







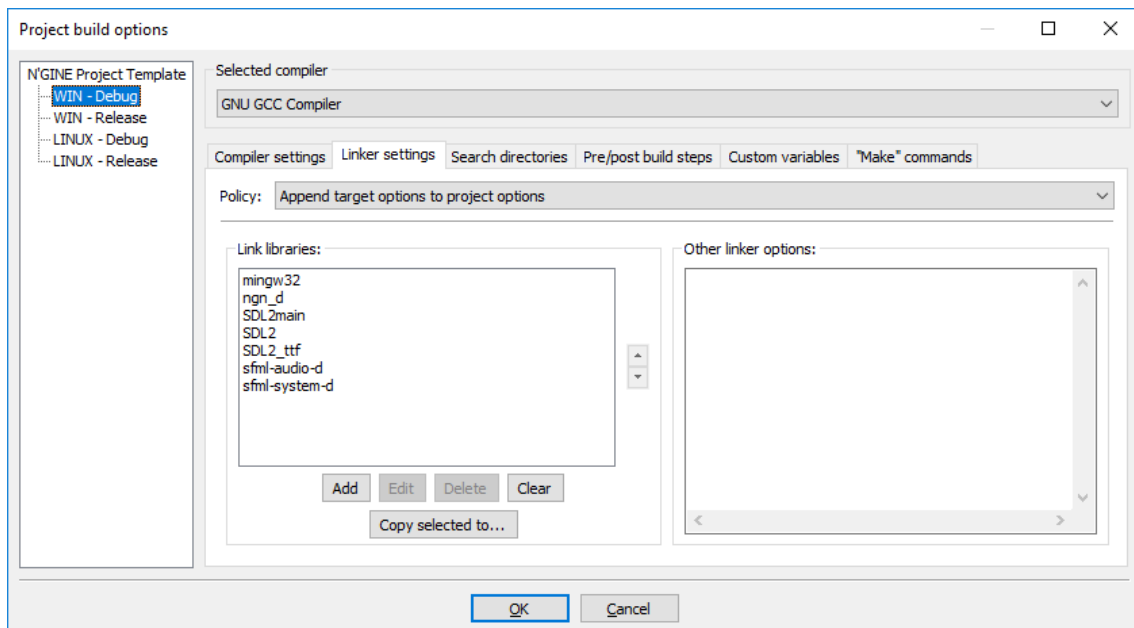


Copiar en cada carpeta los directorios “include”, “lib” y opcionalmente “bin” correspondientes a cada librería.

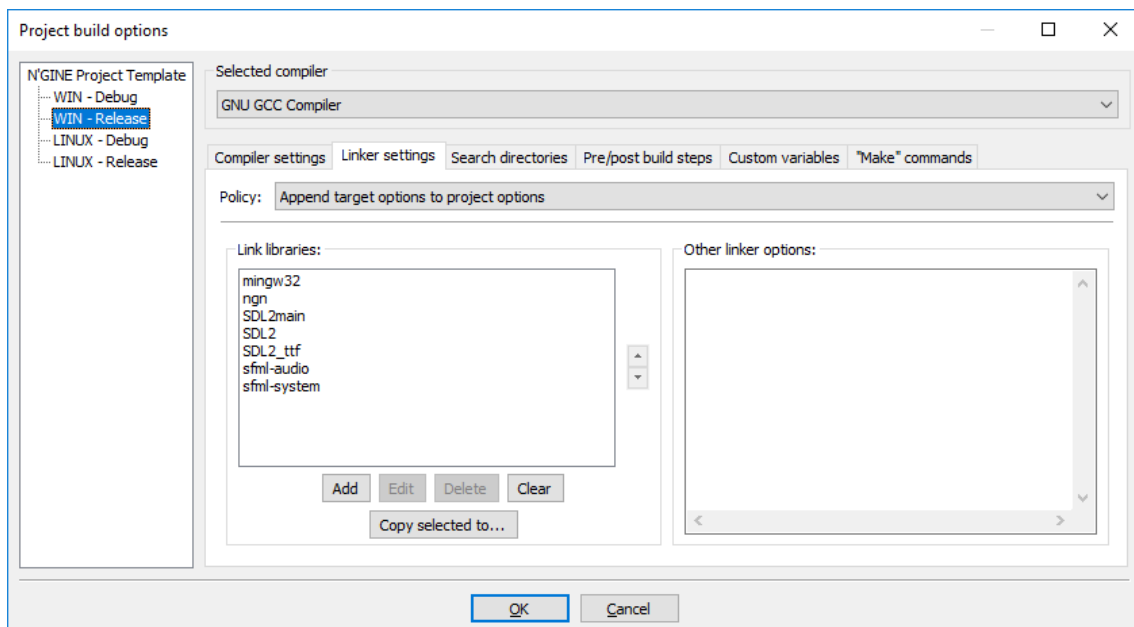
| | | |
|---|---------|----------------------|
| Inicio > C++ > Librerías > CodeBlocks > Libs > ngn > | | |
| <input type="checkbox"/> | Nombre | Fecha de modifica... |
|  | include | 15/06/2017 13:22 |
|  | lib | 15/06/2017 13:22 |
| Inicio > C++ > Librerías > CodeBlocks > Libs > sdl2 > | | |
| <input type="checkbox"/> | Nombre | Fecha de modifica... |
|  | bin | 04/06/2017 16:28 |
|  | include | 04/06/2017 16:28 |
|  | lib | 04/06/2017 16:28 |
| Inicio > C++ > Librerías > CodeBlocks > Libs > sfml > | | |
| <input type="checkbox"/> | Nombre | Fecha de modifica... |
|  | bin | 04/06/2017 16:28 |
|  | include | 04/06/2017 16:28 |
|  | lib | 04/06/2017 16:28 |

