

N'GINE – Configuración de CODE::BLOCKS

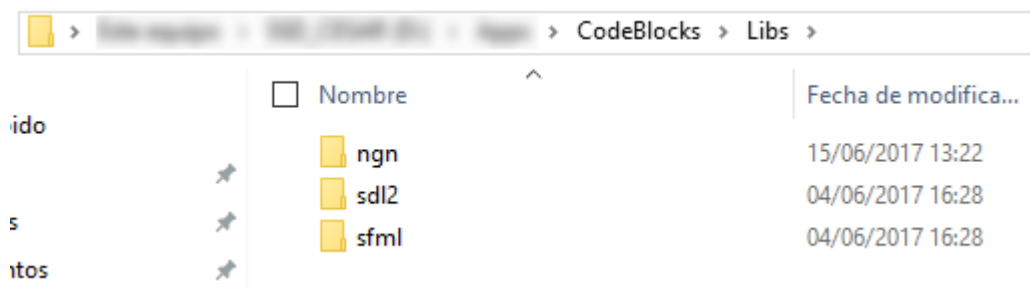
Instalación de las librerías – Windows

Empezaremos descargando las librerías adicionales desde sus páginas oficiales:

SDL2: <https://www.libsdl.org/download-2.0.php>

SFML: <https://www.sfml-dev.org/>

A continuación, crearemos una carpeta llamada “Libs” dentro de la carpeta donde tengamos instalado CODE::BLOCKS. En el interior de esta carpeta, crearemos las carpetas “ngn”, “sdl2” y “sfml”.

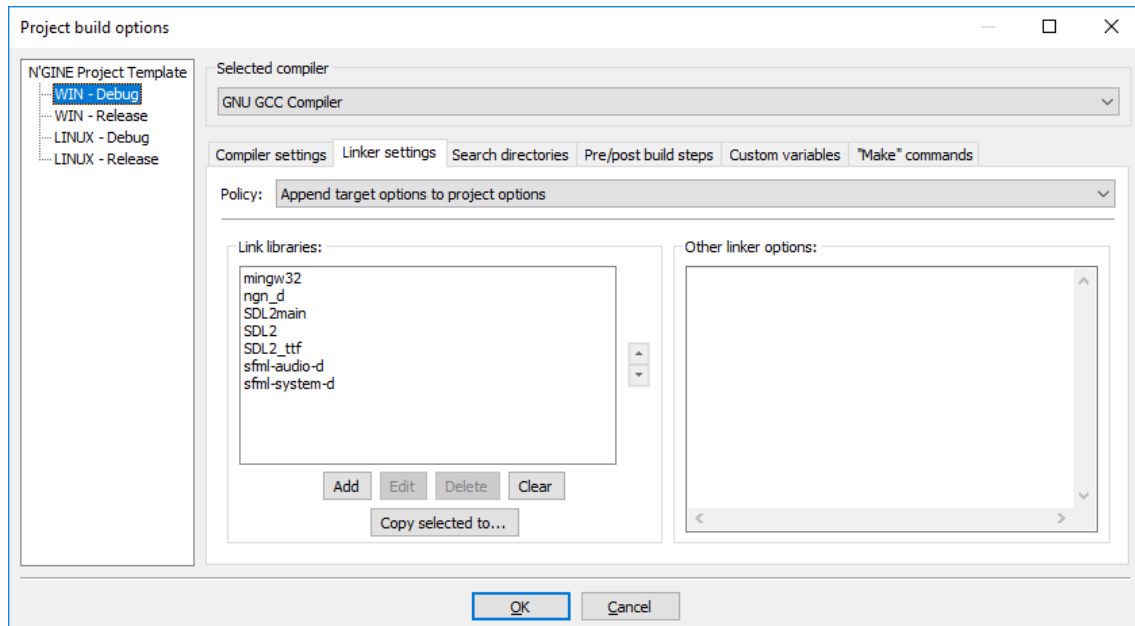


Copiaremos en cada carpeta los directorios “include”, “lib” y opcionalmente “bin” correspondientes a cada librería.

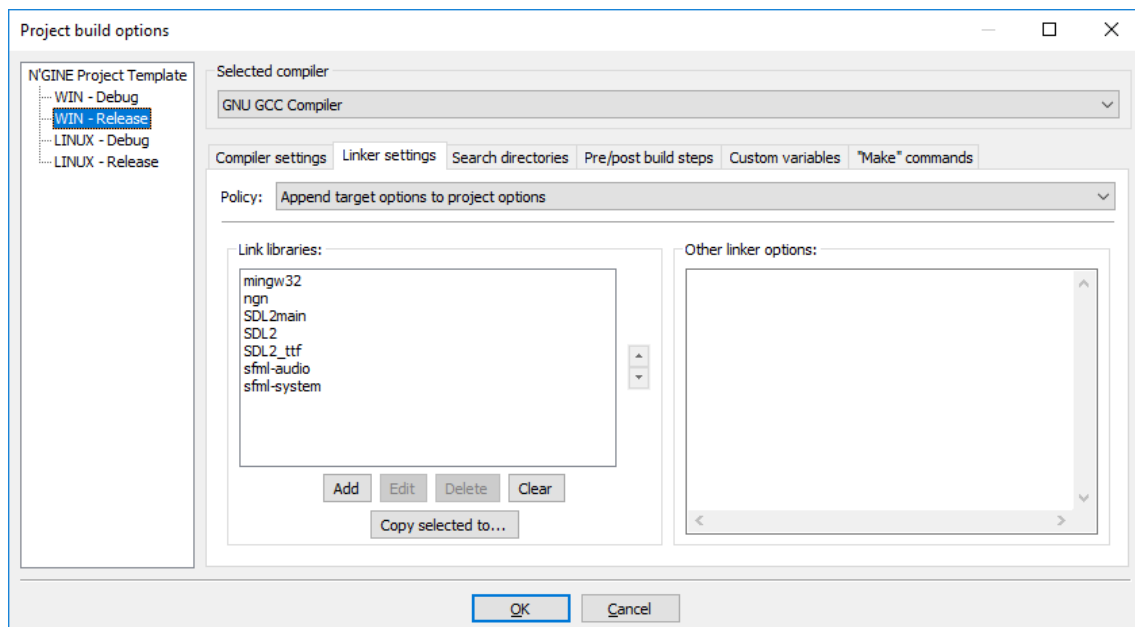
CodeBlocks > Libs > ngn >		
<input type="checkbox"/>	Nombre	Fecha de modifica...
	include	15/06/2017 13:22
	lib	15/06/2017 13:22
CodeBlocks > Libs > sdl2 >		
<input type="checkbox"/>	Nombre	Fecha de modifica...
	bin	04/06/2017 16:28
	include	04/06/2017 16:28
	lib	04/06/2017 16:28
CodeBlocks > Libs >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

