

Guia para ejecutar Omnet++ en Windows:

- 1) Instalar **WSL (Windows Subsystem for Linux)**, **Ubuntu 20.04** (microsoft store) como distribucion y la terminal "**Windows Terminal**" siguiendo las instrucciones en:
<https://docs.microsoft.com/en-us/windows/wsl/install-win10>
- 2) Instalar **Docker** para Windows desde: <https://www.docker.com/products/docker-desktop>
- 3) Instalar **Xming**, que sera el XServer que vamos a usar para ejecutar la ide de Omnet desde Docker:
<https://sourceforge.net/projects/xming/>
- 4) Ejecutar XLaucher, para configurar Xming con la configuración¹:
 - **Select display Settings** Multiple Windows, Display Number: 0
 - **Select to start Xming** Start no client
 - **Specify parameter settings** Clipboard, No Access Control

- 5) Con Xming ya ejecutándose, ir a Windows Terminal y ejecutar(*)

```
PS C:\Users\Pablo> ipconfig
```

Va a devolver varios adaptadores y sus ip's, como ejemplo pongo la salida del adaptador que nos interesa, el que dice vEthernet (WSL).

Adaptador de Ethernet **vEthernet (WSL)** :

```
Sufijo DNS específico para la conexión. . . :  
Vínculo: dirección IPv6 local. . . : fe80::9976:4d2f:384f:f95b%19  
Dirección IPv4. . . . . : 172.18.208.1  
Máscara de subred . . . . . : 255.255.240.0  
Puerta de enlace predeterminada . . . . . :
```

- 6) Ahora, teniendo la ip que nos ha devuelto ipconfig, la utilizaremos para ejecutar en windows terminal el comando:

```
PS C:\Users\Pablo> set-variable -name DISPLAY -value 172.18.208.1:0.0
```

Donde lo que está en rojo es la ip que nos devolvió ipconfig para el adaptador vEthernet. Pueden verificar que la variable haya sido seteada con el comando `echo $DISPLAY`

- 7) Luego ejecutamos en la terminal de Windows:

```
PS C:\Users\Pablo> docker run --rm -ti -e DISPLAY=$DISPLAY -v  
/tmp/.X11-unix:/tmp/.X11-unix mmodenesi/omnetpy bash
```

y ya estaríamos en la terminal de Linux dentro de Windows.

- 8) Finalmente podemos ejecutar `$omnetpp` para arrancarlo.

- Nota: Si al llegar al paso 8 te sale un error como el siguiente, deberás reiniciar el sistema operativo.

```
eclipse.buildId=6.0.0.200507-e16b5b0502
```

```
java.version=11.0.4
```

```
java.vendor=AdoptOpenJDK
```

```
BootLoader constants: OS=linux, ARCH=x86_64, WS=gtk, NL=en_US
```

```
Command-line arguments: -os linux -ws gtk -arch x86_64
```

```
!ENTRY org.eclipse.osgi 4 0 2021-03-11 23:52:55.160
```

```
!MESSAGE Application error
```

```
!STACK 1
```

```
org.eclipse.swt.SWTError: No more handles [gtk_init_check() failed]
```

¹ Tomado de: <https://github.com/handflucht/guiAppInDockerOnWindows>

```
at org.eclipse.swt.SWT.error(SWT.java:4749)
at org.eclipse.swt.widgets.Display.createDisplay(Display.java:1130)
at org.eclipse.swt.widgets.Display.create(Display.java:1070)
at org.eclipse.swt.graphics.Device.<init>(Device.java:175)
at org.eclipse.swt.widgets.Display.<init>(Display.java:636)
at org.eclipse.swt.widgets.Display.<init>(Display.java:627)
at org.eclipse.ui.internal.Workbench.createDisplay(Workbench.java:776)
at org.eclipse.ui.PlatformUI.createDisplay(PlatformUI.java:166)
at
org.eclipse.ui.internal.ide.application.IDEApplication.createDisplay(IDEApplica
tion.java:180)
at
org.eclipse.ui.internal.ide.application.IDEApplication.start(IDEApplication.
java:123)
(....)
```