
Capítulo 5

Ejemplo Tic-Toc

Para comprender el funcionamiento de la herramienta y el proceso inicial de simulación, el primer ejemplo que se documentará será el ejemplo clásico de Tic-Toc, el cual es un ejemplo sencillo a la hora de adentrarse con esta herramienta; consiste en la disposición de dos módulos simples que van a transmitir, de un lado al otro, un mensaje. A continuación se presentan los pasos necesarios para efectuar la correcta instalación y ejecución del ejemplo.

Primero, se crea un nuevo proyecto tipo OMNeT++ project, como se muestra en la Figuras 40, 41 y 42.

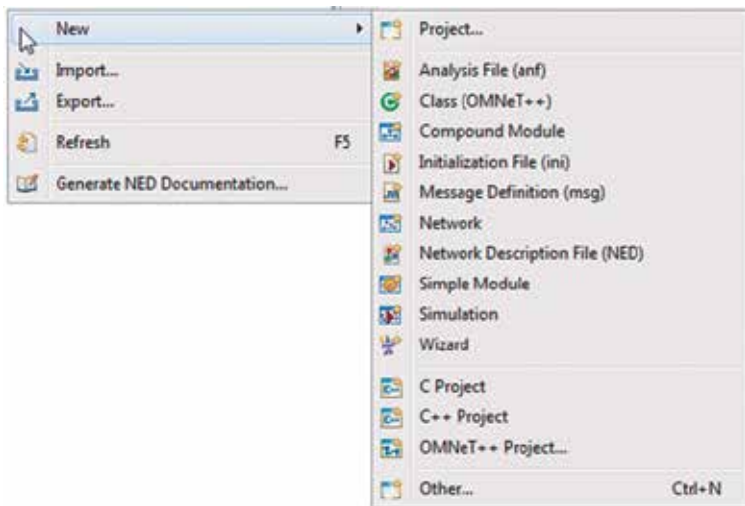


Figura 40. Creación de un proyecto tipo OMNeT++

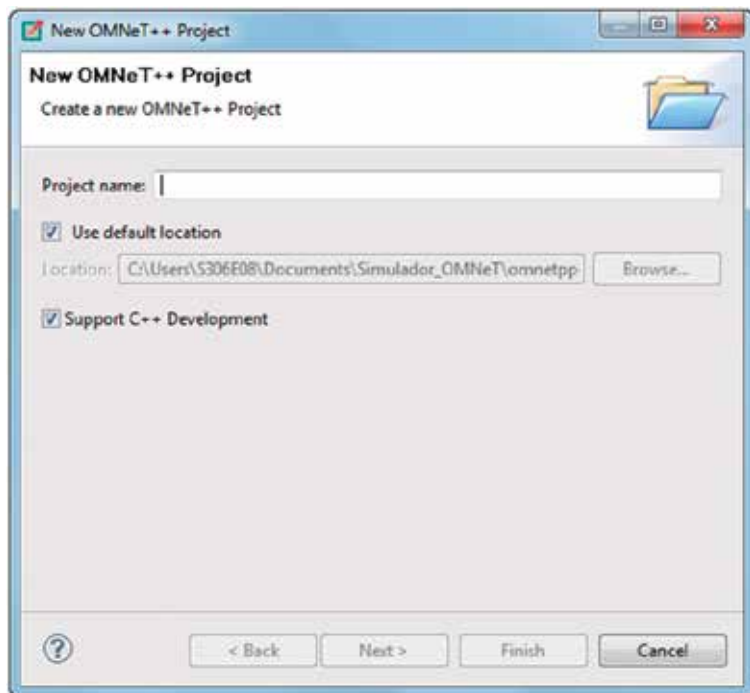


Figura 41. Ventana de creación de un proyecto tipo OMNeT++ (1 de 2)

