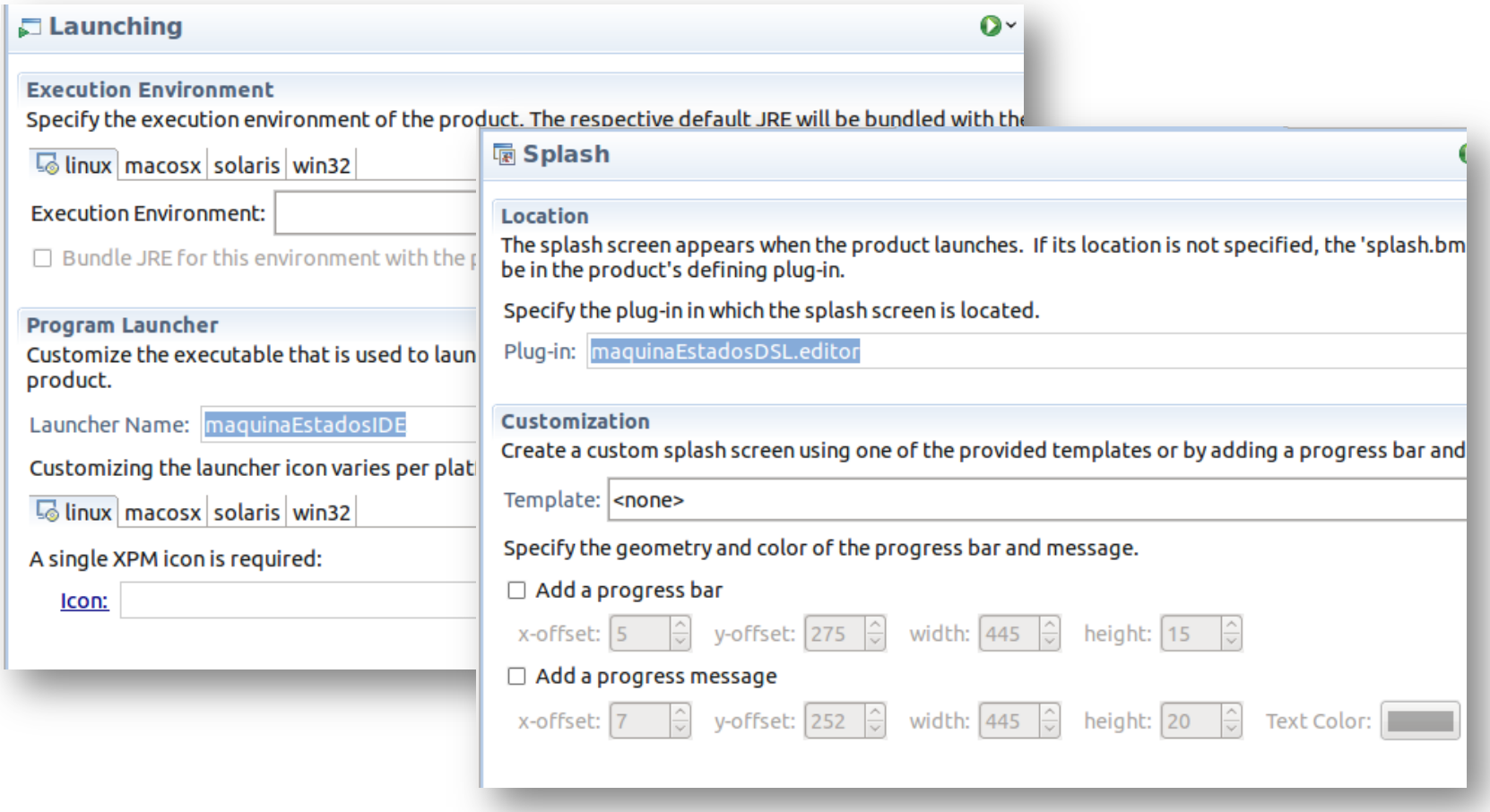
Para definir completamente nuestro producto debemos seleccionar nuestro plugin (proyecto Editor) y un identificador (nombre del fichero .product)

