

Contexto

Una red de petri es un grafo que se usa para representar sistemas distribuidos. Una red de petri está conformada por los siguientes elementos: lugares, transiciones y arcos los cuales son representados por circulos, rectangulos y flechas respectivamente. Una red de petri está regida por las siguientes reglas: Los arcos pueden conectar un lugar a una transición y de igual manera una transición a un lugar, no puede haber arcos entre lugares ni entre transiciones. Los lugares que conecta un arco con una transición son llamados lugares de entrada y los lugares que conecta un arco desde una transición hacia un lugar son llamados lugares de salida. Los lugares y las transiciones tienen nombres y cada transición tiene un tiempo mínimo y máximo de retraso.

La figura 1 muestra una red de petri básica.

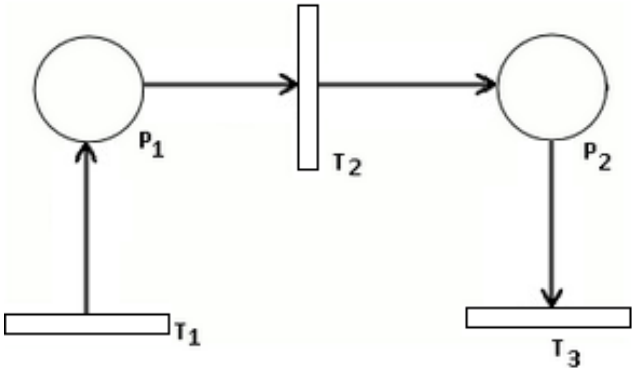


Figura 1

Instrucciones

Tenga en cuenta la representación gráfica del metamodelo de Petrinet:

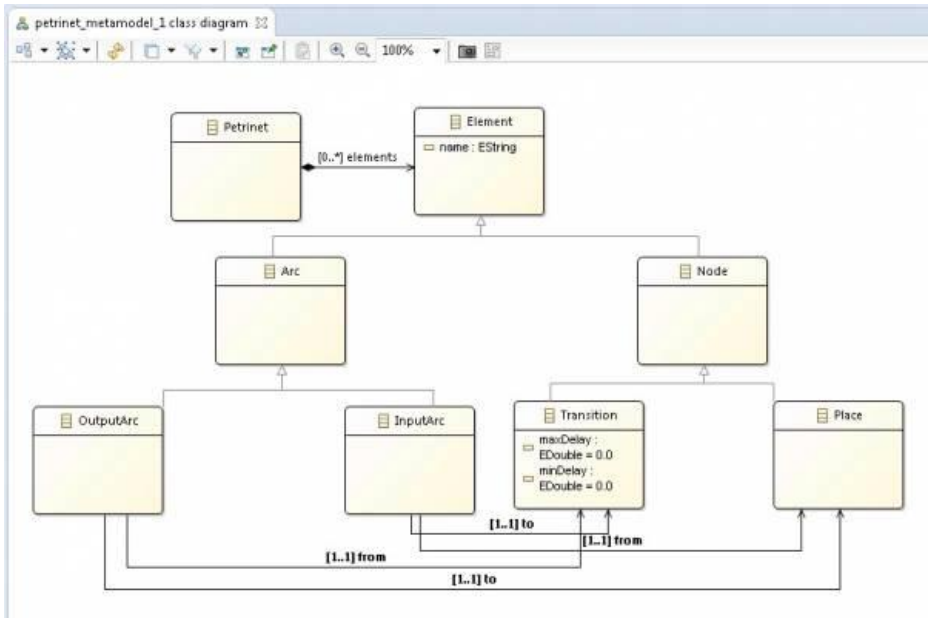


Figura 2

Para crear un modelo conforme al metamodelo de la figura 2 se debe importar el proyecto dado, para ello abra Eclipse y de clic en a File -> Import:

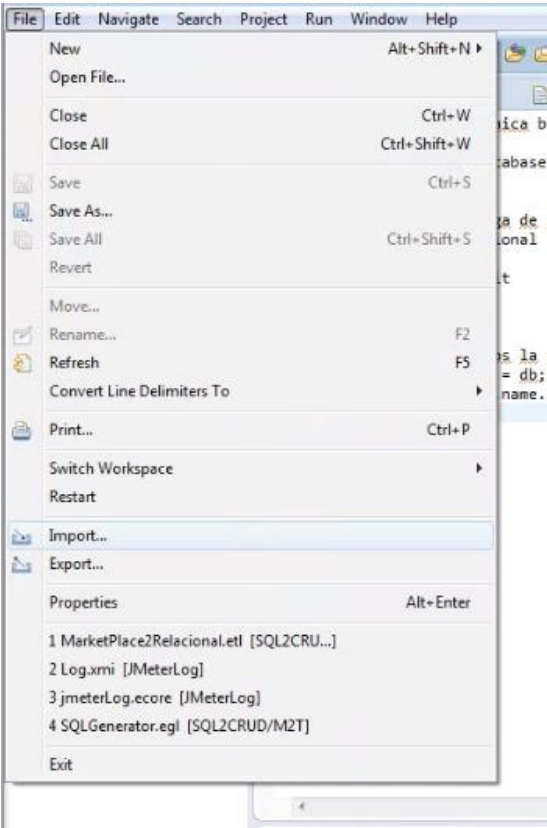


Figura 3

Posteriormente seleccione la opción "Existing Projects into Workspace" dentro de la categoría "General":

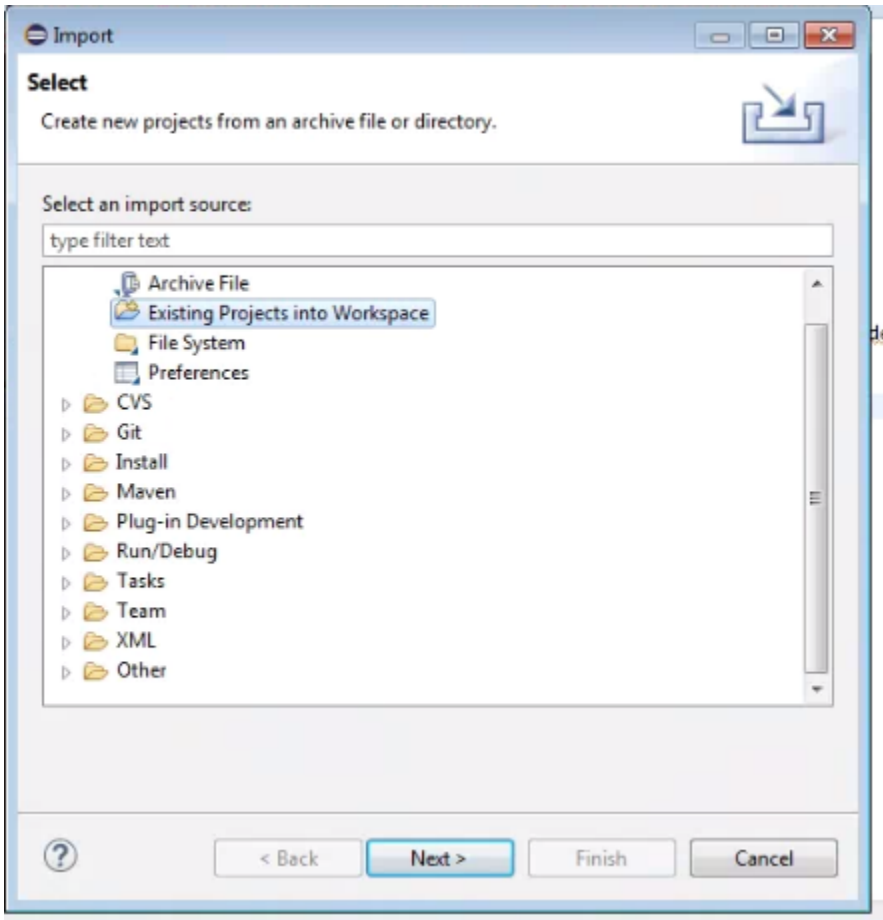


Figura 4

Finalmente, descomprima el proyecto que descargo y luego vuelva a Eclipse, para dar clic en Browse... y ubicar el directorio que contiene al proyecto descomprimido, finalice dando en Finish:

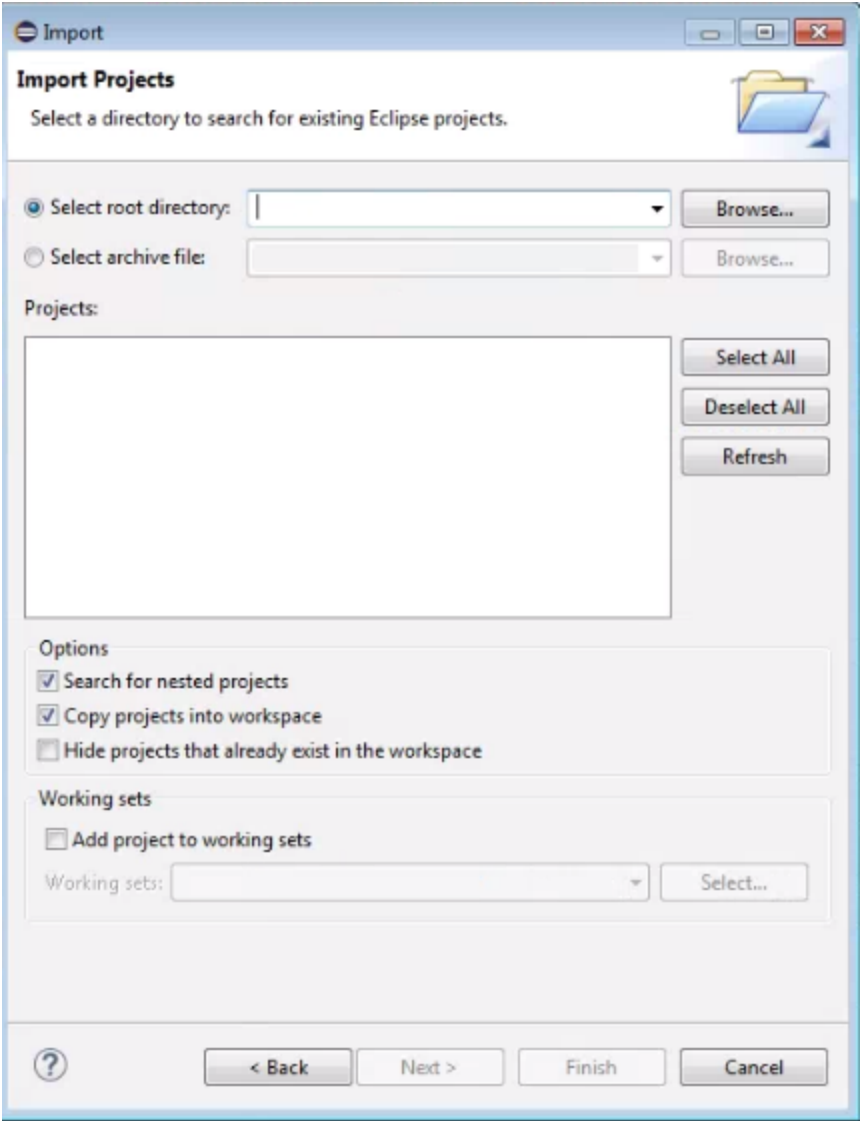


Figura 4

Una vez importado el proyecto, debe cambiarse la perspectiva de Eclipse para hacer uso de las herramientas de modelado, para ello vaya a Window -> Open Perspective -> Other

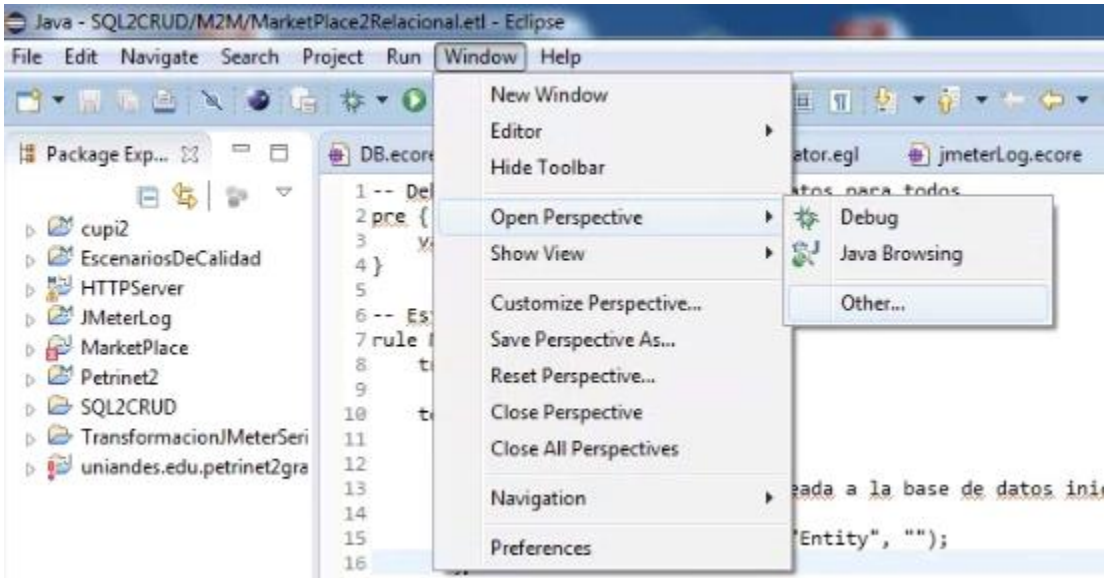


Figura 5

En el dialogo que le aparece seleccione la opción "Modeling":