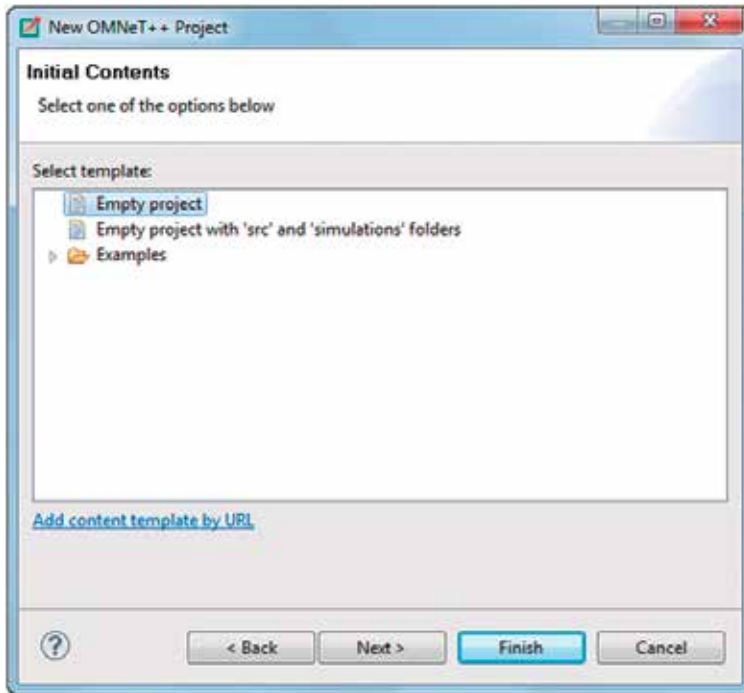


Figura 42. Ventana de creación de un proyecto tipo OMNeT++ (2 de 2)

Una vez finalizada la creación del proyecto se debe observar en el explorador del proyecto lo que se muestra la Figura 43.

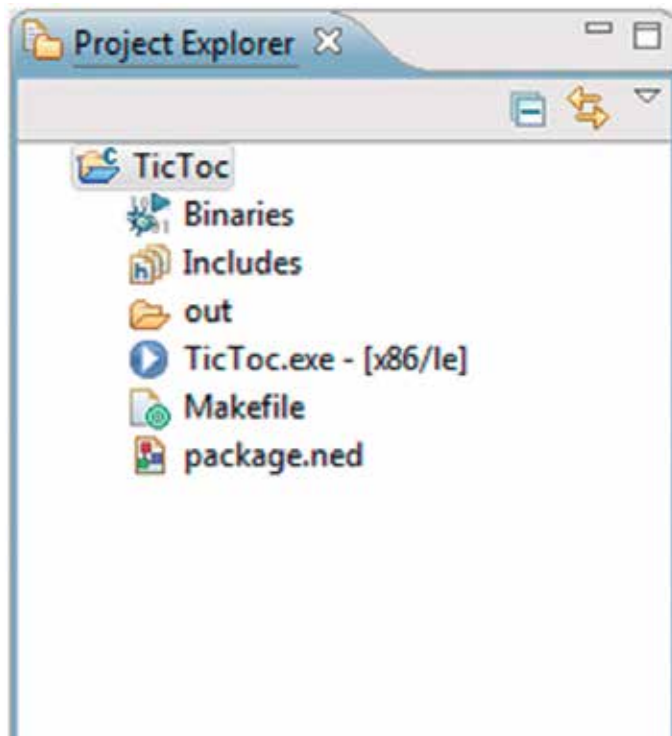


Figura 43. Project Explorer del proyecto TicToc

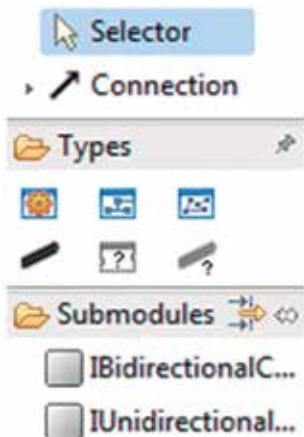


Figura 44. Ventana de utilidades del proyecto

Luego se debe abrir el archivo con extensión *.ned* (*package.ned*). En la parte de Utilidades, cuando se abra el archivo con extensión *.ned* se debe apreciar lo que muestra la Figura 44.

Para la creación de un proyecto OMNeT++ se deben tener en cuenta los siguientes componentes:

- » un archivo con extensión *.ned*;
- » un archivo con extensión *.cc*; y
- » un archivo con extensión *.ini*

Se inicia la creación del archivo con extensión *.ned*, donde va la simulación (parte grafica); en este espacio es donde se definen los módulos que se van a usar en la simulación. Las Figuras 45 y 46 muestran cómo proceder para la creación del archivo *.ned*