Dependencias del producto



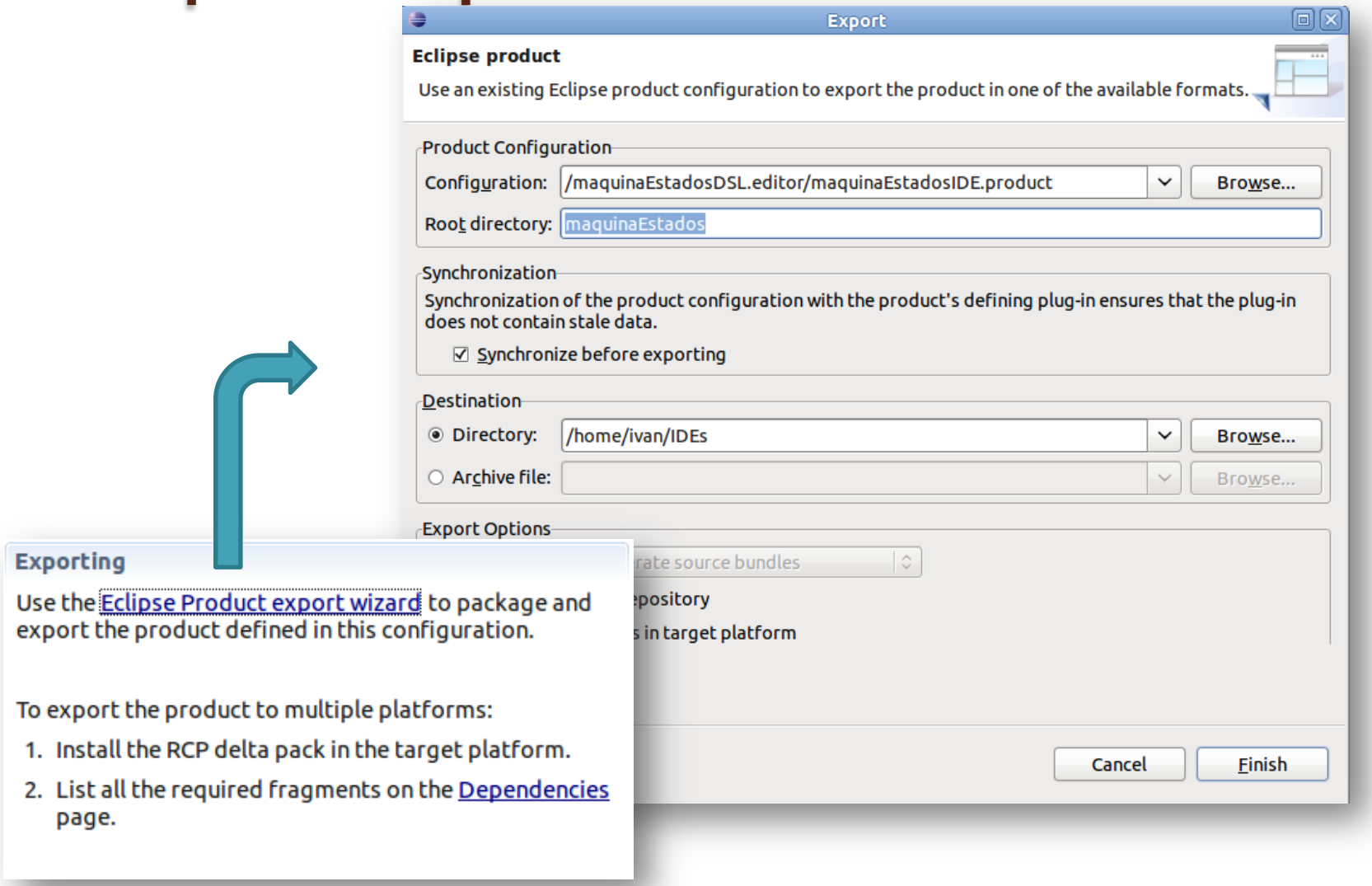
Debemos indicar que nuestro nuevo producto dependerá principalmente del plugin *Editor*. Luego, tenemos que añadir (de forma automática) las dependencias necesarias.