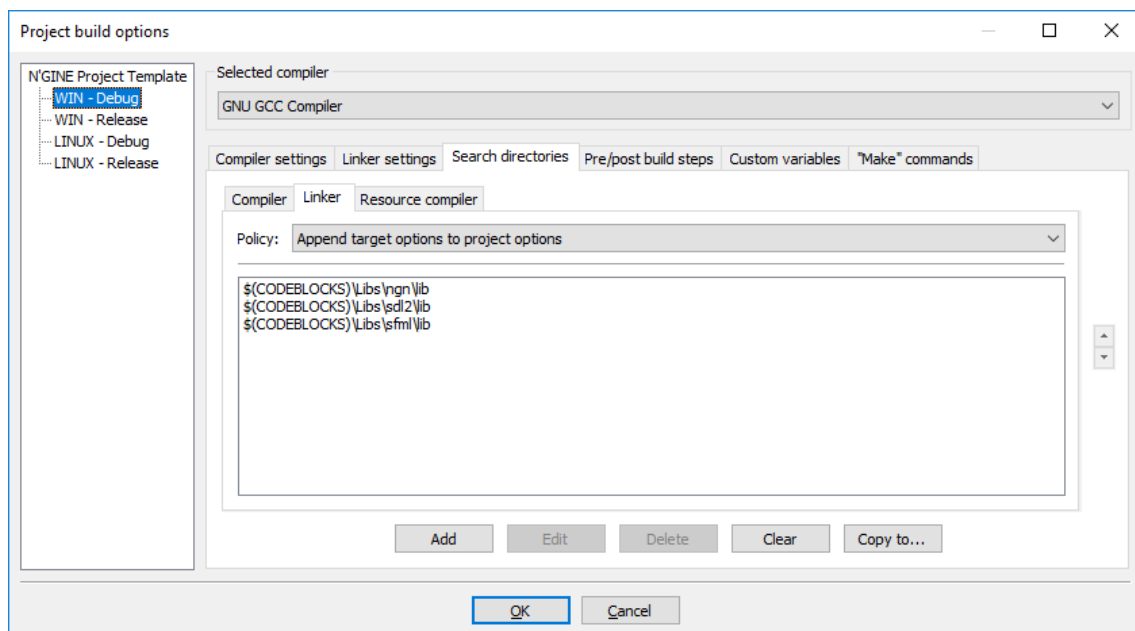
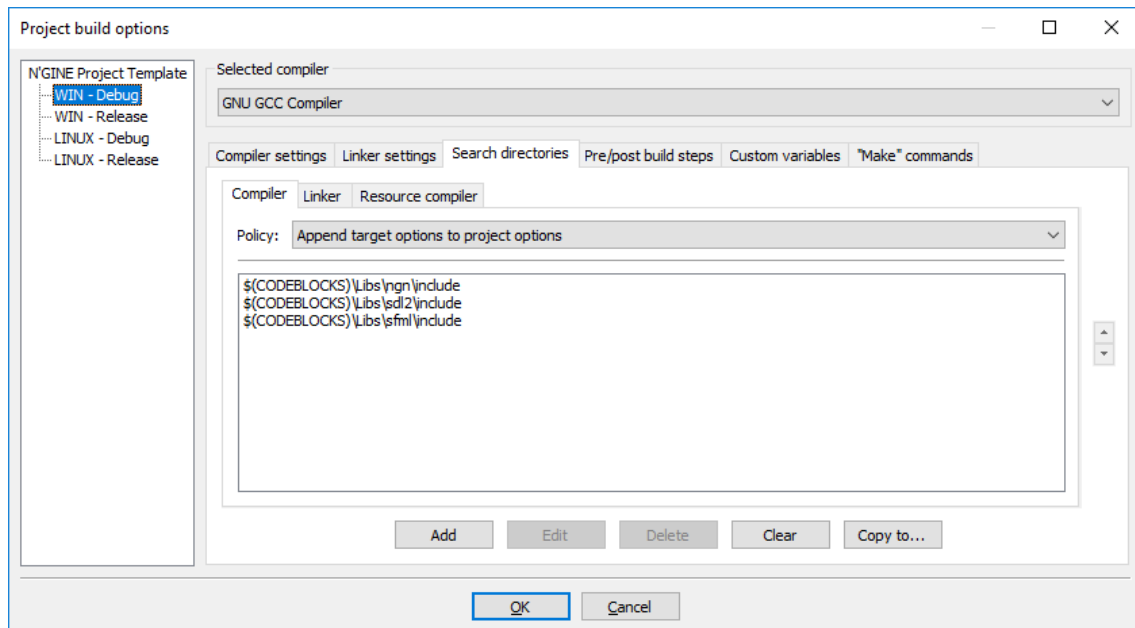
En “Project build options”, en la sección “WIN-DEBUG”, configuraremos los siguientes parámetros:



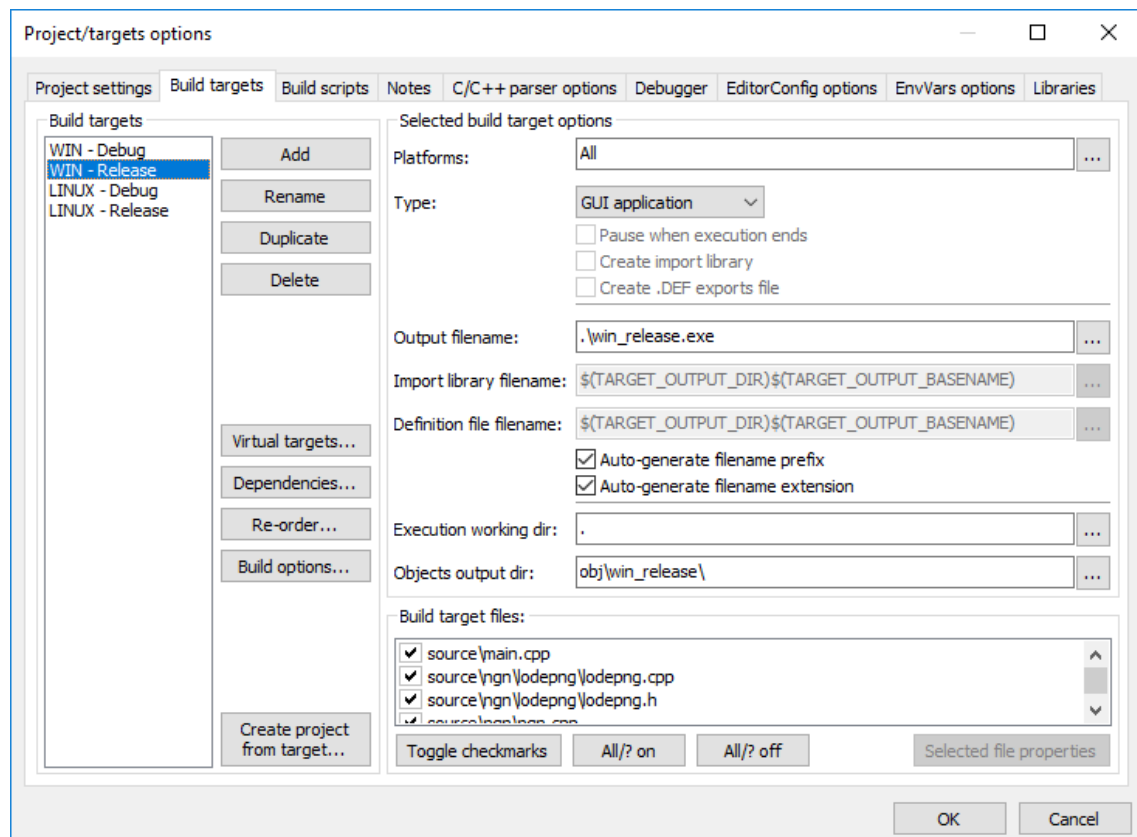
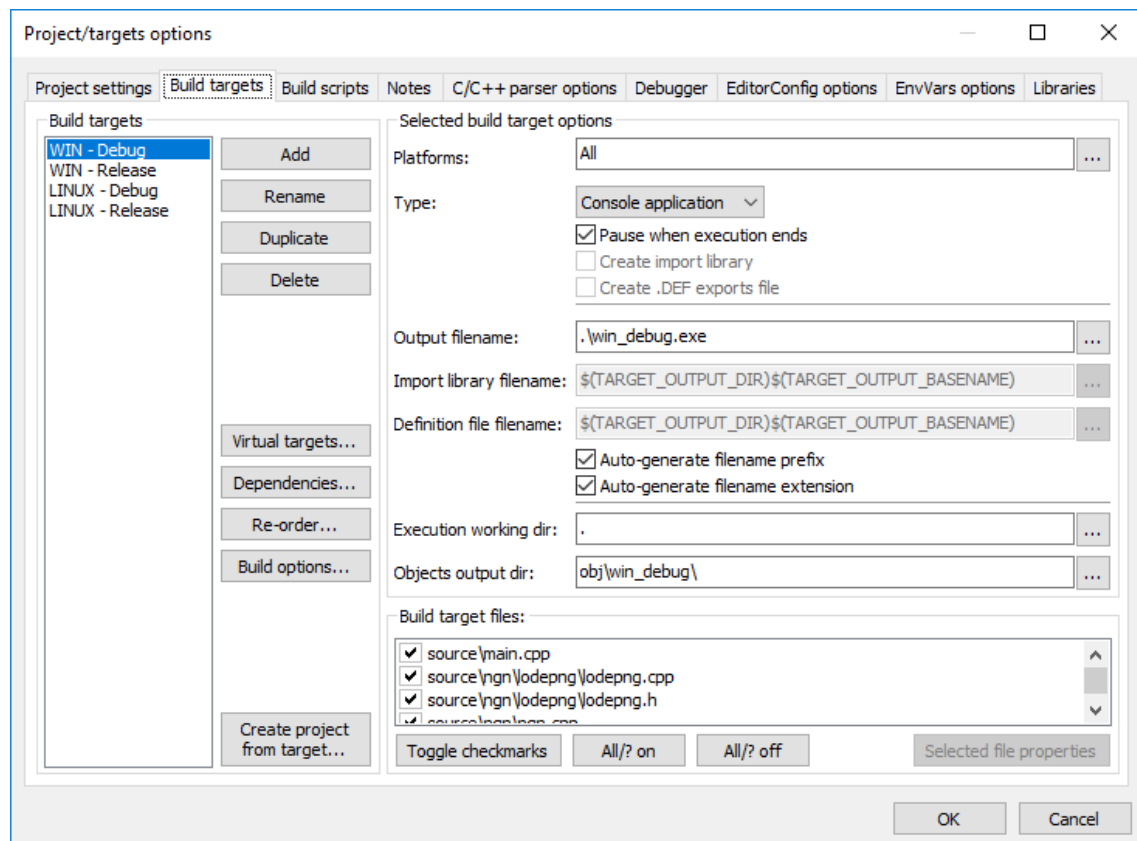
En “Project build options”, en la sección “WIN-RELEASE”, configuraremos los siguientes parámetros:



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



En “Project/target options”, configuraremos los siguientes parámetros:



Instalación sobre CODE::BLOCKS 20 - WINDOWS

La versión 20 de CODE::BLOCKS incluye la versión más reciente del compilador TDM-GCC MinGW. Si hemos instalado la versión de CODE::BLOCKS que incorpora este compilador o si no disponemos de este instalado (instalación de CODE::BLOCKS limpia), deberemos instalar o cambiar la versión del compilador, dado que la última versión estable de la librería SFML esta compilada con MinGW-windows-gcc-8.1.0. y esta no funcionara en versiones anteriores a esta. Para dicha instalación o actualización deberemos de seguir los siguientes pasos:

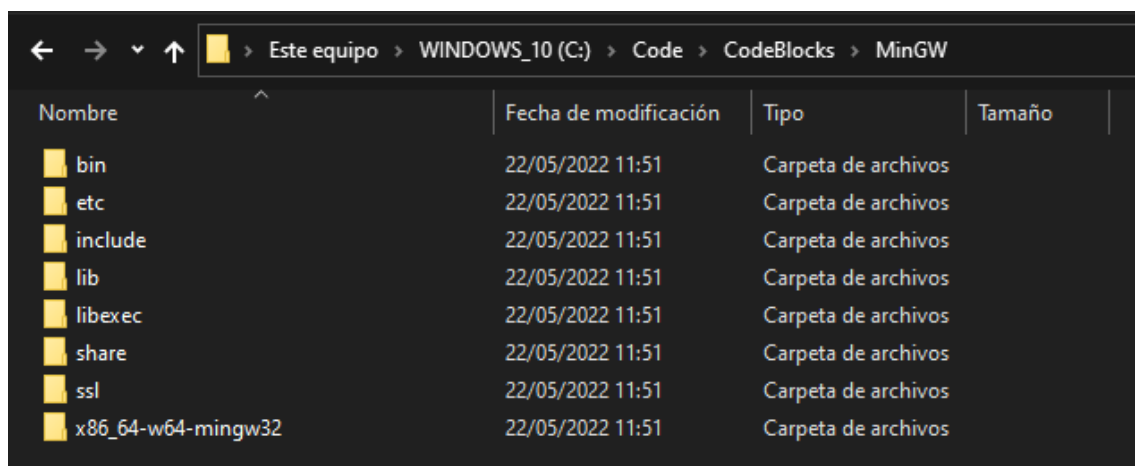
1 - Descargaremos el instalador del compilador desde el siguiente enlace:

<https://www.mingw-w64.org/>

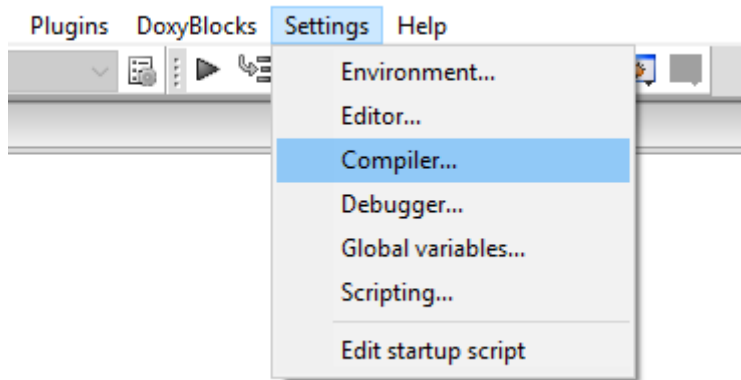
(a fecha de esta revisión, yo instalé la versión 11.3.0 usando MYSYS2 <https://www.mingw-w64.org/downloads/#msys2>)

2 - En la carpeta de instalación de CODE::BLOCKS borraremos todo el contenido de la carpeta **MinGW**, de no existir esta carpeta, la crearemos.