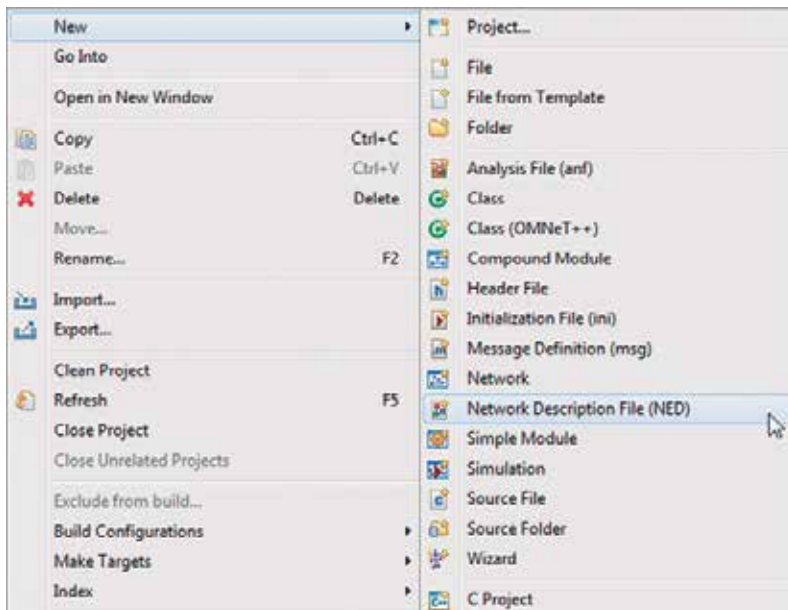


Figura 45. Creación del archivo *.ned*

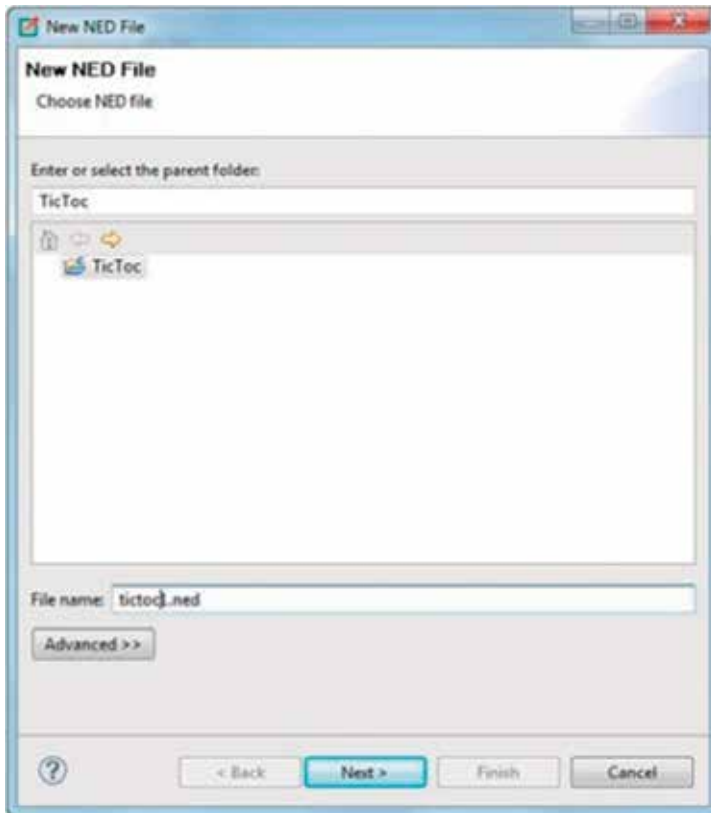
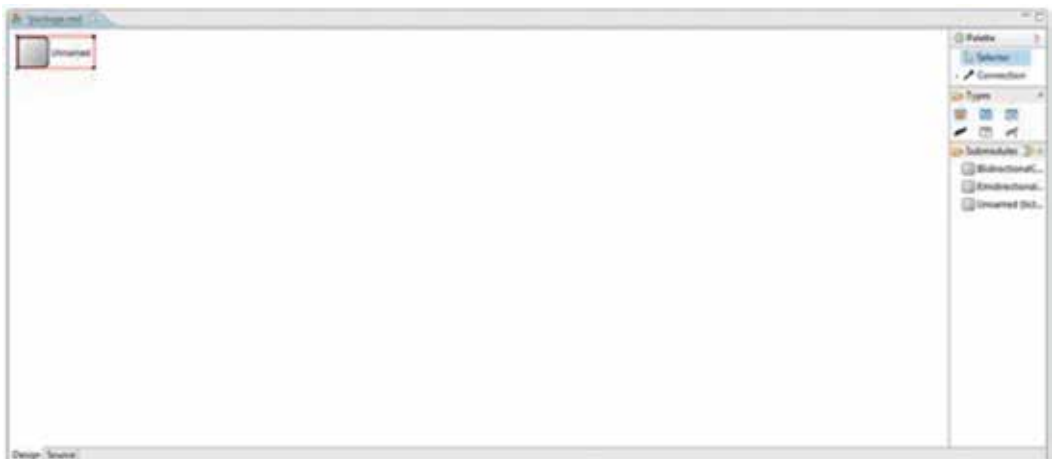


Figura 46. Ventana de creación del archivo .ned

Una vez creado el archivo se observa lo que se muestra la Figura 47. Como se puede observar en ella, hay dos pestañas en la parte inferior, una, *Design*, donde se encuentra la parte gráfica, el diseño (que es la que se observa en la figura), y otra, *Source* que hace referencia al código fuente de ese diseño.