Configuración del proyecto - CODE::BLOCKS

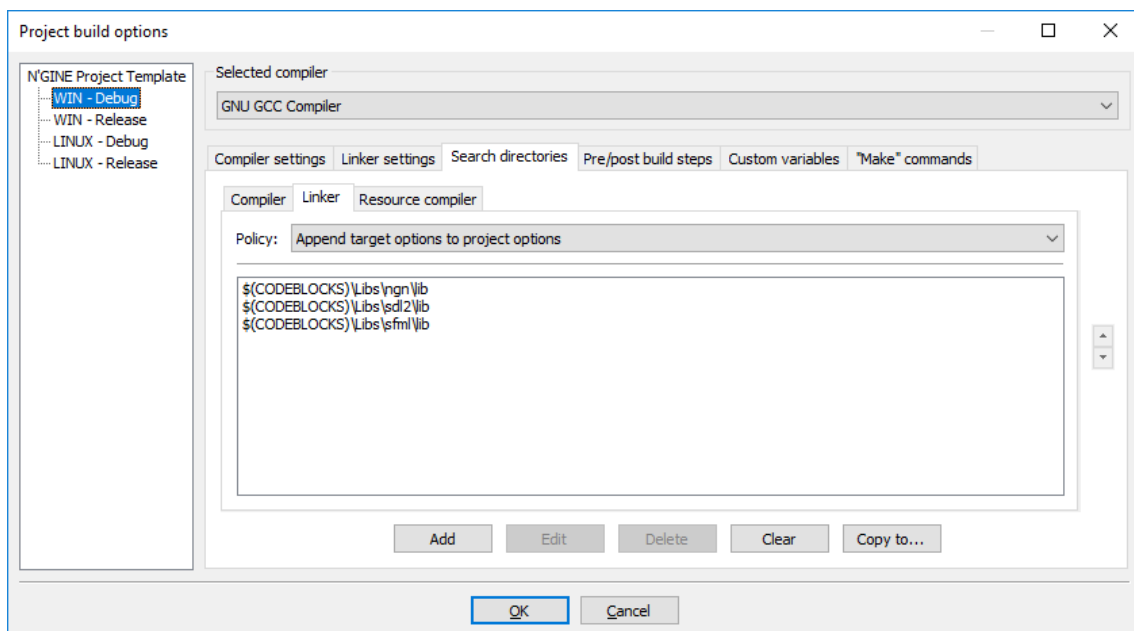
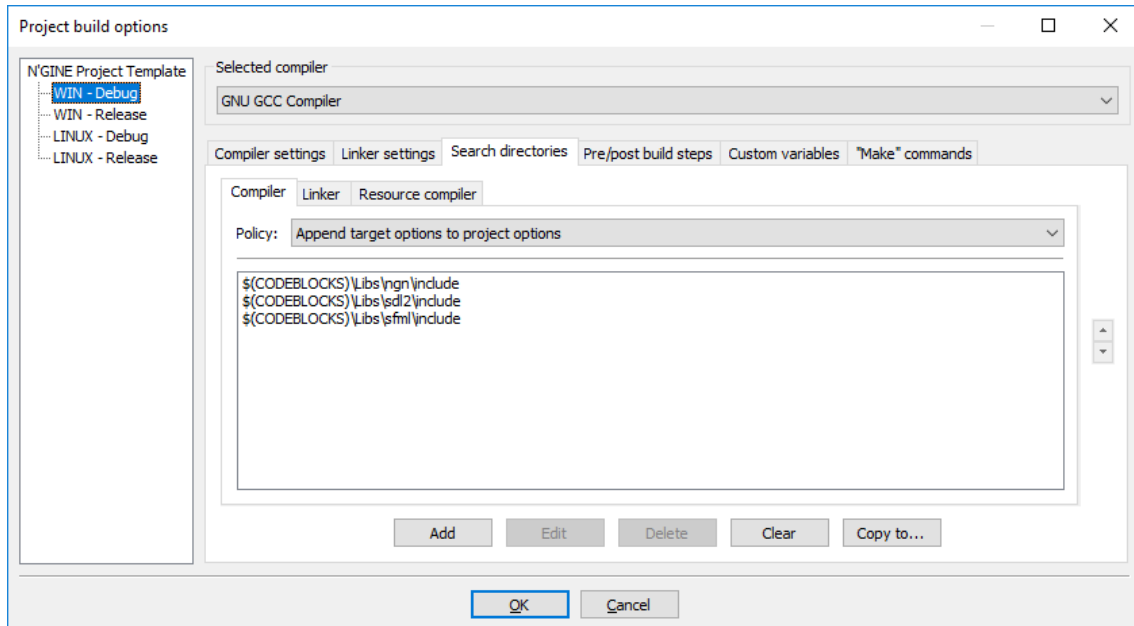
En “Project build options” configuraremos en la sección WIN-DEBUG:



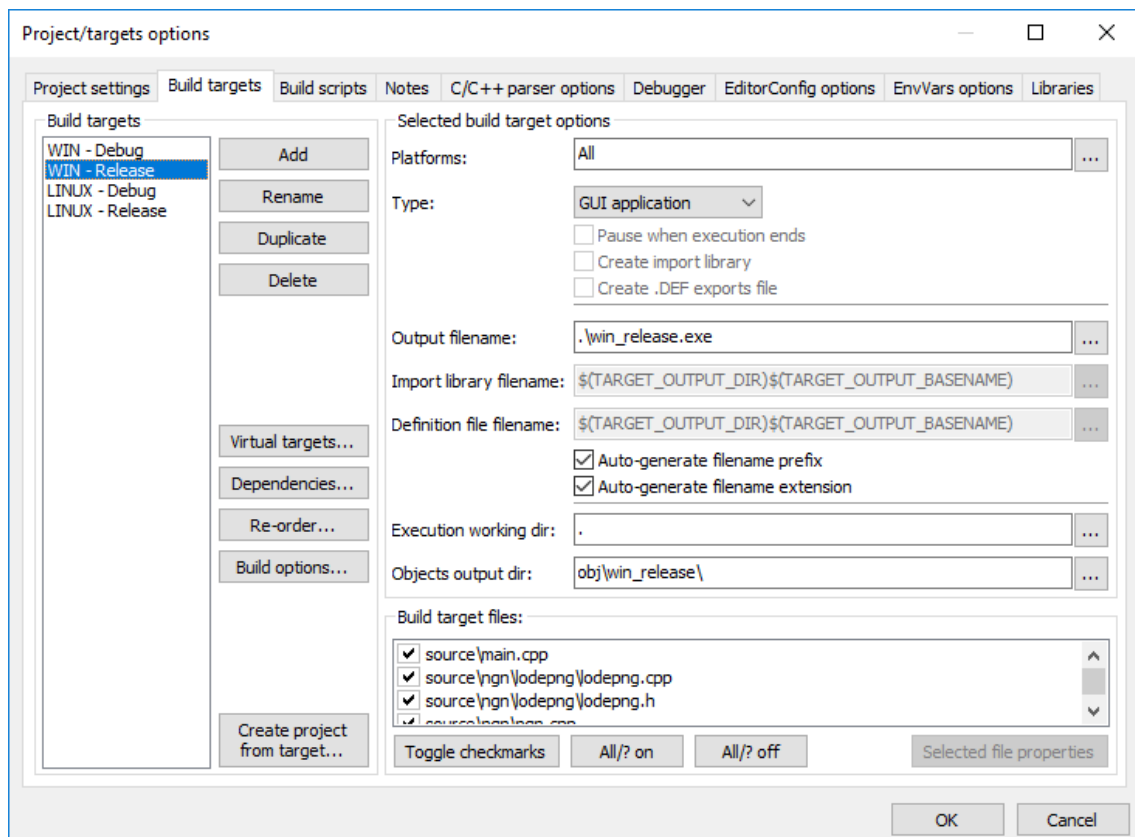
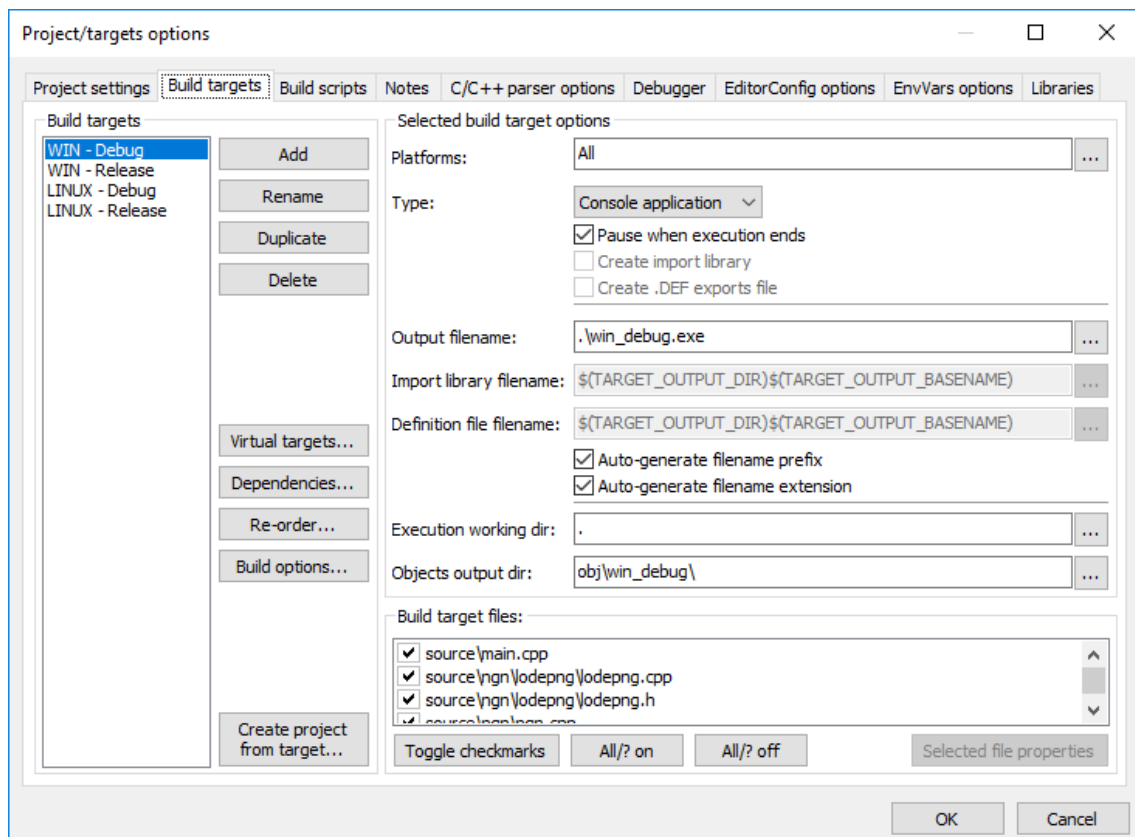
En “Project build options” configuraremos en la sección WIN-RELEASE:



En “Project build options” configuraremos tanto en la sección WIN-DEBUG como en WIN-RELEASE:



En “Project/target options” configuraremos:



Instalación de las librerías - Linux

Desde el terminal, buscaremos las librerías SDL2 con el siguiente comando:

```
sudo apt-cache search libsdl2
```

A continuación, instalaremos las librerías con este otro:

```
sudo apt-get update  
sudo apt-get install libsdl2-dev
```

Buscaremos las librerías SDL2-TTF con el siguiente comando:

```
sudo apt-cache search libsdl2-ttf
```

A continuación, instalaremos las librerías con este otro:

```
sudo apt-get update  
sudo apt-get install libsdl2-ttf-dev
```

Buscaremos las librerías SFML con el siguiente comando:

```
sudo apt-cache search sfml
```

A continuación, instalaremos las librerías con este otro:

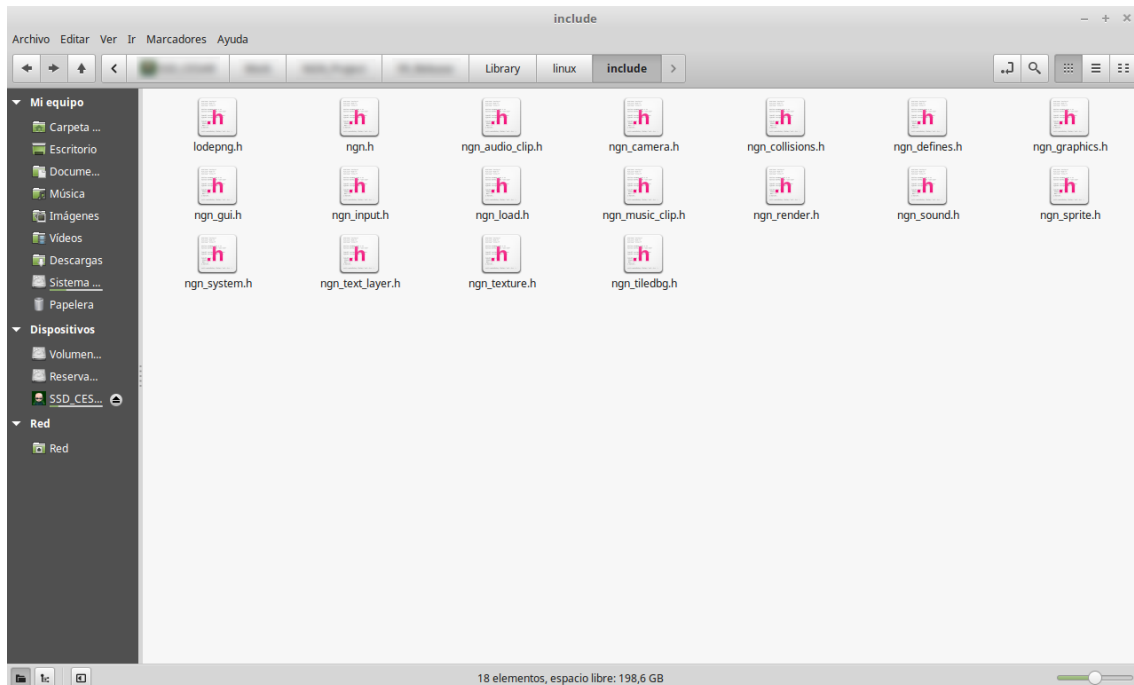
```
sudo apt-get update  
sudo apt-get install libsFML-dev
```

Es posible que falten algunas dependencias del compilador de C++. En ese caso las instalaremos con los siguientes comandos:

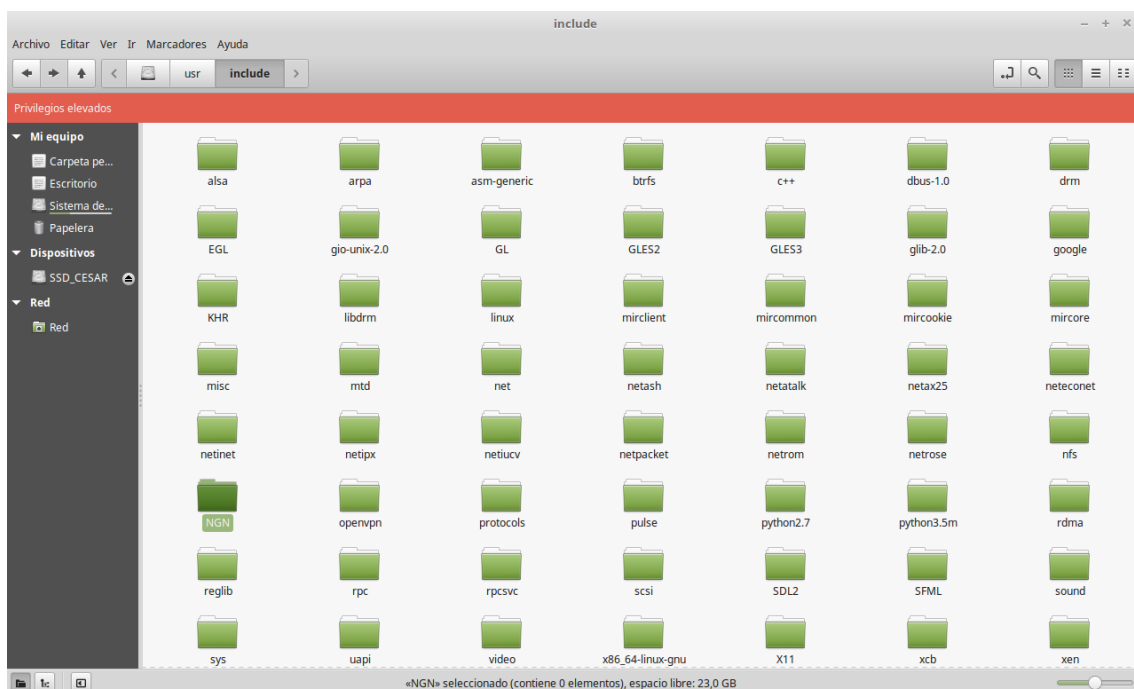
```
sudo apt-get update  
sudo apt-get install build-essential
```

Para la instalación de la librería N'gine, seguiremos los siguientes pasos:

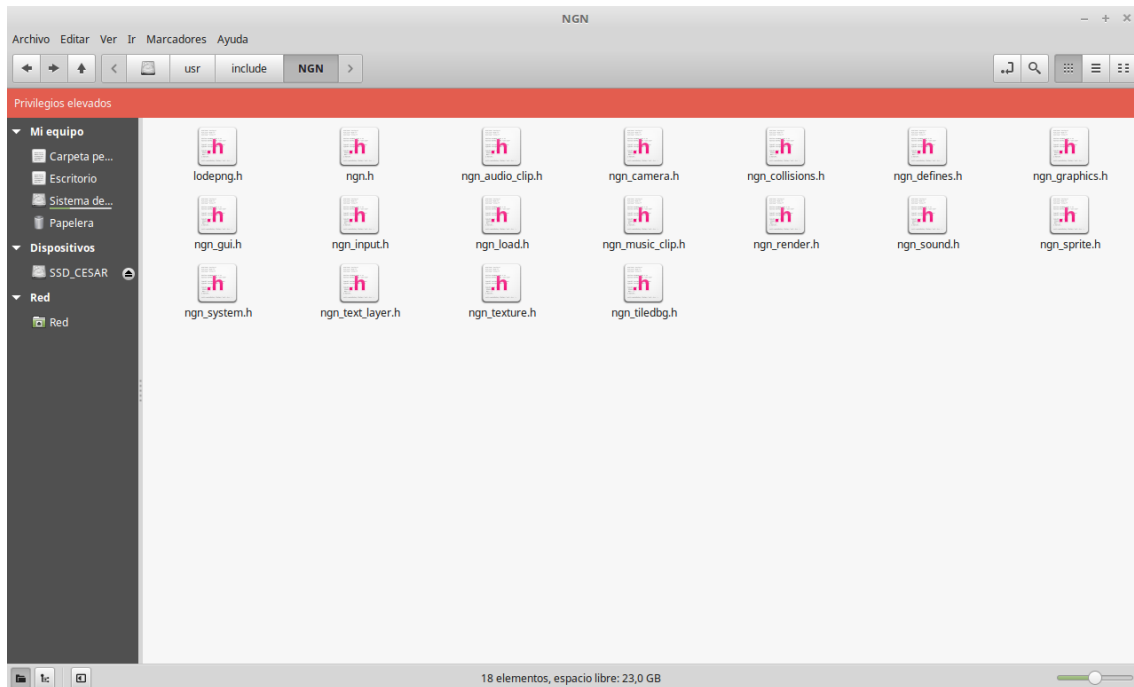
Copiaremos los archivos INCLUDE (.h) contenidos en la librería