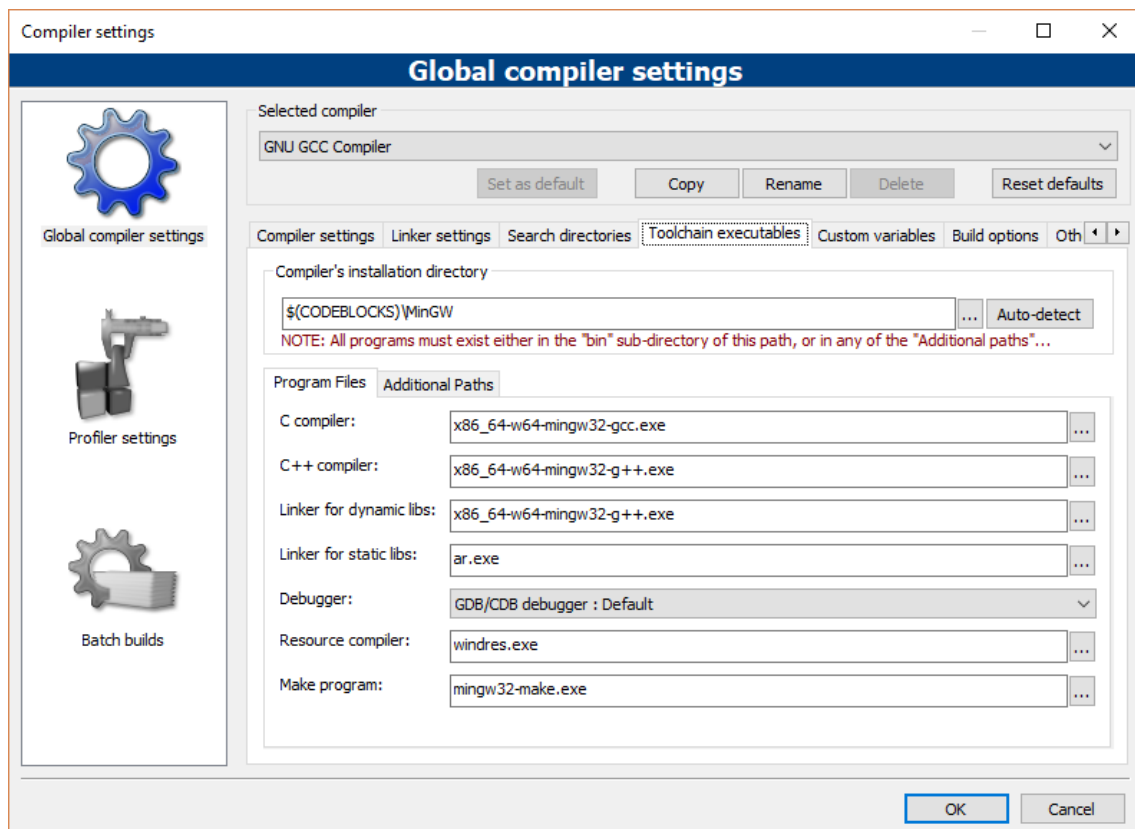
3 - Copiaremos todo el contenido del archivo comprimido que hemos descargado en su interior.



4 - Ejecutaremos CODE::BLOCKS para que detecte el compilador que acabamos de instalar. Si no lo hace de manera automática, iremos **SETTINGS, COMPILER**:



Nos colocaremos en la pestaña **TOOLCHAIN EXECUTABLES** e introduciremos la ruta donde estén los archivos del compilador. Si queremos que sea portable, podemos usar la variable de sistema **\$(CODEBLOCKS)** como ruta. El botón **AUTO-DETECT** debería detectar los ejecutables necesarios.



Hecho esto, CODE::BLOCKS debería de compilar sin problemas.

Instalación de las librerías - Linux y Raspbian

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:

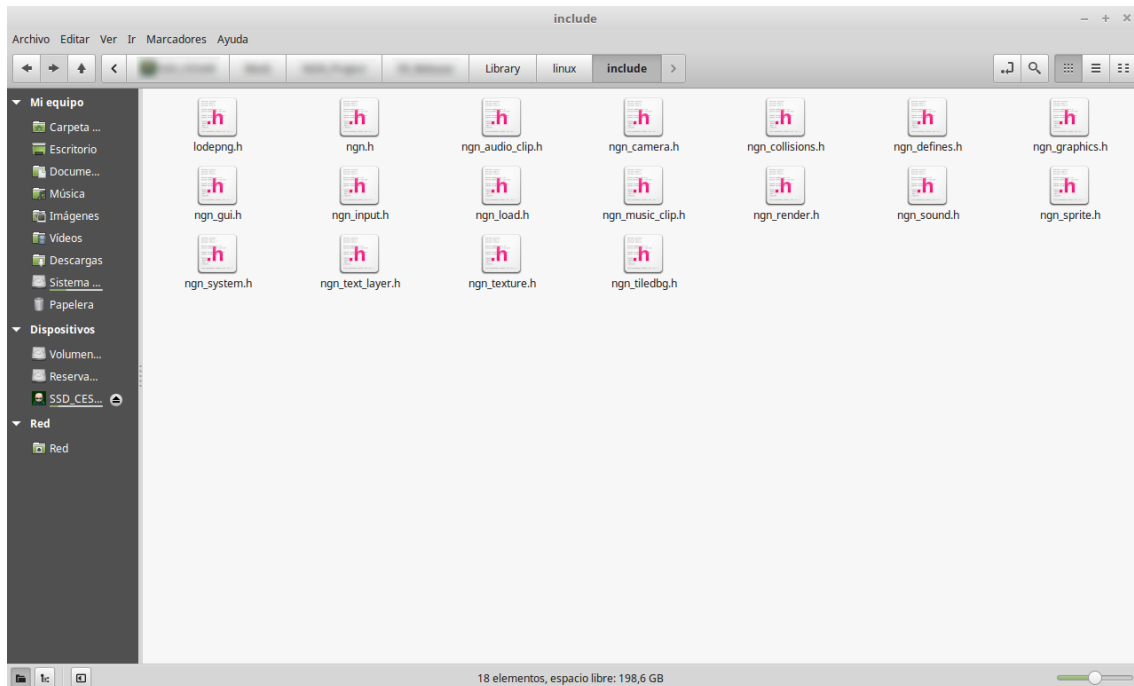
```
sudo apt-get update  
sudo apt-get install libsFML-dev
```