Figura 47. Ventana de diseño del archivo .ned



Como lo que se quiere en este ejemplo es tener un módulo simple que tenga un puerto de entrada y otro de salida, para realizar el traspaso de mensajes —que es en lo que consiste este ejemplo—, en la pestaña *Source* se va a disponer el código que se presenta a continuación:

```
// This program is free software: you can redistribute it and/or modify
// it under the terms of the GNU Lesser General Public License as publish
ed by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; Without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public Licen
se
// along with this program. If not, see http://www.gnu.org/licenses/.
//

package tictoc;

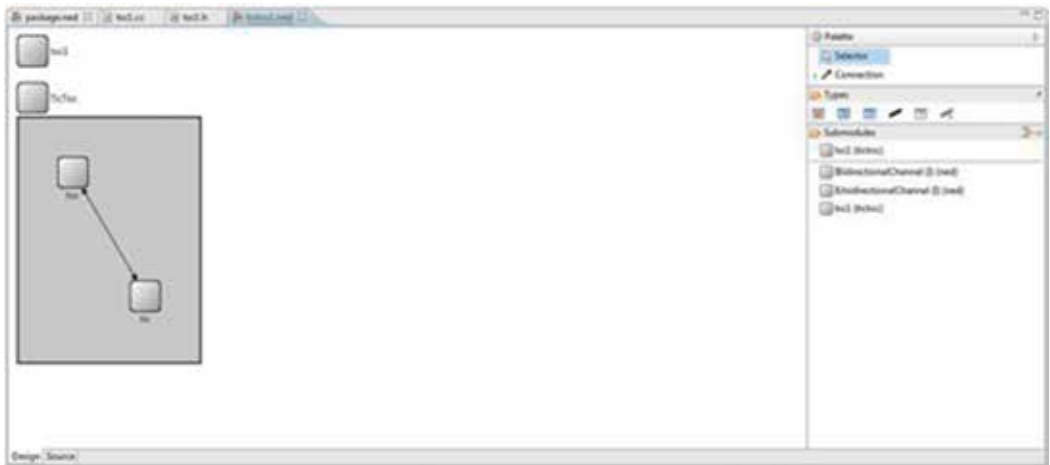
import ned.DelayChannel;
import ned.DatarateChannel;

simple Txcl
{
    gates:
        input in;
        output out;
}

//
// Two instances (tic and toc) of Txcl connected both ways.
// Tic and toc will pass messages to one another.
//
network Tictoc1
{
    submodules:
        tic: Txcl {
            @display("p=205, 70");
        }
        toc: Txcl;
    connections:
        tic.out >DelayChannel > toc.in;
        toc.out >DelayChannel > tic.in;
}
```

Este código, primero define un módulo simple, con un puerto de entrada y otro de salida. Luego, dentro de las llaves donde se define la red, determina las instancias que van a ser usadas en la simulación, una se llama *Tic* y la otra *Toc*. También determina que los enlaces que conectan a *Tic* con *Toc*, son de tipo *DelayChannel*.

Figura 48. Tic-Toc archivo .ned



Al pasar a la pestaña de diseño se va a observar que se crearon unos módulos y una topología como se muestra en la Figura 48.

En la Figura 49, por su parte, se puede observar que aparece el nuevo tipo de módulo que fue definido en el archivo .ned

A continuación se procede a crear la clase .cc, que va a ser la encargada de tener las funcionalidades de la simulación, la que, cómo se explicó, lo que va a realizar es el paso de un mensaje de un módulo a otro. Por esta razón, se debe redefinir el método *handleMessage(cMessage *msg)* y adicionalmente el método *Initialize()*, porque es tanto el que se llama al inicio de la simulación, como donde se define si va a ser el Tic o el Toc, quien inicie la comunicación. En las Figuras 50 y 51 se ilustra el proceso de creación de la clase .cc