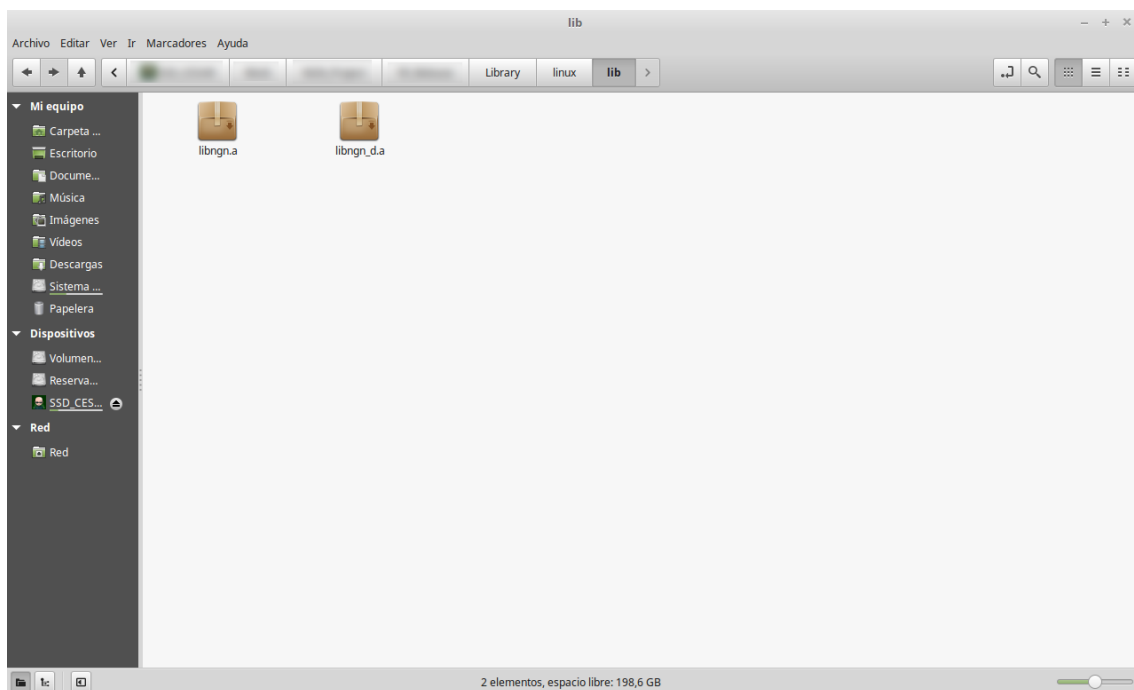
Nos colocaremos en la carpeta del sistema “usr/include” con privilegios elevados y crearemos una carpeta con el nombre “NGN” (en mayusculas).



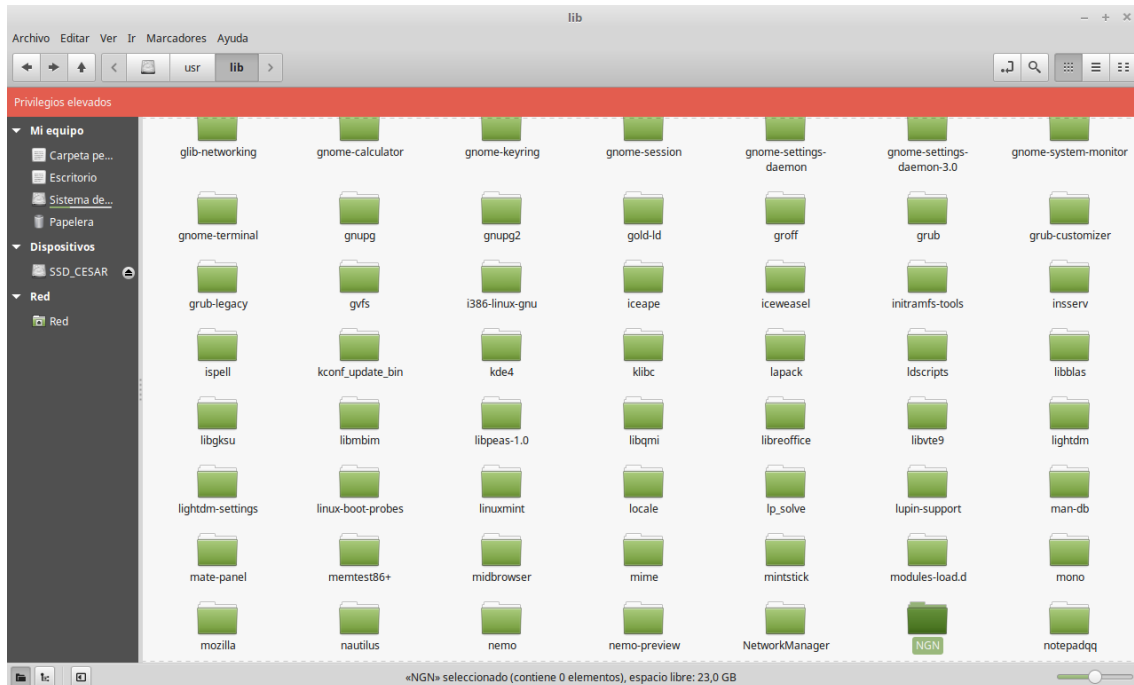
A continuación, pegaremos los archivos .h en su interior.