Aspectos adicionales del producto

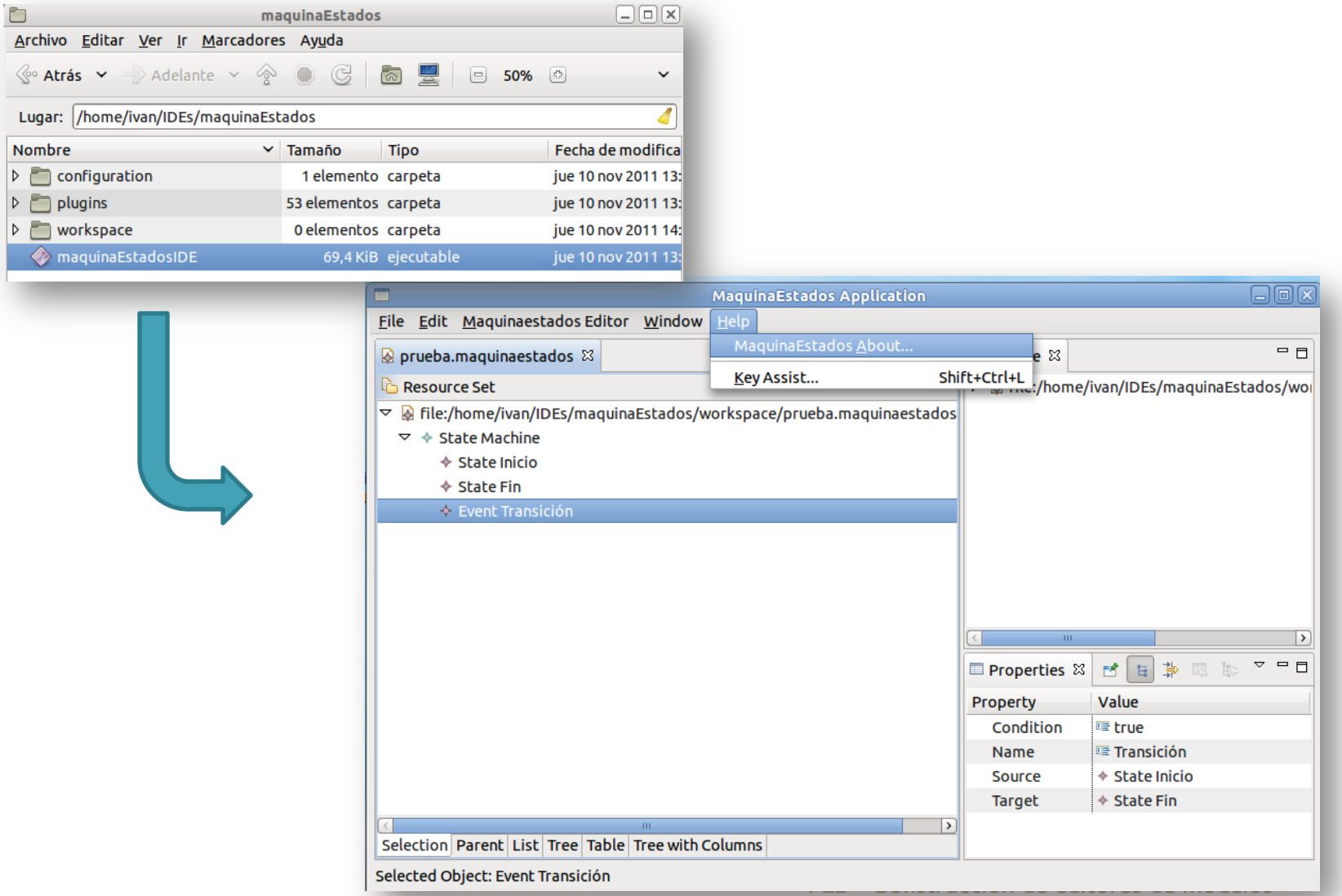


Podemos especificar los distintos entornos de ejecución de nuestro producto, pantalla de inicio, iconos, licencia, etc.

Exportar producto



Exportar producto (III)



CONSTRUCCIÓN DE EDITORES DE MODELOS CON EMF

RESUMEN

¿Qué hemos aprendido hoy?

- Desarrollar editores sencillos de modelos basados en árbol, utilizando Eclipse EMF.
- Desplegar los editores como plugins del propio IDE.
- Desplegar los editores como productos impedientes.

Procesadores de Lenguajes 2

Construcción de editores de modelos con EMF

Curso 2013-2014

Iván Ruiz Rube
ivan.ruiz@uca.es