Figura 49. Ventana de utilidades del proyecto con el nuevo módulo

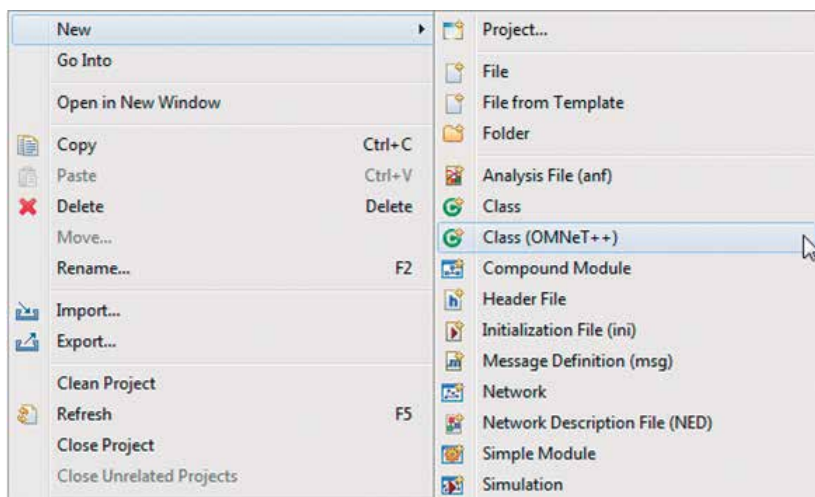


Figura 50. Creación de la clase .cc

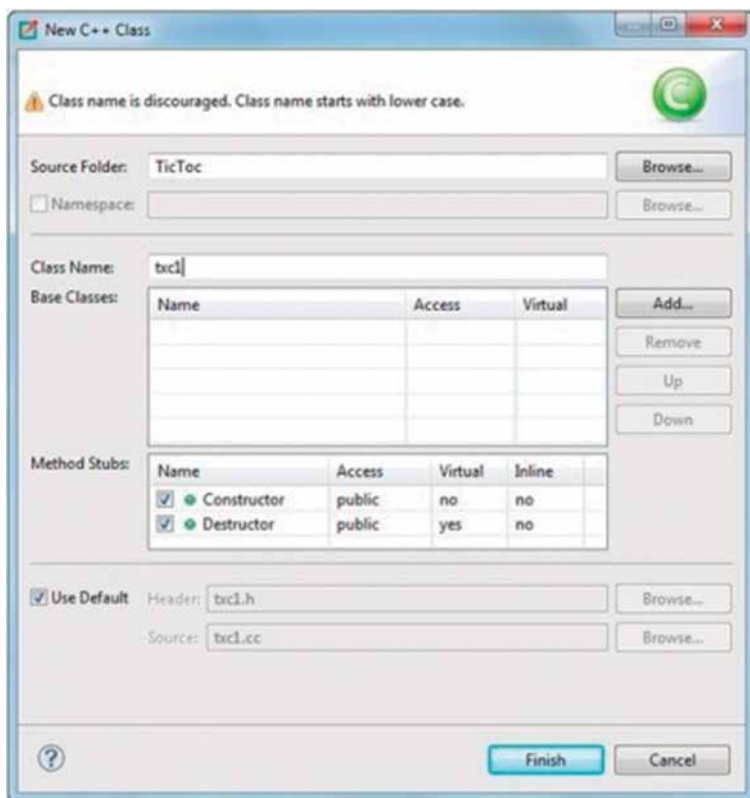


Figura 51. Ventana de creación de la clase .cc

Una vez realizado lo anterior en el Explorador del proyecto se debe observar lo que presenta la Figura 52.

Al hacer clic en la clase .cc (*txc1.cc*) para comenzar a editarla, se dispondrá el siguiente código, en el cual se encuentra la redefinición de los métodos que se mencionaron, el *handleMessage* y el *Initialize*:

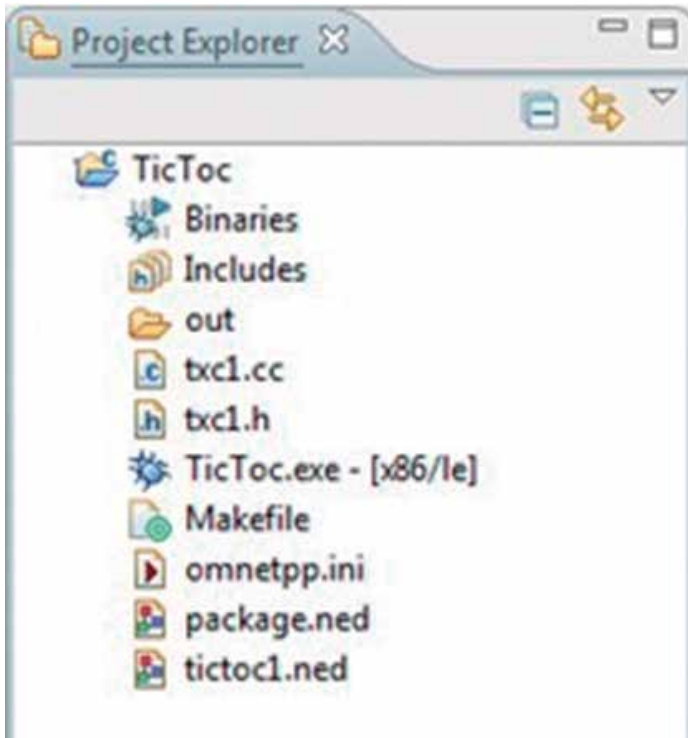


Figura 52. Ventana *Project Explorer* con la clase *.cc*

Hecho esto, solo resta editar un último archivo para poder dar inicio a la simulación, el archivo *.ini*, el archivo de configuración de la simulación, que es usado por la herramienta para realizar la compilación y ejecución de la simulación. En las Figuras 53 y 54 se observa la creación de dicho archivo.