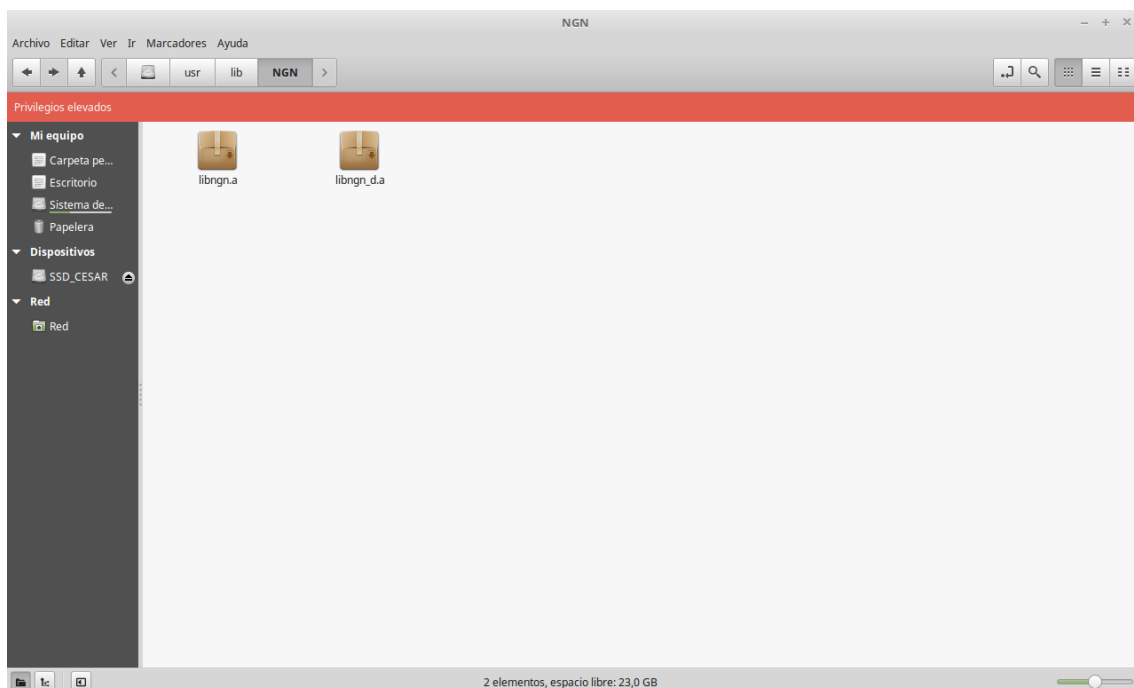
Ahora copiaremos los archivos binarios de la librería (.a) de la carpeta lib.



Nos colocaremos en la carpeta del sistema “usr/lib” con privilegios elevados y crearemos otra carpeta con el nombre “NGN” (en mayusculas).

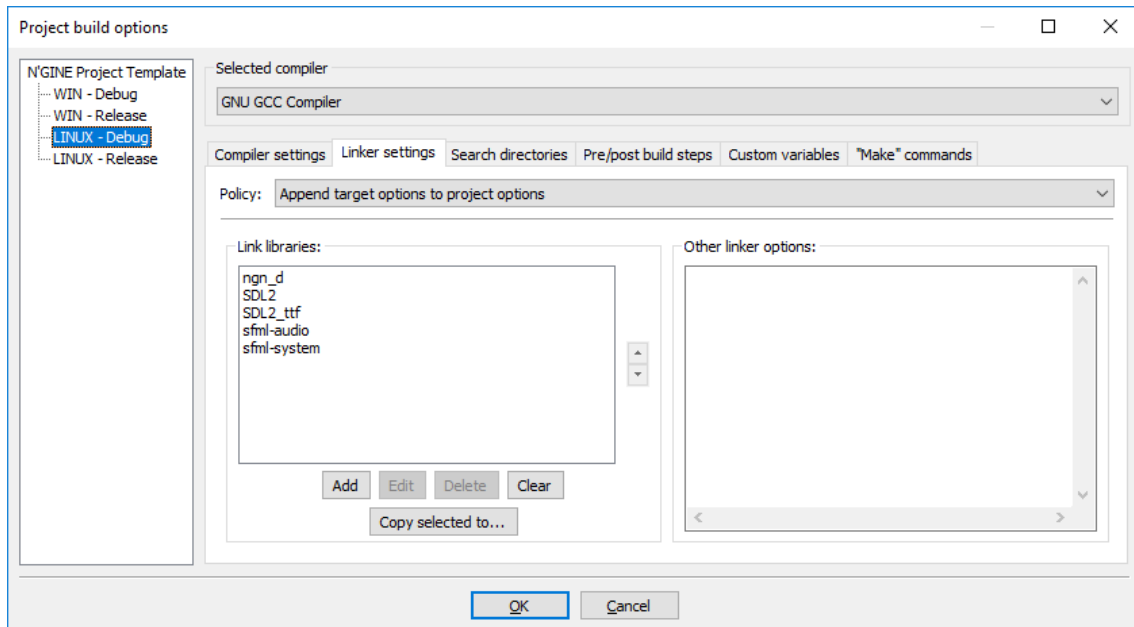


Para terminar, pegaremos los archivos .a en el interior de esta carpeta.