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

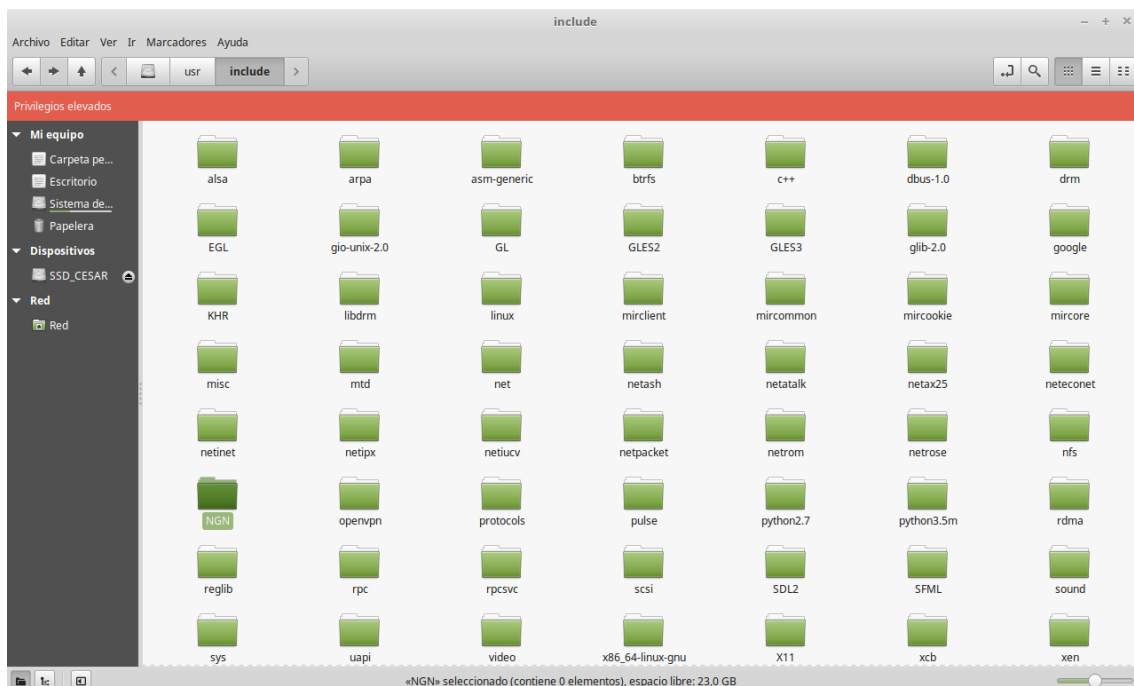
```
sudo apt-get update  
sudo apt-get install build-essential
```


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

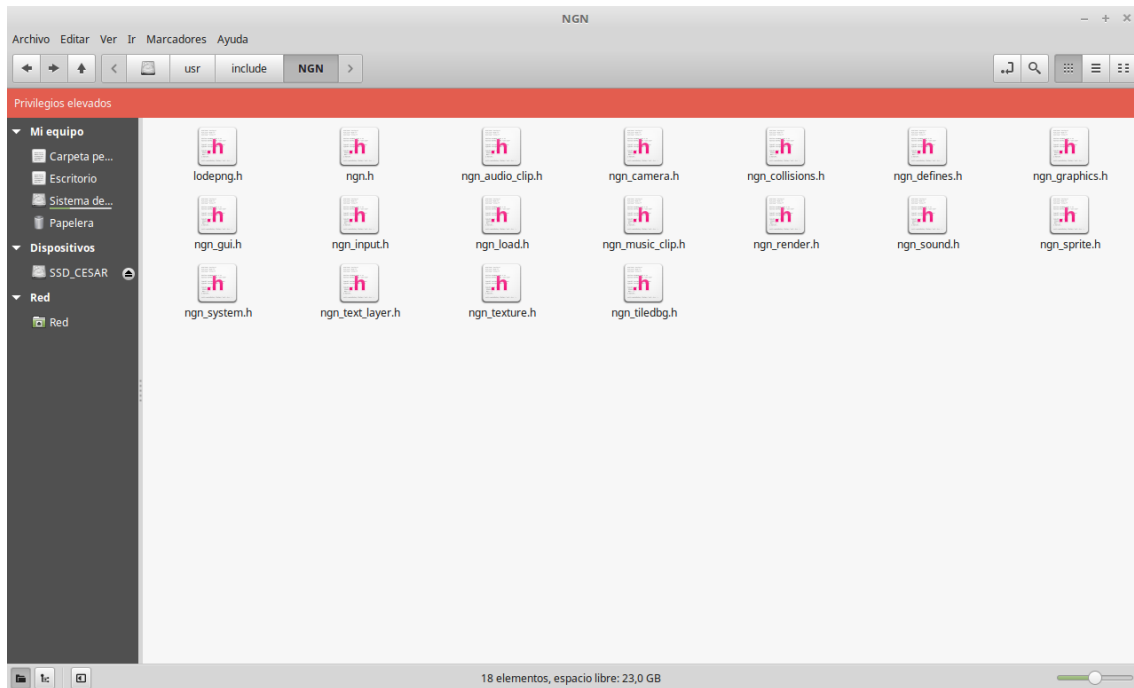
Copiaremos los archivos INCLUDE (.h) contenidos en la librería:



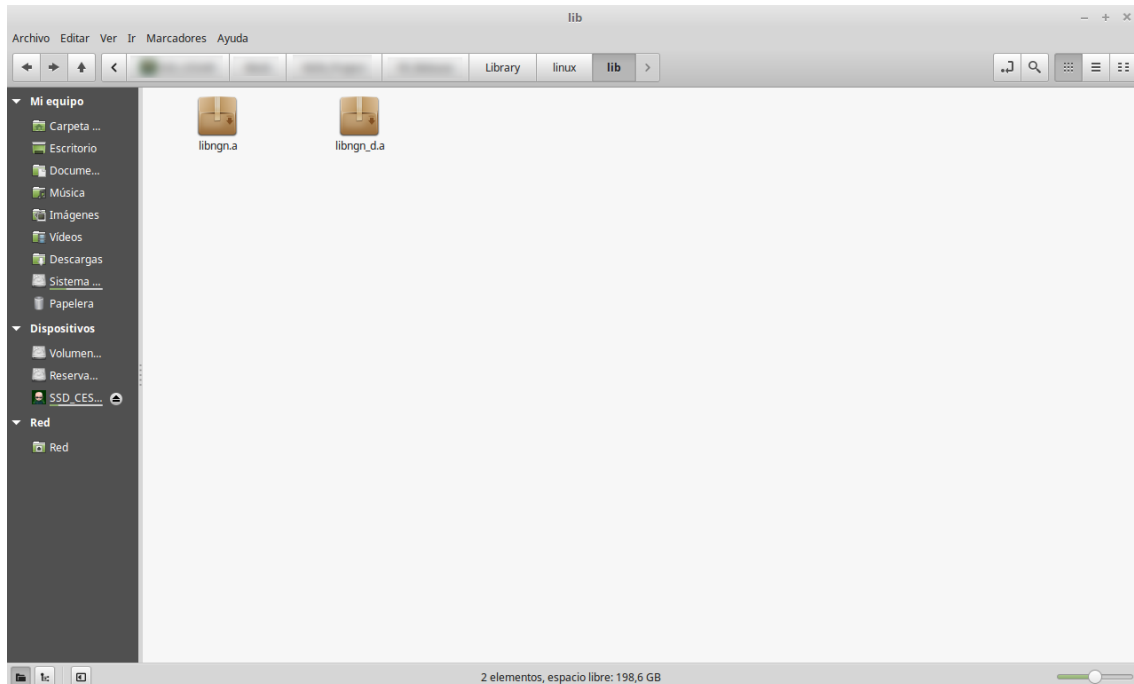
Nos colocaremos en la carpeta del sistema “usr/include” con privilegios elevados y crearemos una carpeta con el nombre “NGN” (en mayúsculas):