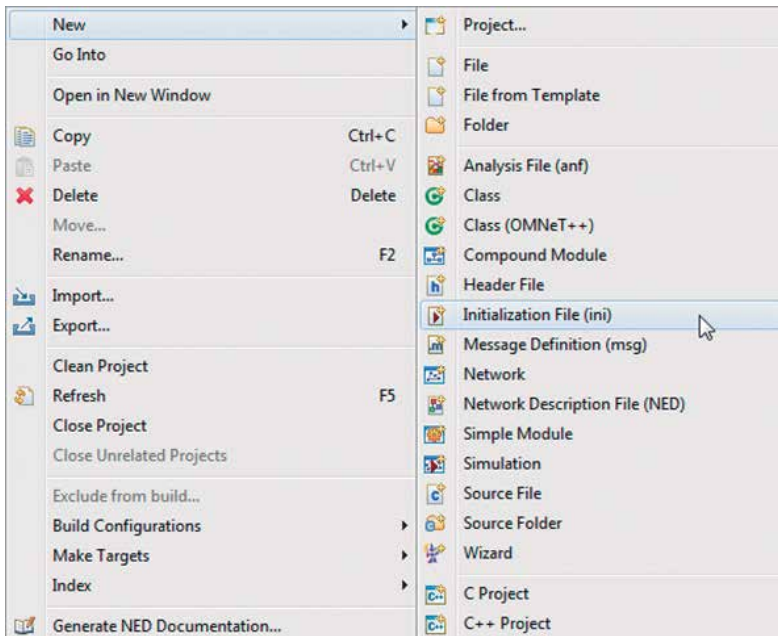


Figura 53. Creación del archivo *.ini*

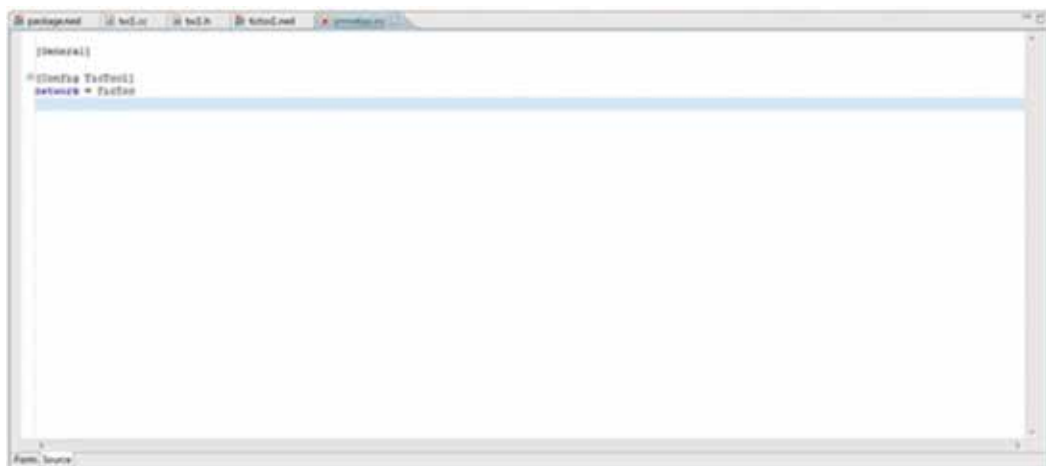


Figura 54. Ventana de la creación del archivo *.ini*

Una vez se abre el archivo *.ini* se observa una interfaz como la que muestra la Figura 55. Aquí, al igual que en el archivo *.ned*, existen dos pestañas, una denominada *Form* y otra denominada *Source*; en esta última se va a disponer el código de la red a simular.

Hecho esto, ubicándose en el proyecto, se presiona el clic derecho y se selecciona la opción de *Run as*, como se muestra en la Figura 56.