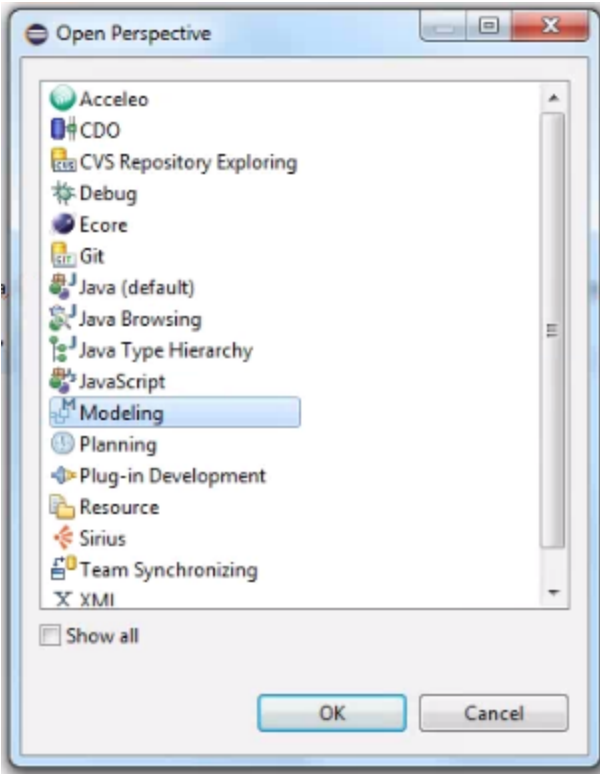
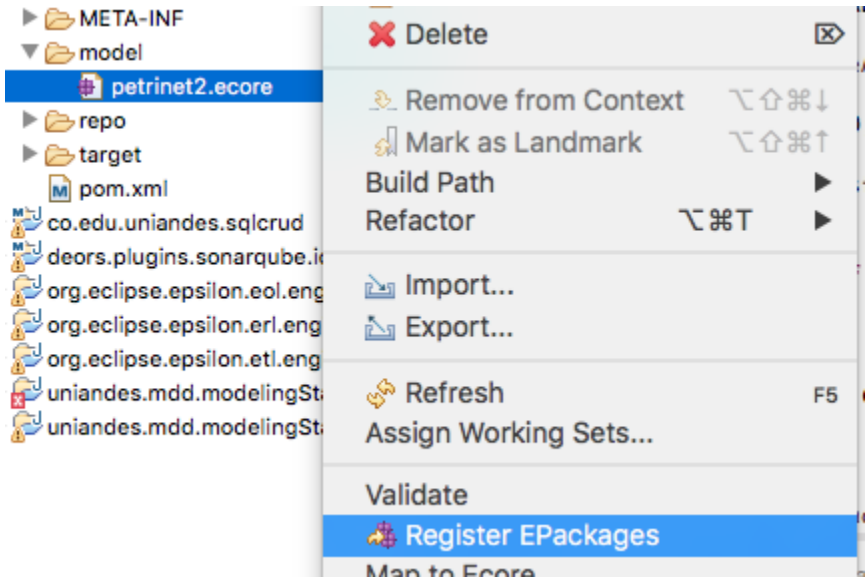


Figura 6

Ahora es necesario registrar el metamodelo de Petrinet para usarlo en los modelos del taller, para ello abramos la carpeta “model” del proyecto, allí veremos un archivo .ecore (este archivo almacena toda la información del metamodelo) hay que dar clic derecho sobre el mismo y seleccionar la opción "Register EPackage":



Al desplegar el metamodelo podremos ver el diagrama del metamodelo propuesto, luego debemos hacer doble click sobre este diagrama de clases de petrinet, como se muestra en la siguiente imagen:

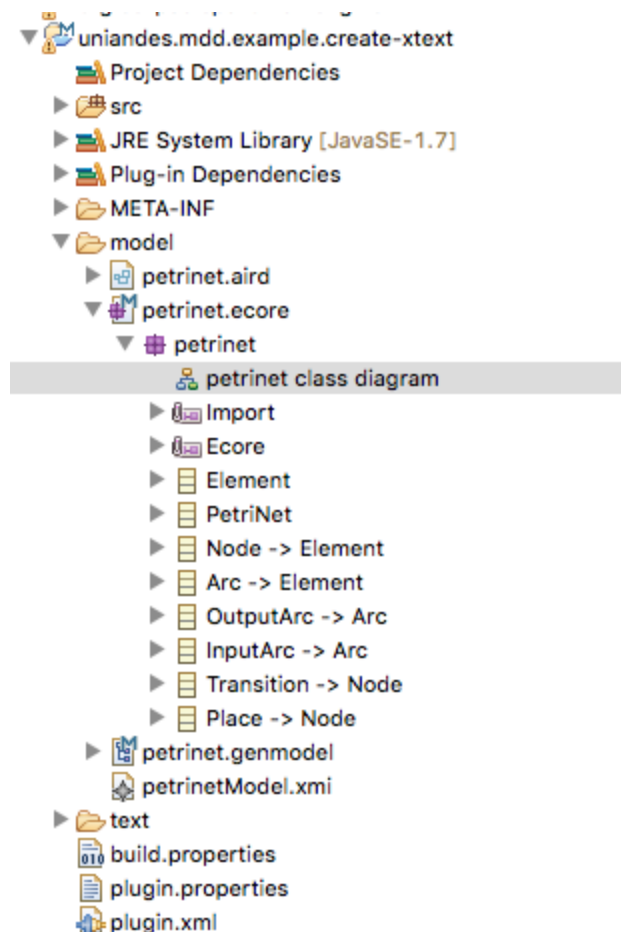


Figura 7

A continuación abrimos el archivo .ecore (dando doble clic sobre el mismo) como se muestra en la figura 8

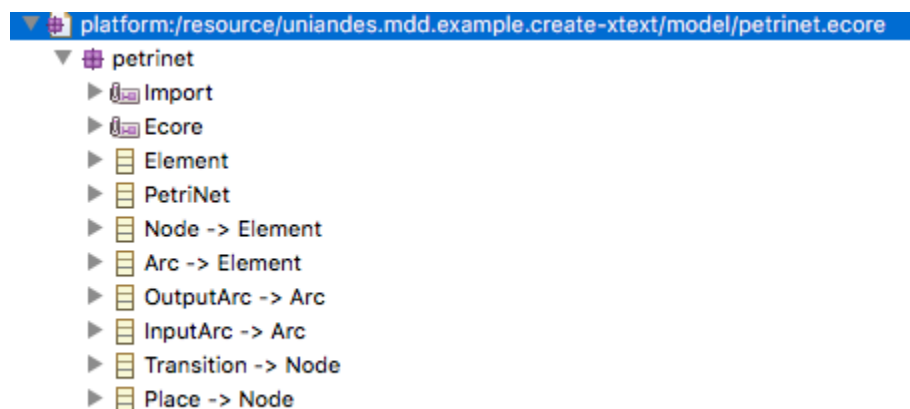


Figura 8

Ahora bien, debemos crear una instancia del elemento raíz del modelo, en este caso debemos buscar el elemento "Petrinet" y a este darle click derecho, posteriormente seleccionamos la opción "Create Dynamic Instance". La figura 9 representa el proceso:

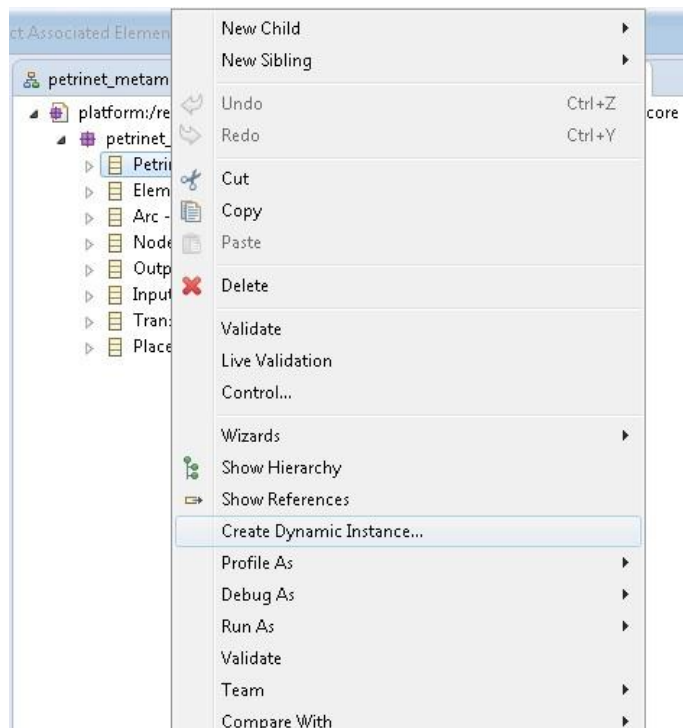
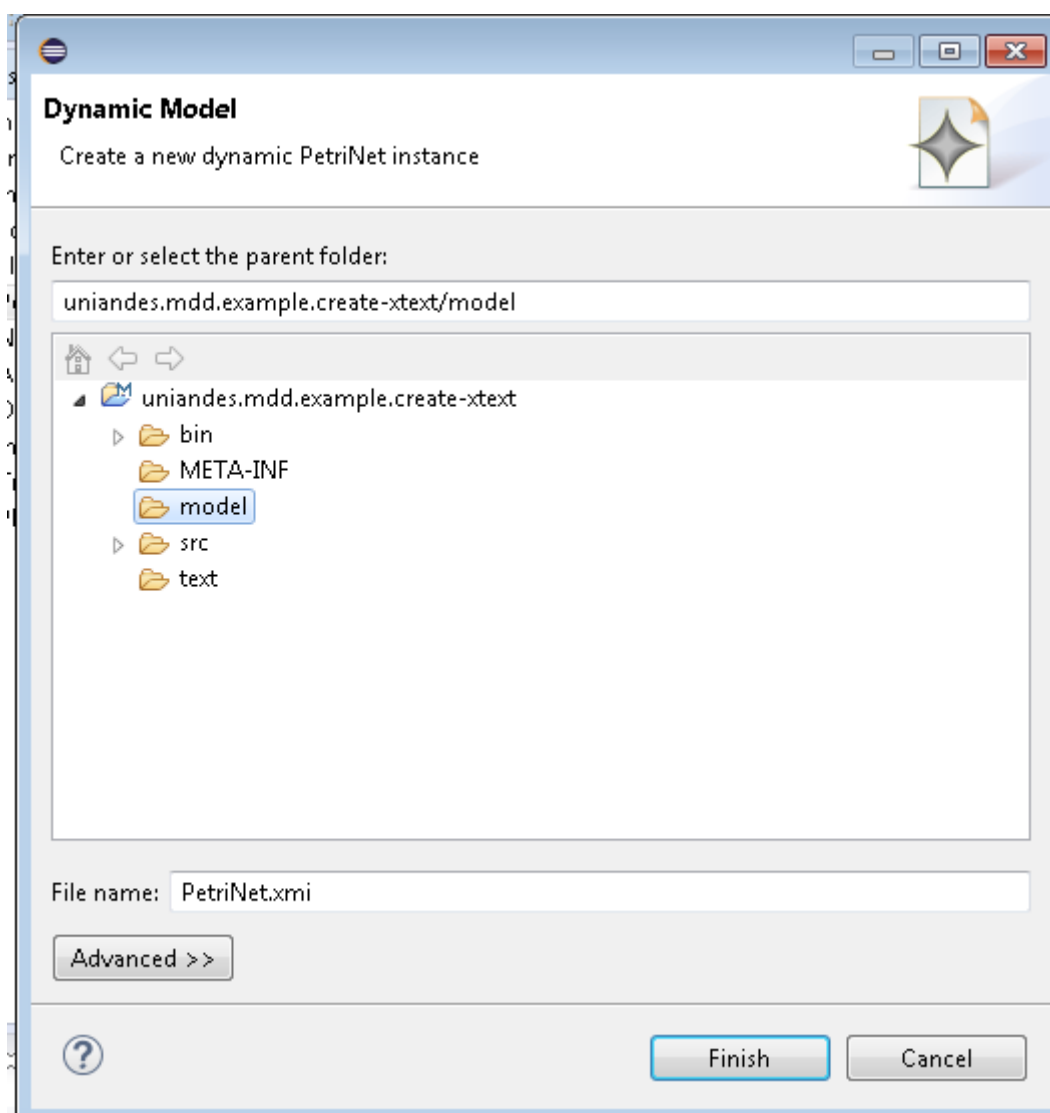


Figura 9

Seleccione la carpeta model y asigne un nombre al modelo que se va a generar, luego de clic en Finish como se muestra en la siguiente imagen:



Esto generará un nuevo archivo con extensión .xml en la carpeta “model” del proyecto. Damos clic derecho en este archivo, seleccionamos la opción “Open With” y por último damos click en “Exeed Editor”, como ve en la figura 10.

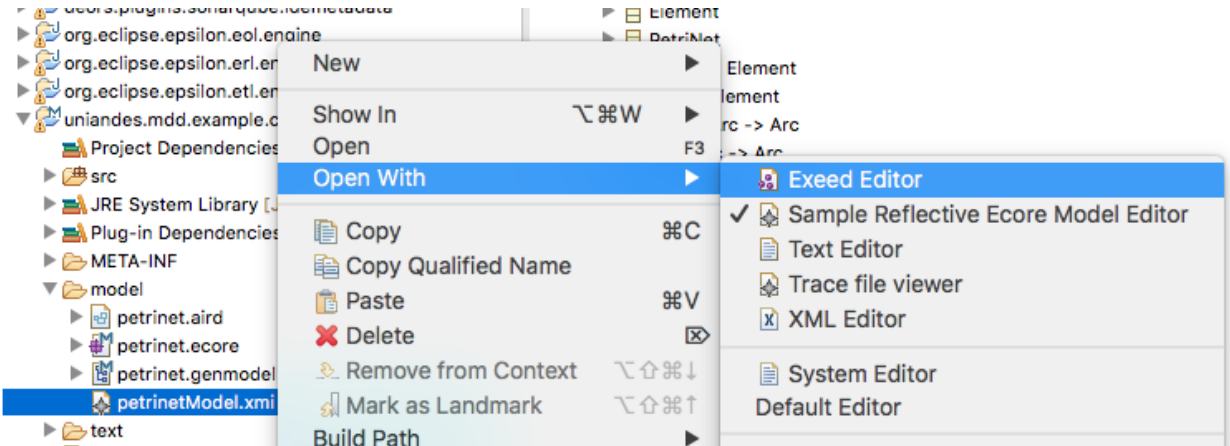


Figura 10

Para finalizar debemos estar en la pestaña del editor de modelos (conforme al metamodelo de Petrinet). Aquí podremos crear, remover o editar elementos para así crear el modelo deseado. Estos elementos son conformes a los conceptos del metamodelo propuesto. Se espera que por cada modelo a crear lo primero que se debe hacer es imaginar una petrinet, la cual tiene “X Places”, “Y Transitions”, “Z Arcs”. Por ejemplo, la imagen adjunta al inicio del taller (Figura 1) ilustra una petrinet que tiene 2 Places, 3 Transitions, y 4 Arcs. Eso quiere decir que en el modelo del editor arborescente se debe crear cada uno de esos elementos en la cantidad indicada y establecer relaciones entre ellos (por ejemplo, debe decir que el Place no. 1 tiene un InputArc y un OutputArc).

En la figura 11, podemos ver como se crean los elementos del modelo, debemos ubicarnos en el elemento principal (del tipo Petrinet) y darle click derecho, al hacerlo podemos ver los elementos que pueden ser hijos del mismo (Arc, Node, OutputArc, InputArc, Transition y/o Place) únicamente debemos seleccionar el que nos interese y ajustar sus atributos para que correspondan al modelo desea.