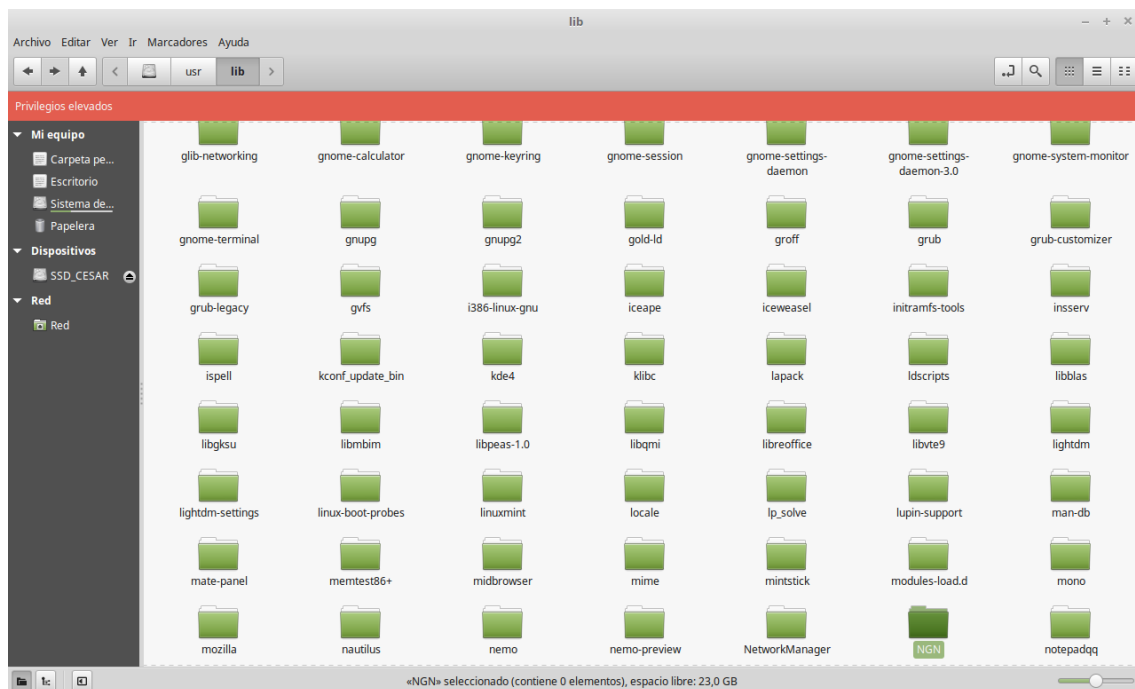
A continuación, pegaremos los archivos .h en su interior:



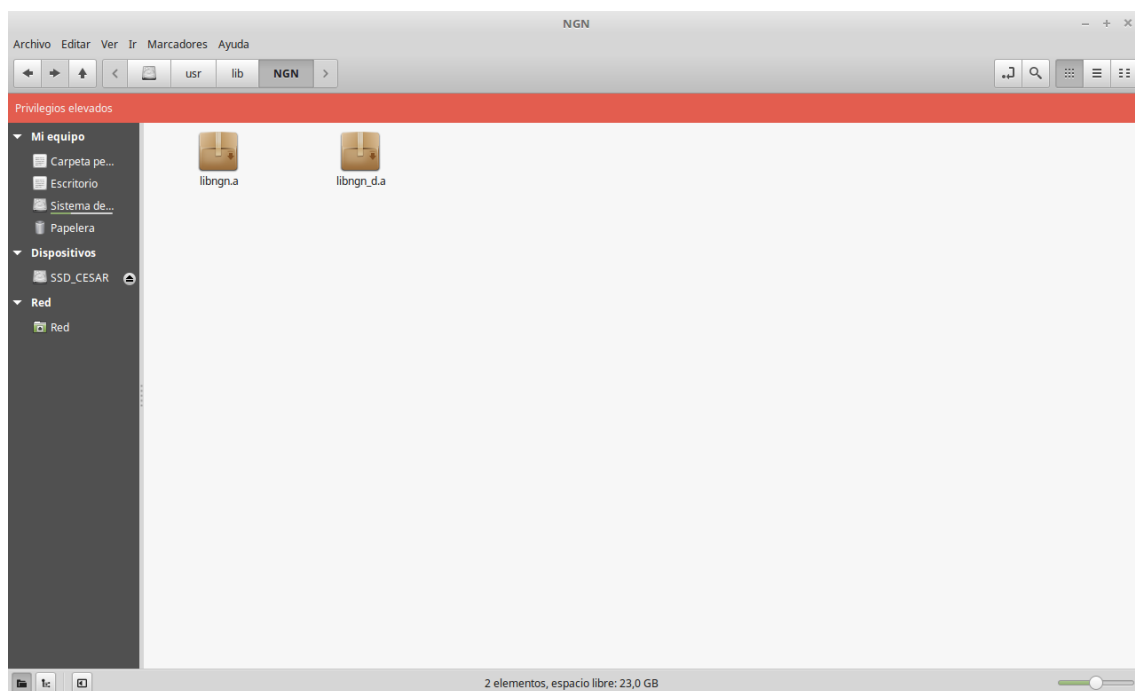
Ahora copiaremos los archivos binarios de la librería (.a) de la carpeta lib:



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

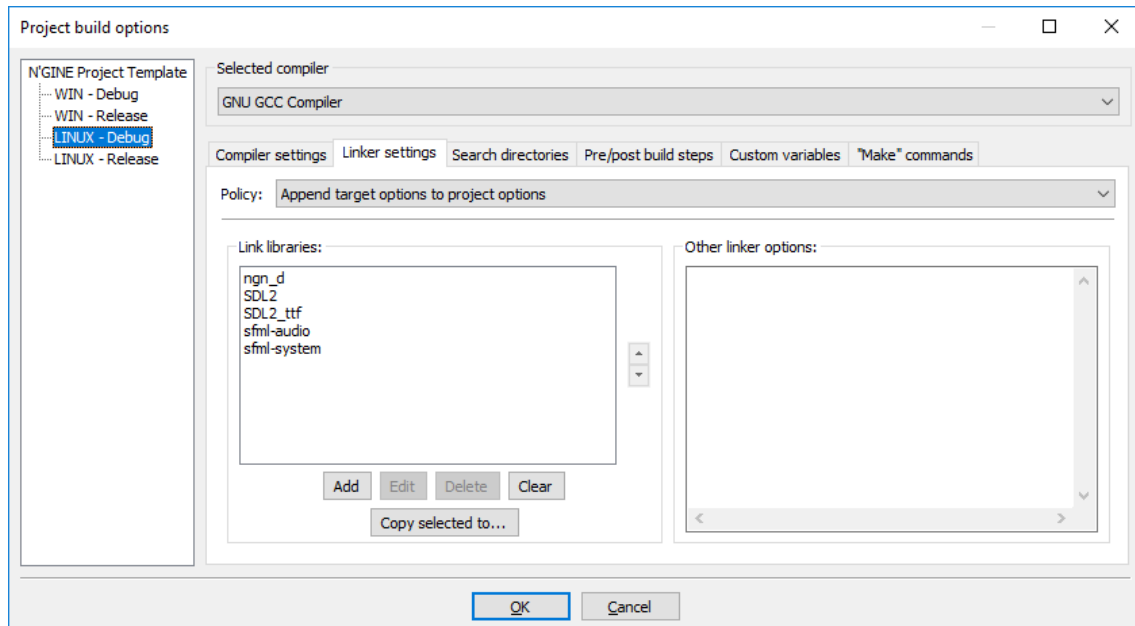


Para terminar, pegaremos los archivos .a en el interior de esta carpeta:

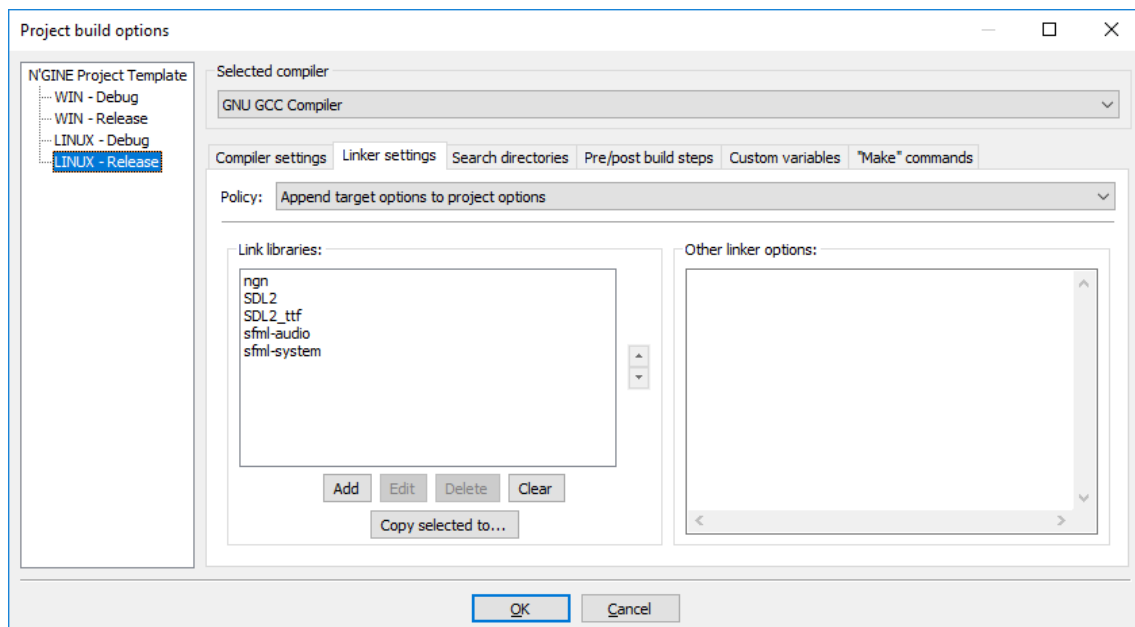


Configuración del proyecto - CODE::BLOCKS

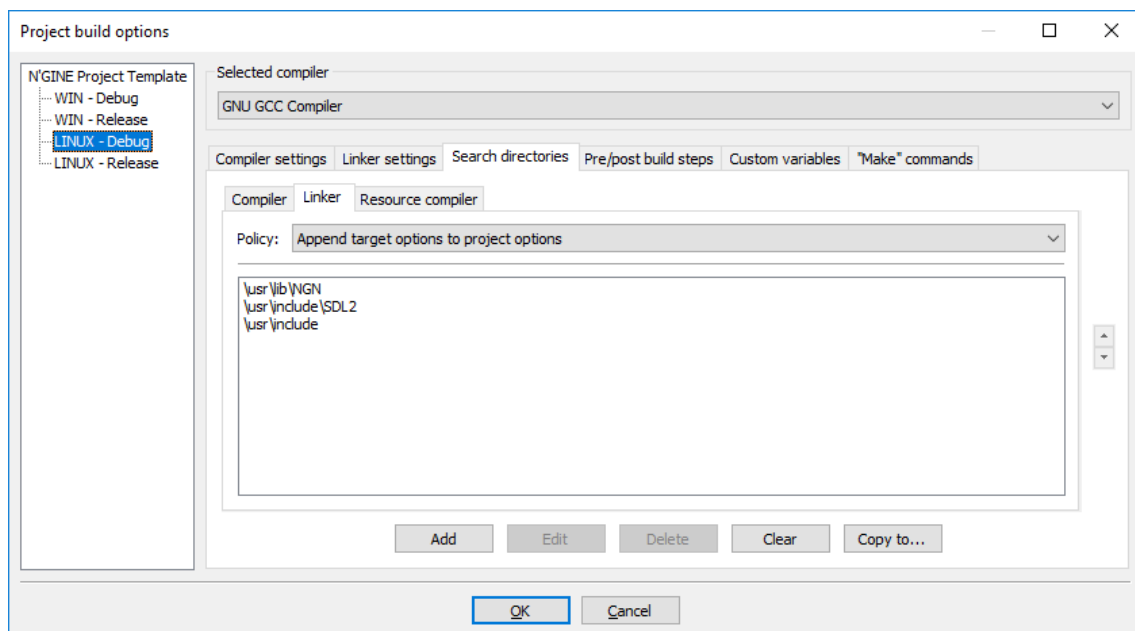
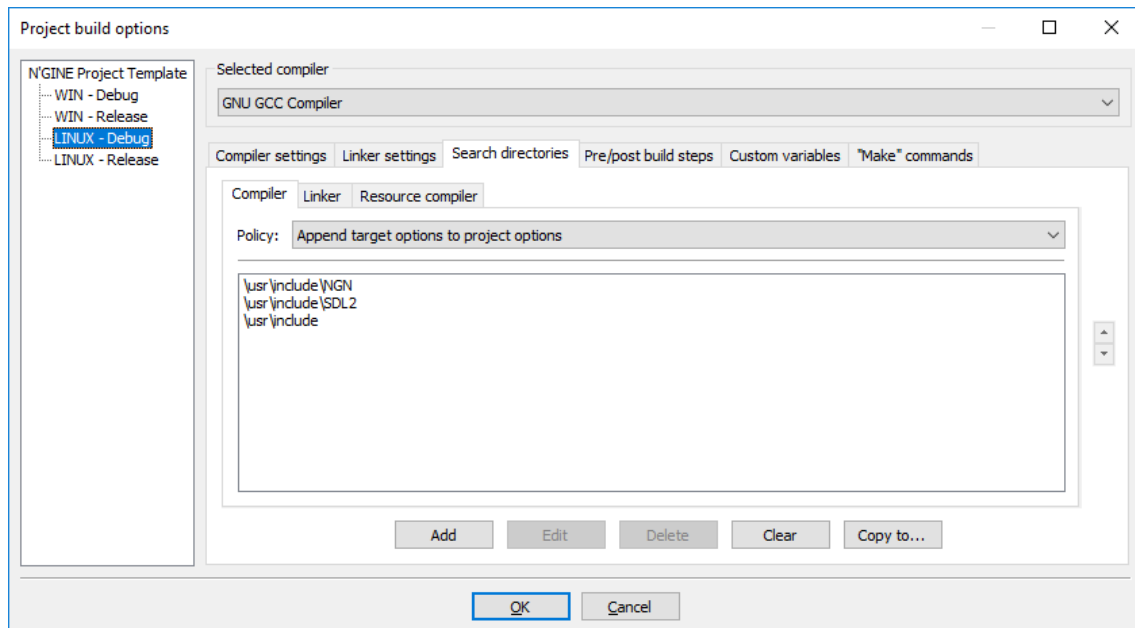
En “Project build options”, en la sección “LINUX-DEBUG”, configuraremos los siguientes parámetros:



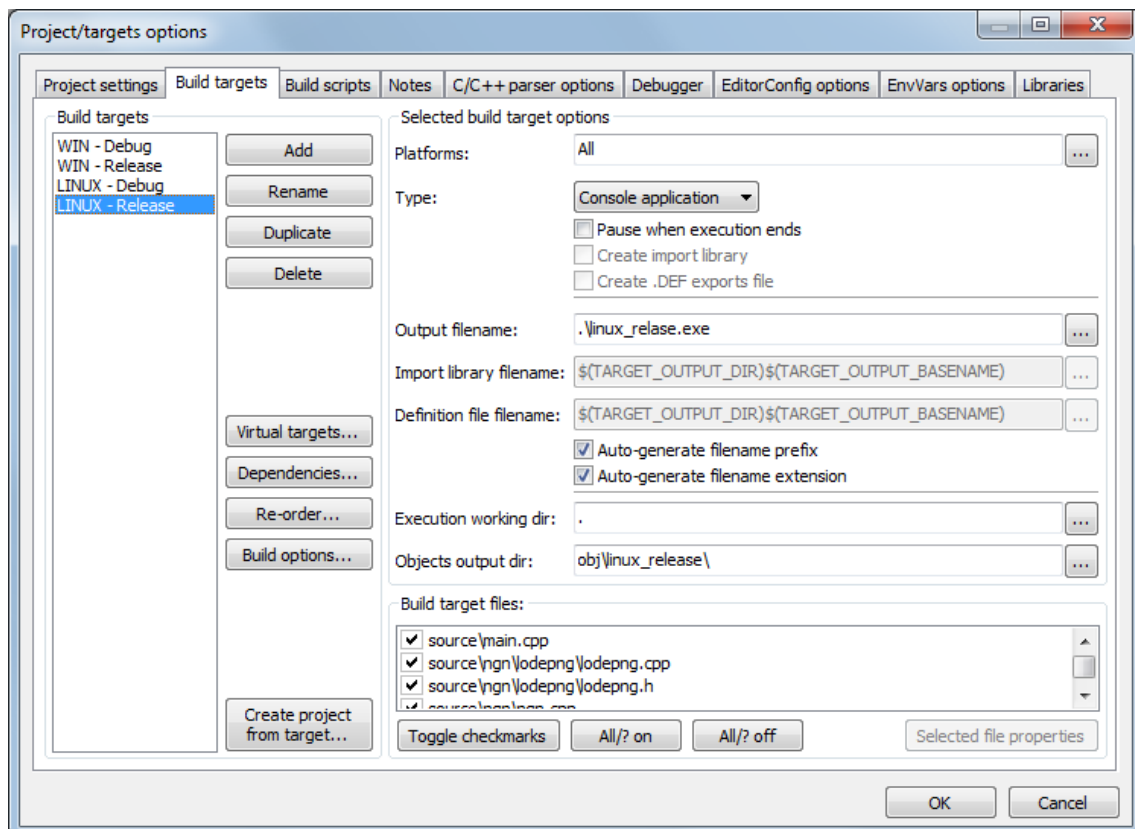
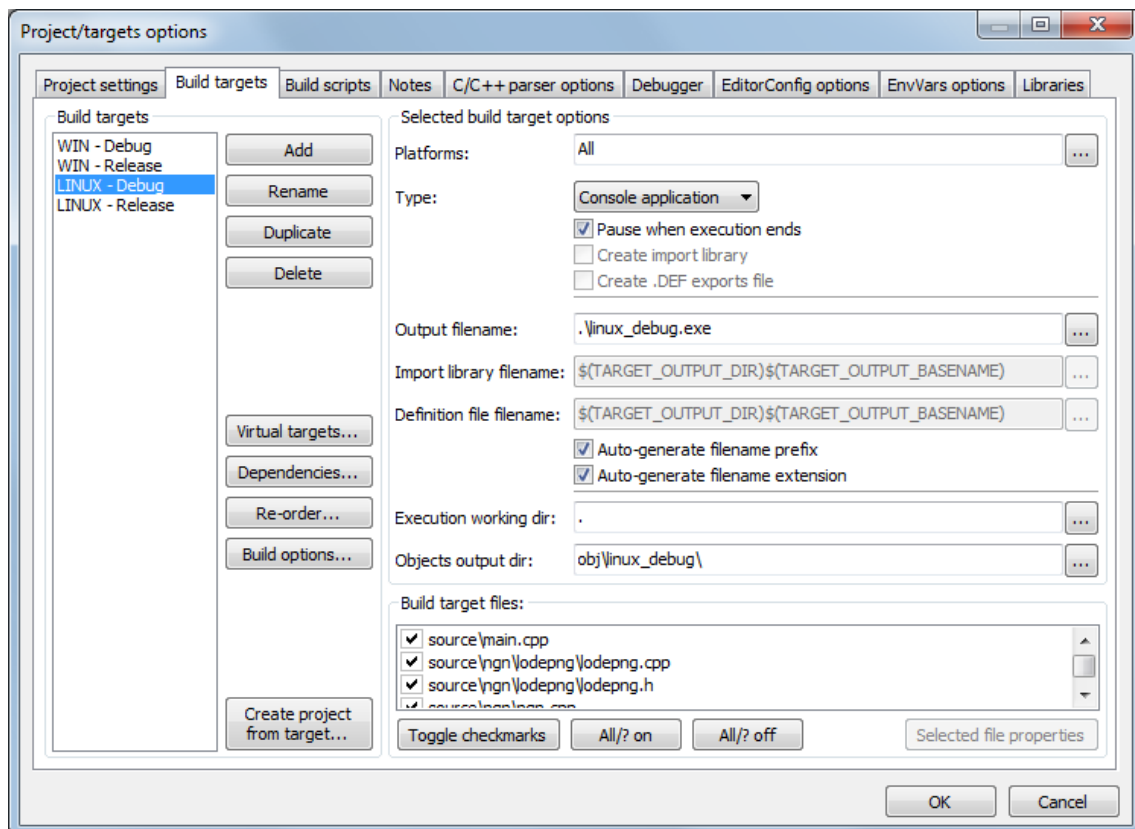
En “Project build options”, en la sección “LINUX-DEBUG”, configuraremos los siguientes parámetros:



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



En “Project/target options” configuraremos los siguientes parámetros:



Opcionalmente, la librería incluye diversas plantillas con todas estas opciones ya configuradas, tanto para Windows como para Linux y Raspbian. En caso de usar estas plantillas en la creación de los proyectos, solo será necesario realizar el paso de instalación de las librerías.

Para los entornos Linux Mint y Raspbian se incluyen también unos scripts de instalación automática con las ultimas versiones de la librería N'gine, así como las versiones usadas de SDL2, SDL2_ttf y SFML3. Estas versiones se instalarán en las carpetas del sistema correspondientes y puede que eliminen las versiones existentes de dichas librerías.

- Fecha de revisión: viernes, 8 de diciembre del 2023.