Figura 55. Código fuente del archivo *.ini*



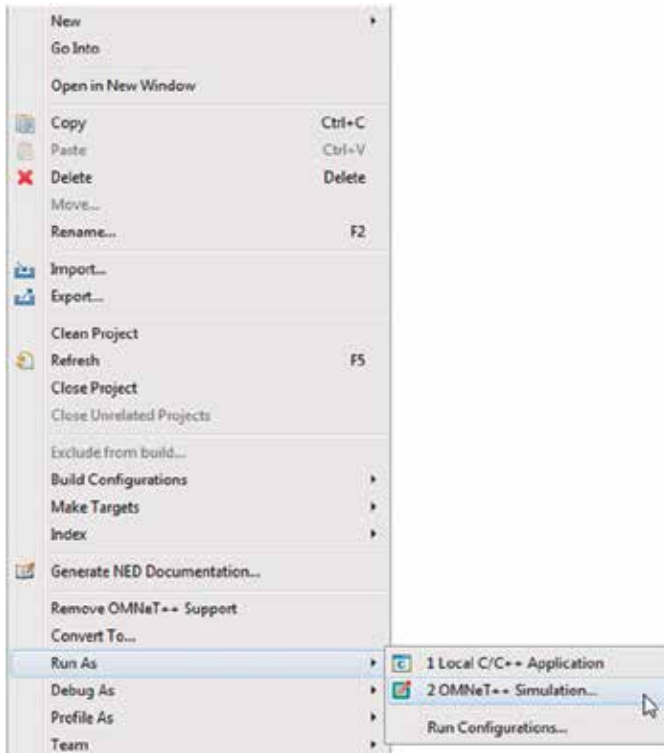


Figura 56. Inicio de la simulación

Cuando se realiza esta acción un mensaje de advertencia se despliega en la pantalla, indicando que se ha creado una nueva configuración, como muestra la Figura 57.

Al aceptar el mensaje aparece otra ventana emergente que indica la selección del archivo .ini, como muestra la Figura 58.

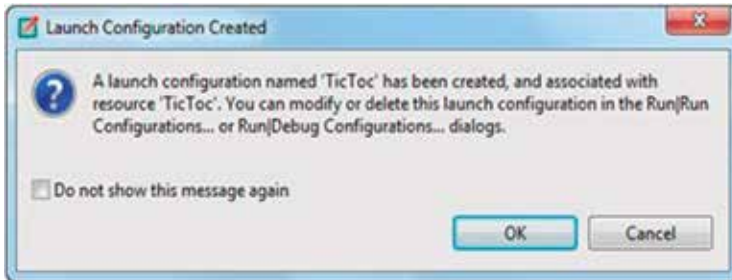


Figura 57. Mensaje de advertencia



Figura 58. Ventana - Set up an Inifile Configuration

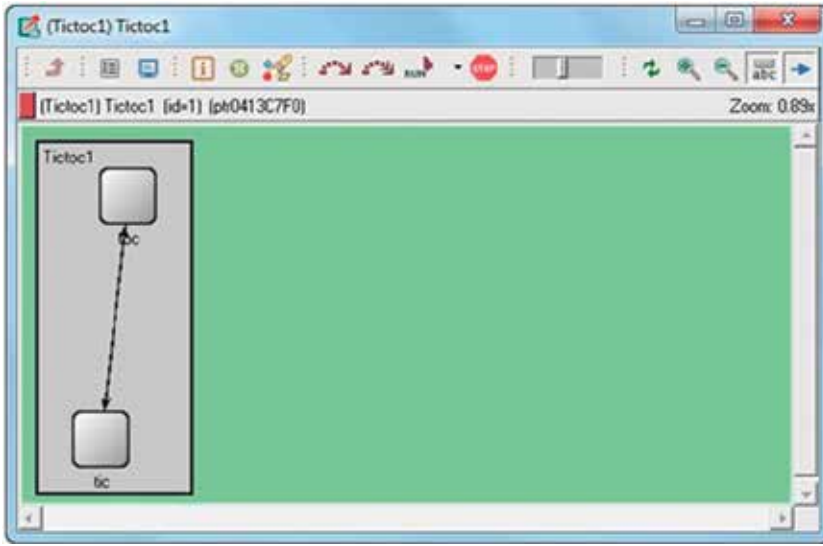


Figura 59. Ventana - Interfaz gráfica de OMNeT++ (1 de 3)

Finalizada la acción anterior se desplegará en pantalla una interfaz como la que muestra la Figura 59.

Para iniciar la simulación se presiona el botón de *Run* que se encuentra la parte central del menú superior de la Figura 59. Al hacerlo se observa lo que muestra la Figura 60 y 61, en la que el recuadro rojo indica que módulo está actuando y el círculo rojo que se mueve a través del enlace, hace referencia al mensaje que está siendo transmitido.

Además de la ventana de interfaz gráfica de OMNeT++, el simulador abre otra ventana que arroja toda la información de la simulación, como se puede observar en la Figura 62.