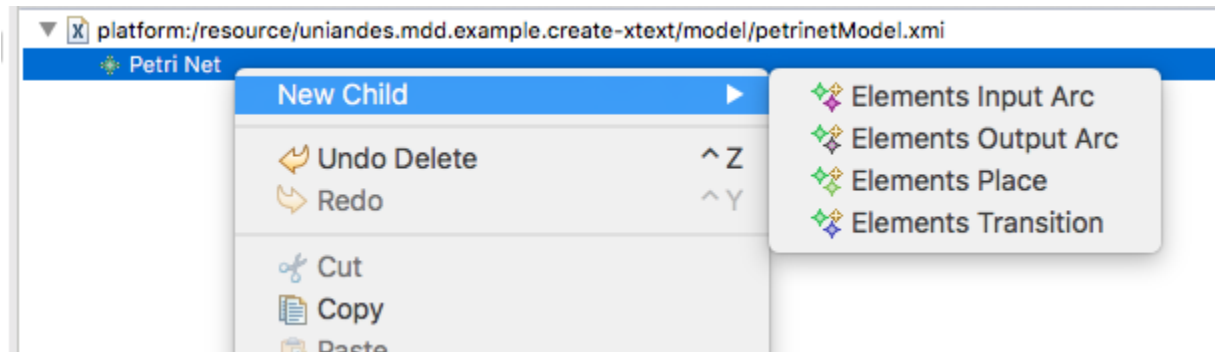


Figura 11

Para parametrizar cada uno de los elementos de una instancia, basta con usar la vista de propiedades con la cual podremos ajustar los parametros de los elementos basandonos en nuestra idea. Si en Eclipse no tiene habilitada la pestaña de propiedades, debe dar clic en Window -> Show View -> Other, como lo muestra la Figura 12:

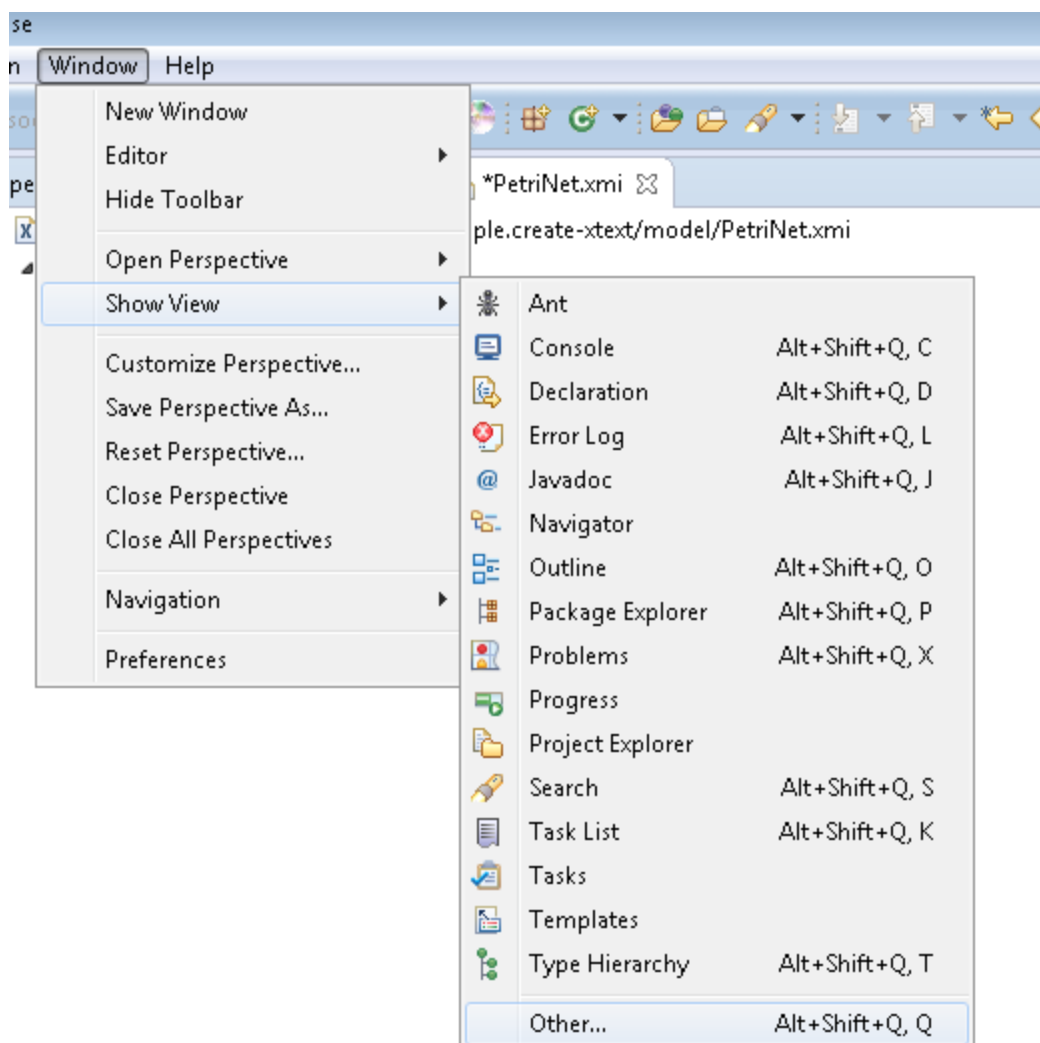


Figura 12

Ahora debe buscar la vista de propiedades (Properties), únicamente debe escribir en el campo de búsqueda "Properties", seleccionarla y dar clic en ok:

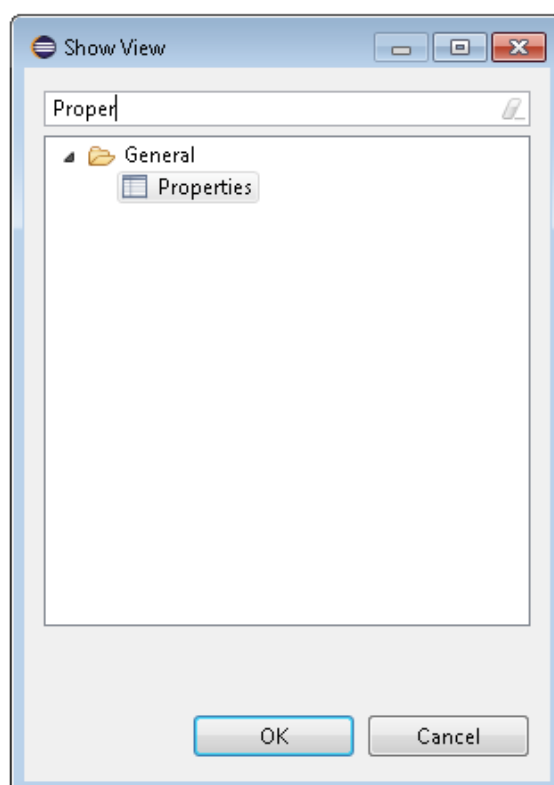


Figura 13

Finalmente aparecerá la pestaña "Properties", allí será posible configurar las propiedades de cada elemento, basta con seleccionarlo y modificar los campos de texto o selección respectivos:

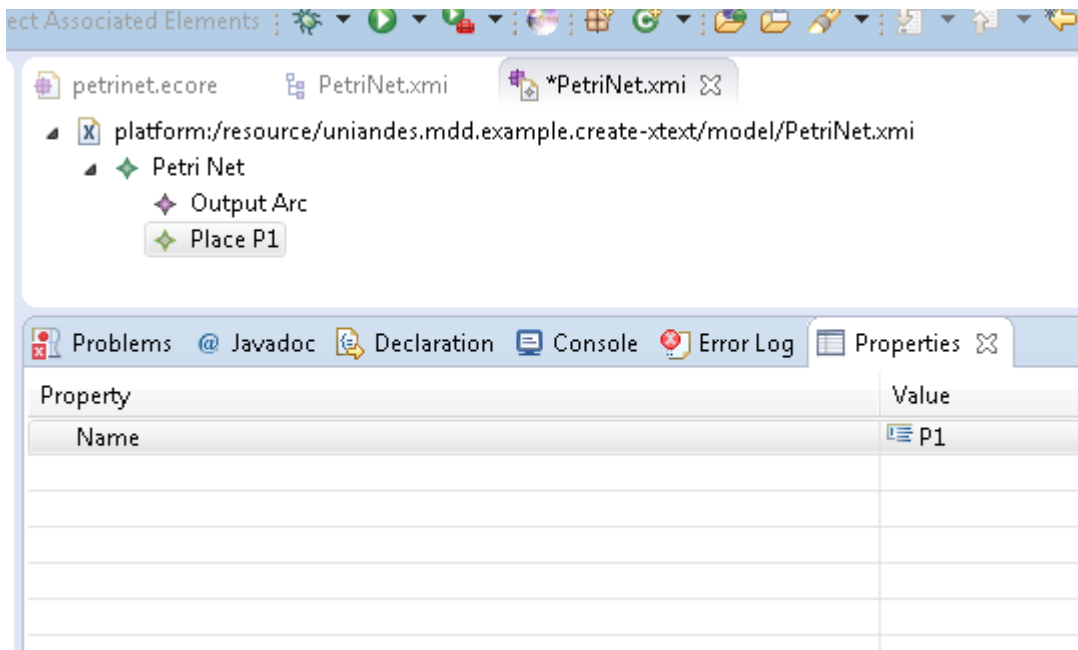


Figura 14

Esta pestaña también nos servirá para hacerle modificaciones a los metamodelos (.ecore), para hacerlo debemos ubicar el archivo Ecore que deseamos modificar, darle clic derecho Open With -> Exeed Editor:

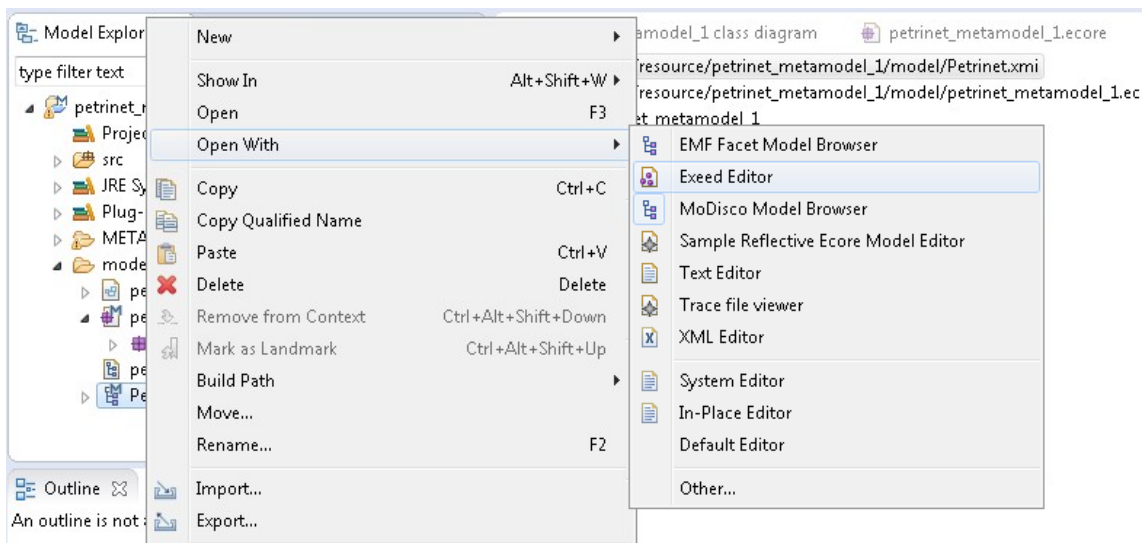


Figura 15

Una vez dentro podemos seleccionar los elementos del ecore y modificar sus propiedades de la misma manera, como se ve en la figura 16 hay algunas propiedades heredadas del concepto de metamodelo que serán mostradas en adición a las que agreguemos:

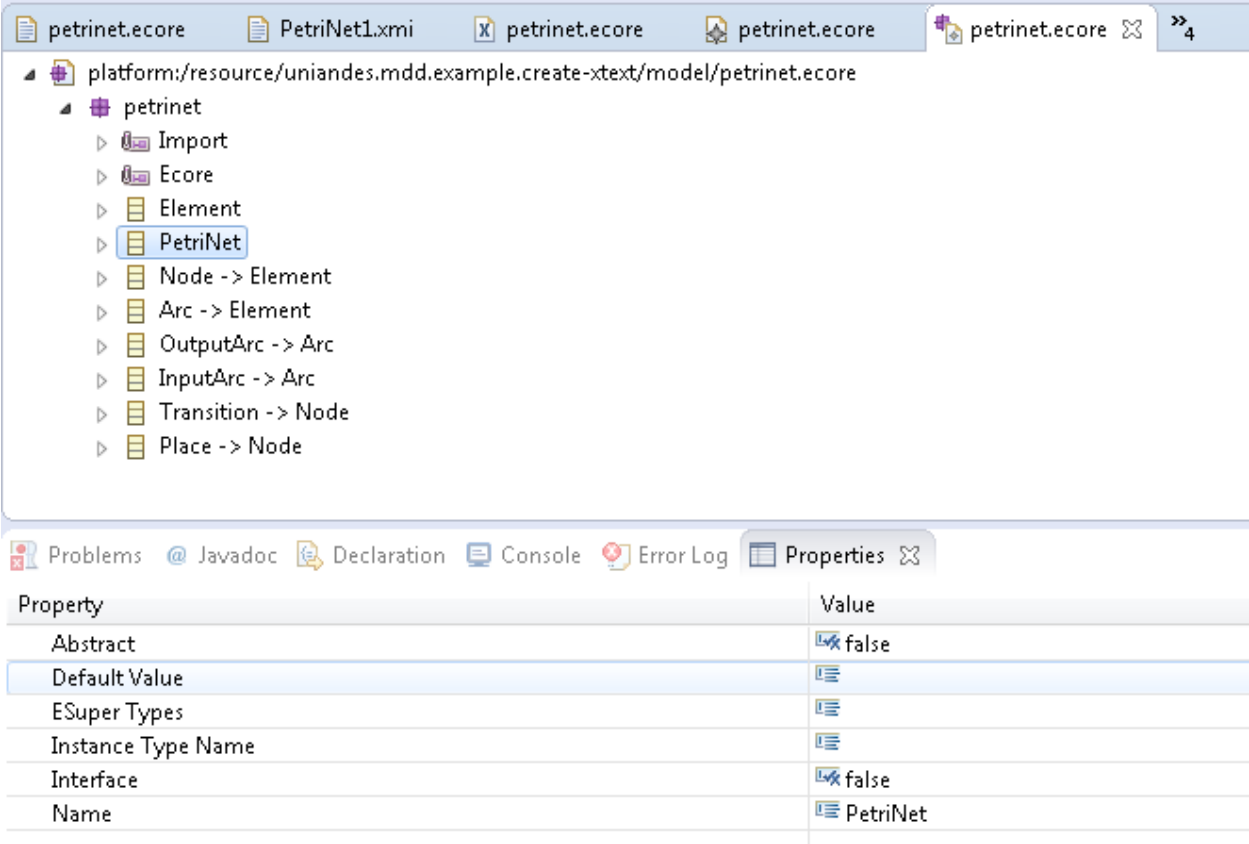


figura 16

Adicionalmente es posible modificar directamente los .xml utilizando texto directamente, para usar este metodo debemos darle click derecho al archivo deseado, seleccionamos Open With -> Text Editor:

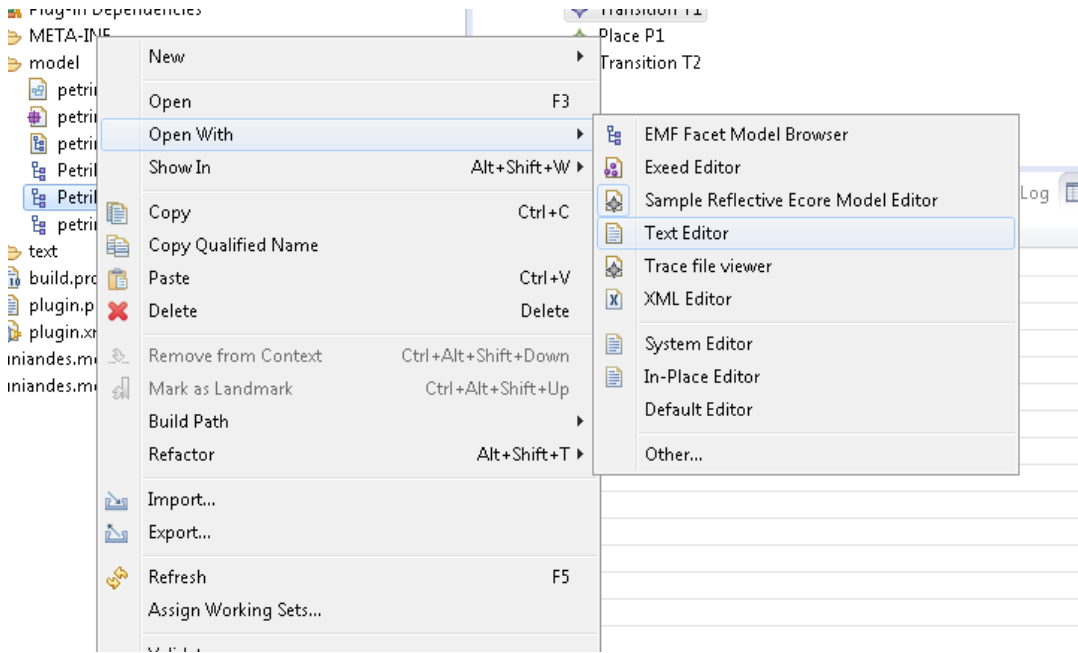


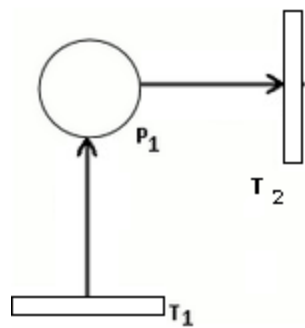
Figura 17

Una vez se abra el editor, será posible hacer modificaciones directamente en el archivo de texto, como se muestra en la siguiente imagen para un diagrama de Petrinet con 5 elementos:

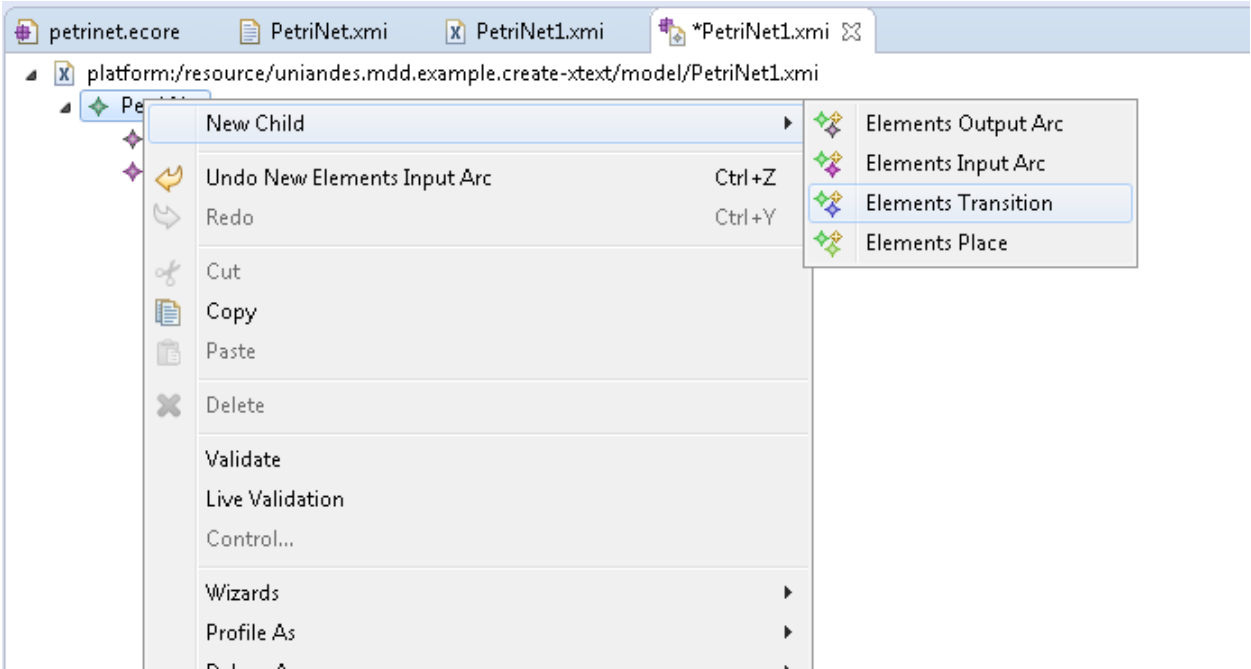
```
1 <?xml version="1.0" encoding="ASCII"?>
2 <petrinet:PetriNet
3   xsi:version="2.0"
4   xmlns:xmi="http://www.omg.org/XMI"
5   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6   xmlns:petrinet="http://www.example.org/petrinet">
7   <elements
8     xsi:type="petrinet:OutputArc"
9     from="//@elements.2"
10    to="//@elements.3"/>
11   <elements
12     xsi:type="petrinet:InputArc"
13     to="//@elements.4"
14     from="//@elements.3"/>
15   <elements
16     xsi:type="petrinet:Transition"
17     name="T1"/>
18   <elements
19     xsi:type="petrinet:Place"
20     name="P1"/>
21   <elements
22     xsi:type="petrinet:Transition"
23     name="T2"/>
24 </petrinet:PetriNet>
25
```

Figura 18

Ahora bien, ilustremos cómo poblar el modelo creado en el tutorial para la siguiente petrinet:

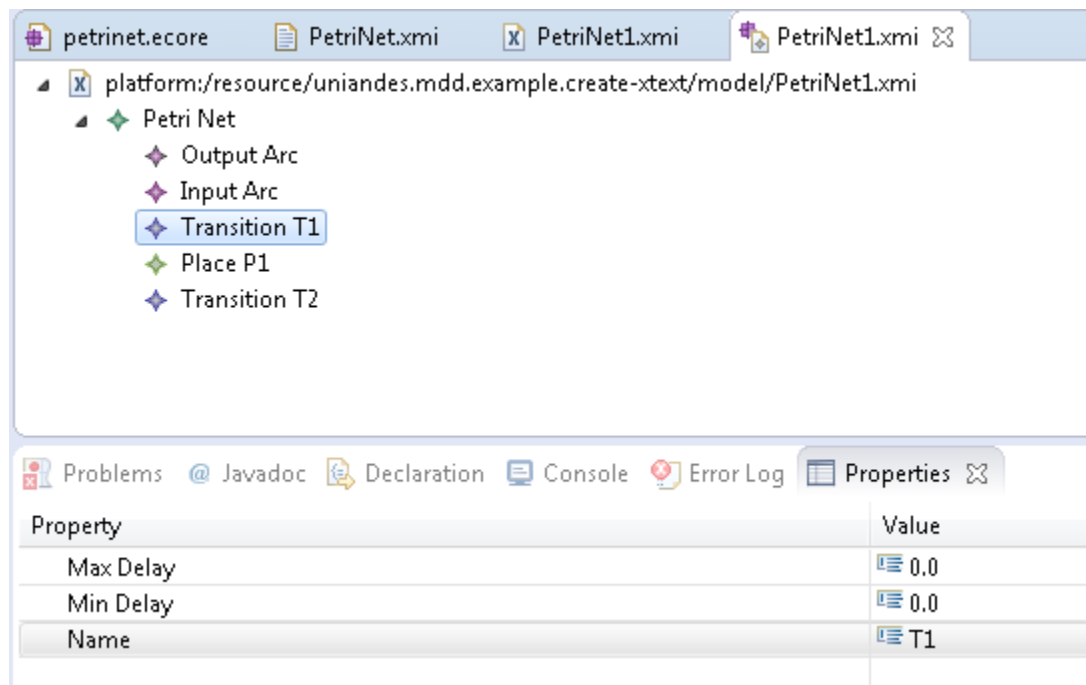


Para hacerlo, únicamente hay que agregar 5 elementos: 2 transition (T1 y T2), 1 place (P1), 1 output Arc (comunicando T1 con P1) y 1 InputArc (comunicando P1 con T2), de la siguiente manera:



Al haber creado los 5 elementos se deben modificar las propiedades de cada uno para ajustarlas a nuestro modelo, el resultado debe ser el siguiente:

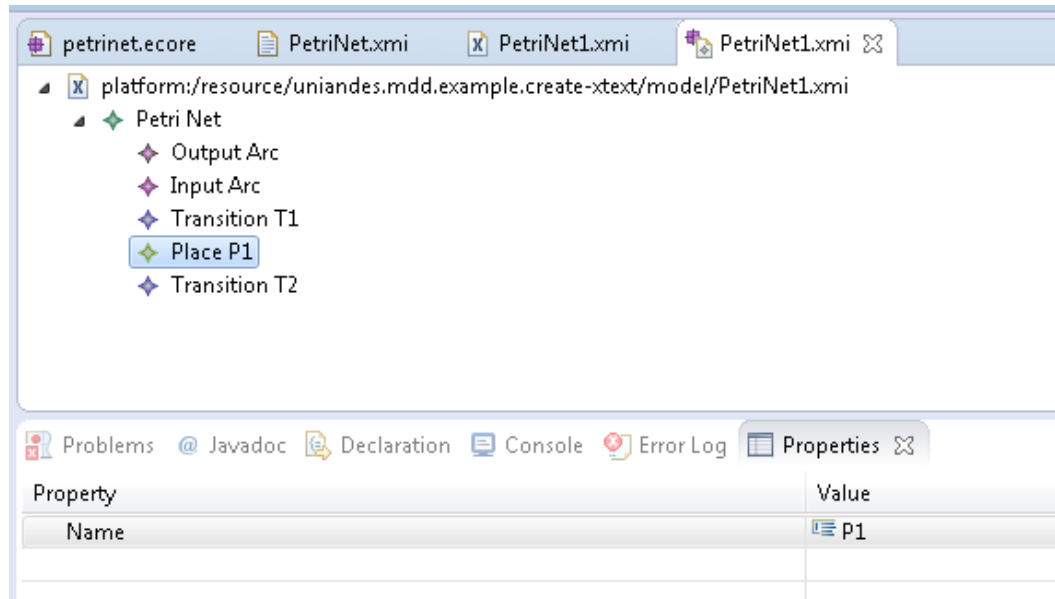
Para T1



The screenshot shows the IDE interface with the PetriNet1.xmi file selected in the project explorer. The Properties panel at the bottom right displays the following properties:

Property	Value
Max Delay	0.0
Min Delay	0.0
Name	T1

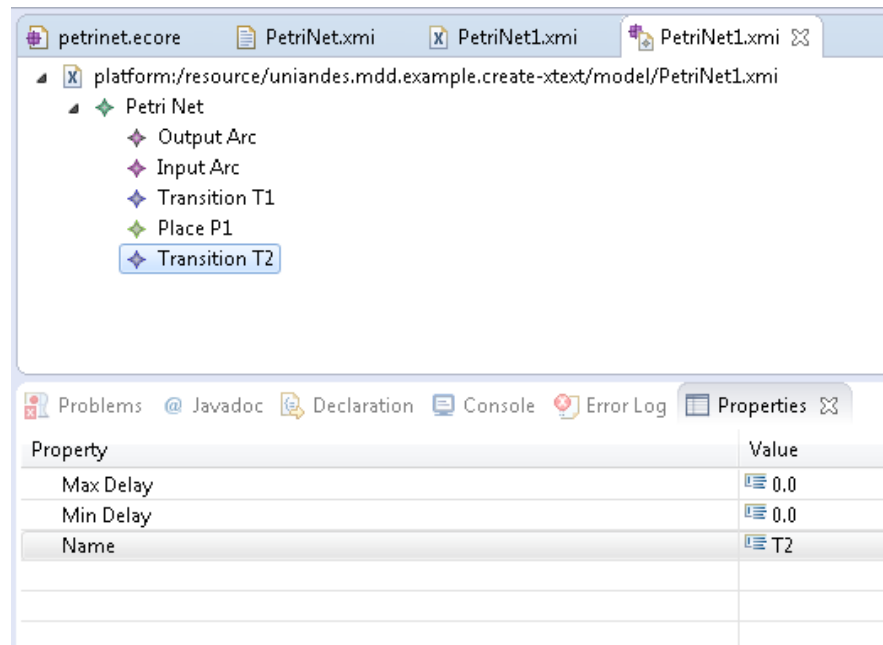
Para P1



The screenshot shows the IDE interface with the PetriNet1.xmi file selected in the project explorer. The Properties panel at the bottom right displays the following properties:

Property	Value
Name	P1

Para T2



The screenshot shows the IDE interface with the PetriNet1.xmi file selected in the project explorer. The Properties panel at the bottom right displays the following properties:

Property	Value
Max Delay	0.0
Min Delay	0.0
Name	T2

Para InputArc (P1 -> T2)

petrinet.ecorePetriNet.xmiPetriNet1.xmiPetriNet1.xmi

platform:/resource/uniandes.mdd.example.create-xtext/model/PetriNet1.xmi

Petri Net

Output Arc

Input Arc

Transition T1

Place P1

Transition T2

ProblemsJavadocDeclarationConsoleError LogProperties

Property	Value
From	Place P1
Name	
To	Transition T2

Para OutputArc (T1 -> P1)

petrinet.ecorePetriNet.xmiPetriNet1.xmiPetriNet1.xmi

platform:/resource/uniandes.mdd.example.create-xtext/model/PetriNet1.xmi

Petri Net

Output Arc

Input Arc

Transition T1

Place P1

Transition T2

ProblemsJavadocDeclarationConsoleError LogProperties

Property	Value
From	Transition T1
Name	
To	Place P1