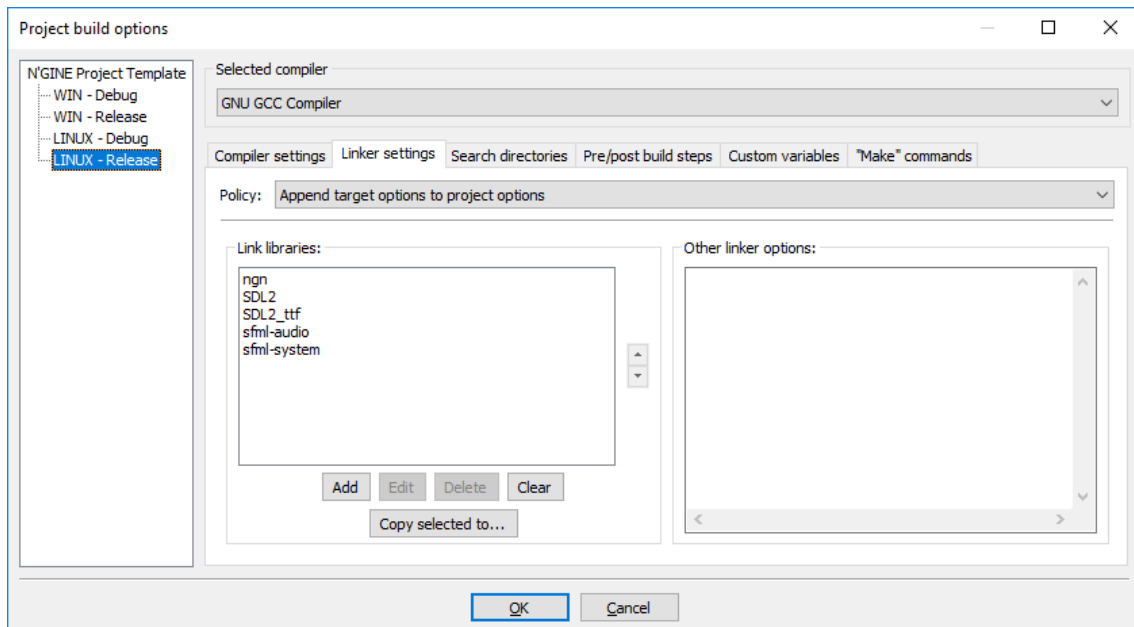


Configuración del proyecto - CODE::BLOCKS

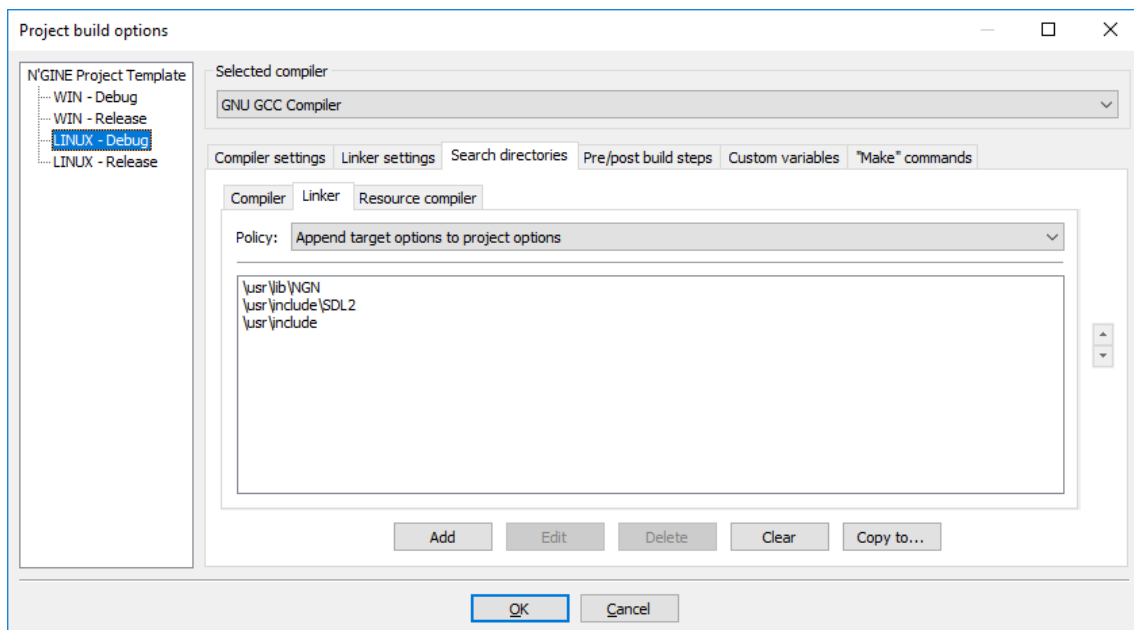
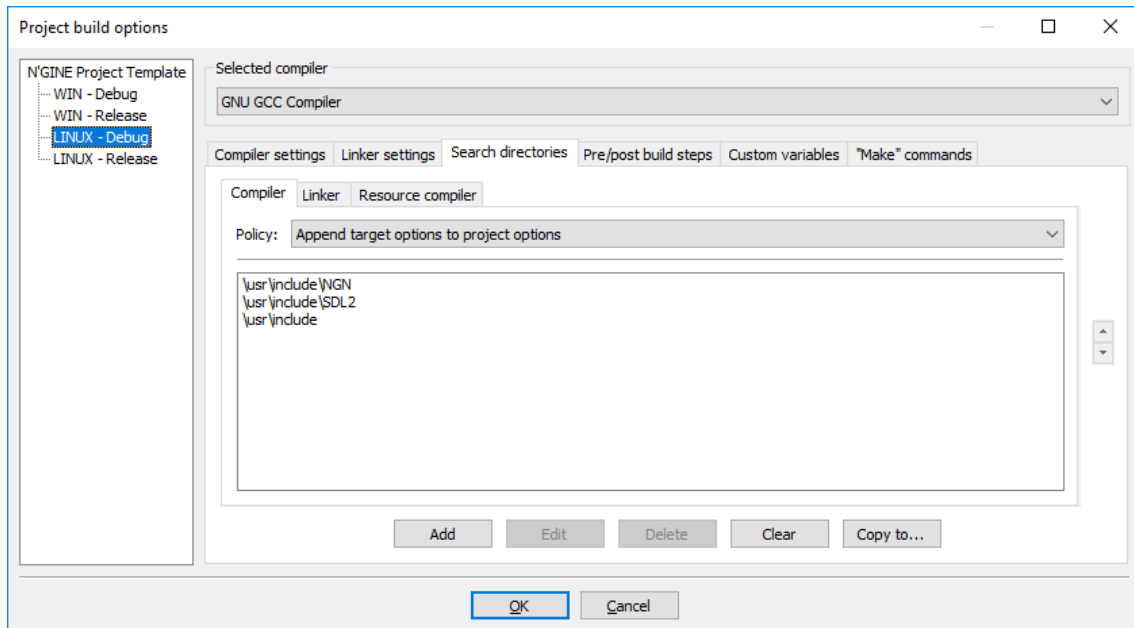
En “Project build options” configuraremos en la sección LINUX-DEBUG:



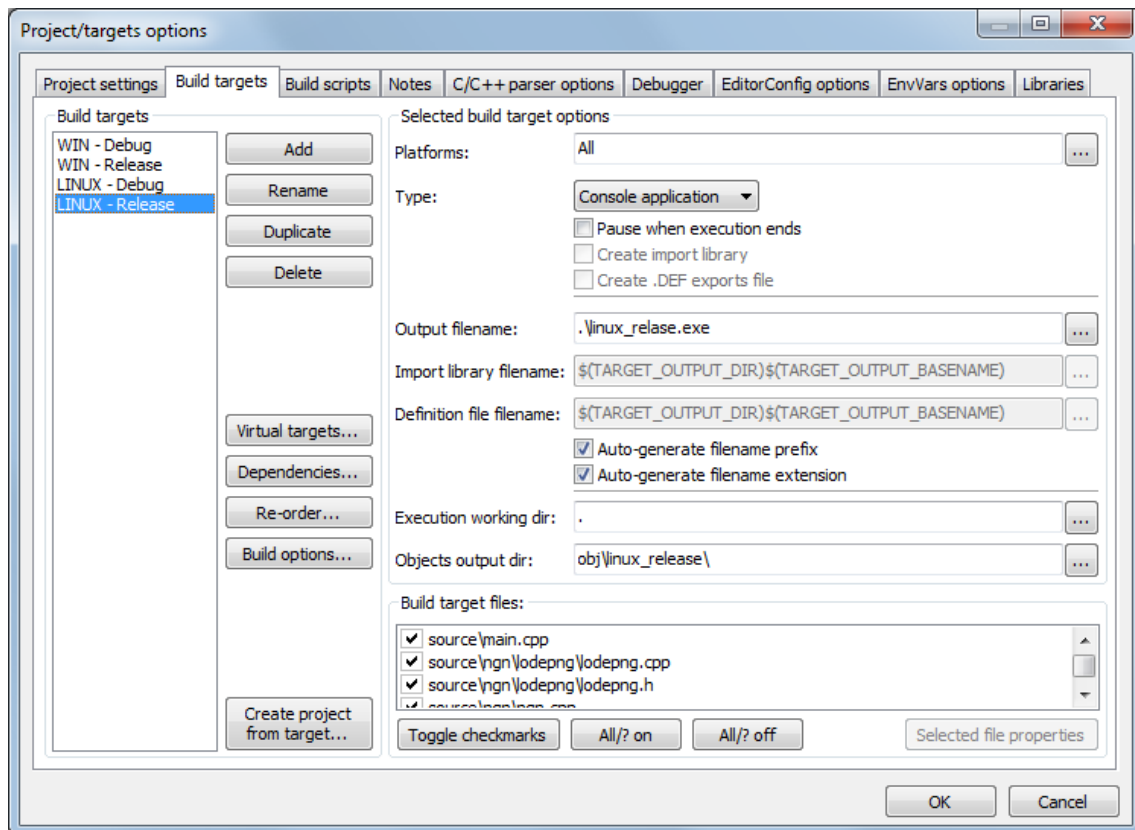
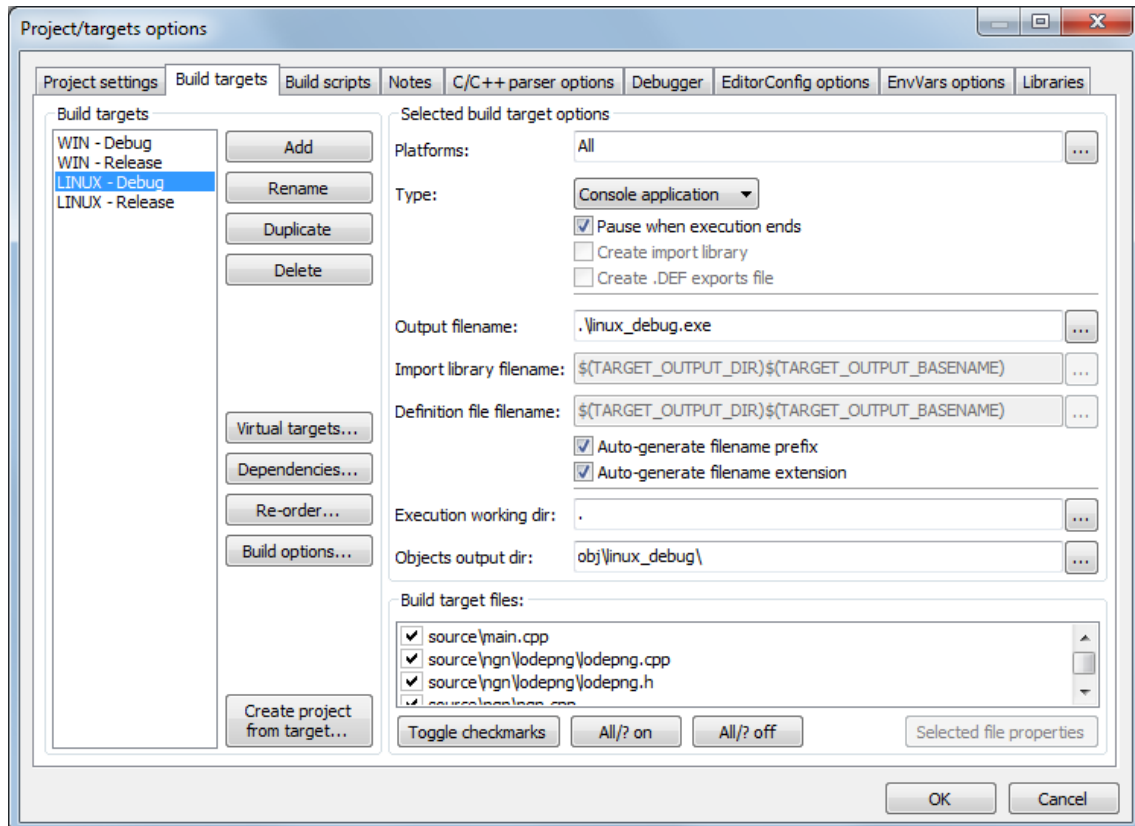
En “Project build options” configuraremos en la sección LINUX-DEBUG:



En “Project build options” configuraremos tanto en la sección LINUX-DEBUG como en LINUX-RELEASE:



En “Project/target options” configuraremos:



Opcionalmente, la librería incluye un “template” con todas estas opciones ya configuradas, tanto para Windows como para Linux. En caso de usar este template, solo será necesario realizar el paso de instalación de las librerías.