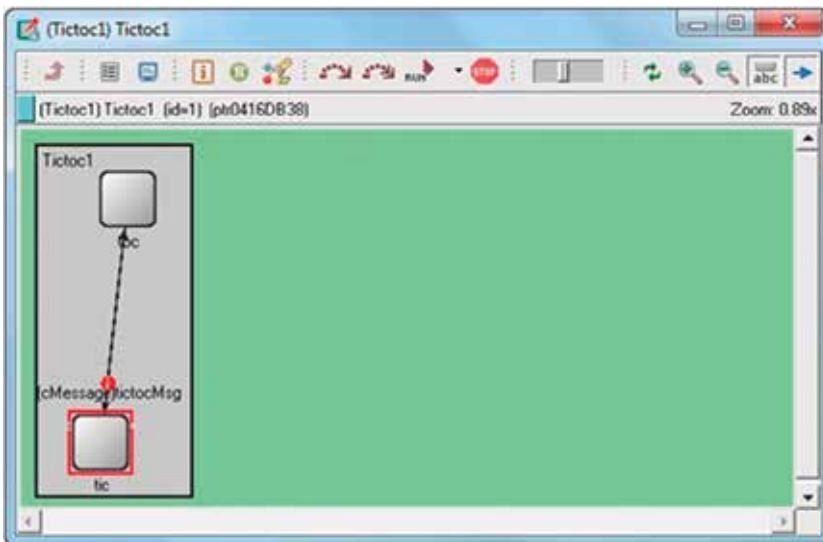


Figura 60. Ventana - Interfaz gráfica de OMNeT++ (2 de 3)

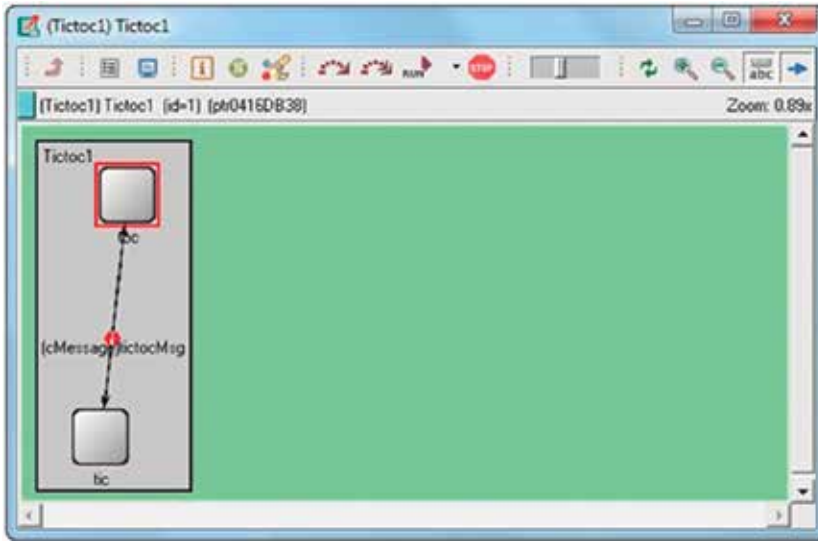


Figura 61. Ventana - Interfaz gráfica de OMNET++ (3 de 3)

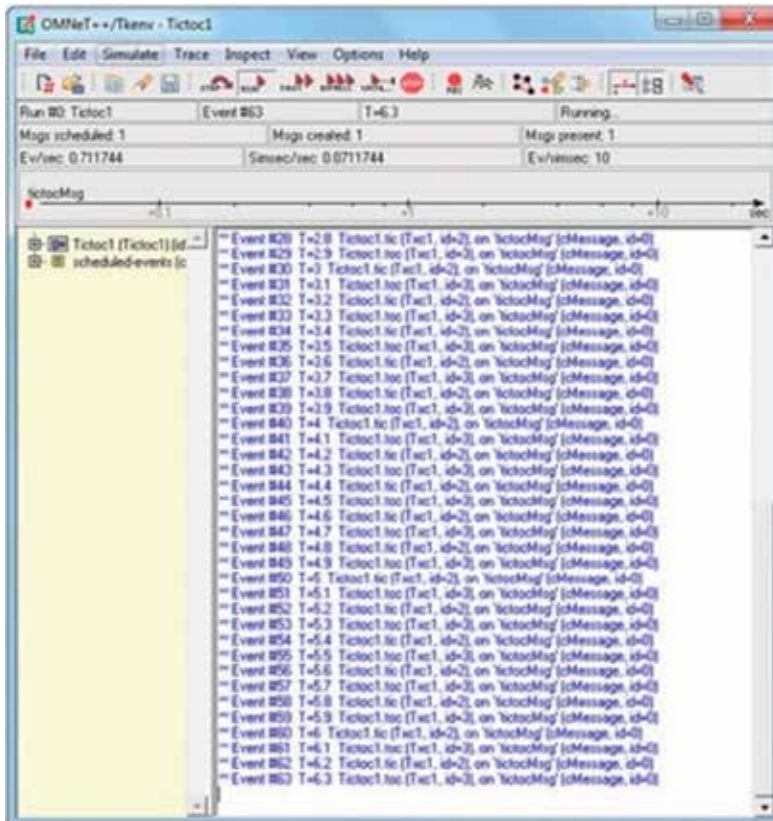


Figura 62. Ventana de información del proceso de simulación del TicToc

Para terminar la simulación se presiona el botón *Stop*, que se encuentra en la parte central del menú superior de la ventana de la interfaz (ver Figura 61).