

Introducción a Code::Blocks

Proyectos

- Creación
- Administración y configuración

Depuración

Code::Blocks

Es un entorno de desarrollo integrado o IDE (Integrated Development Enviroment) para los lenguajes de programación C y C++.

Sus componentes y características principales son:

- Administrador de espacios de trabajo (workspaces) y proyectos.
- Editor con marcador de sintaxis personalizable y autocompletado.
- Soporte para varios compiladores. Durante el curso utilizaremos el compilador de C++ de la GCC (GNU Compiler Collection). En plataformas Windows, utilizaremos la portación de este compilador llamada MinGW, la cual ya viene incluida en uno de los paquetes binarios provistos por Code::Blocks.
- Soporte para ejecución y depuración de programas.

Tanto Code::Blocks como GCC y MinGW son software libre. Para más información y descargas:

<http://www.codeblocks.org>

Proyectos

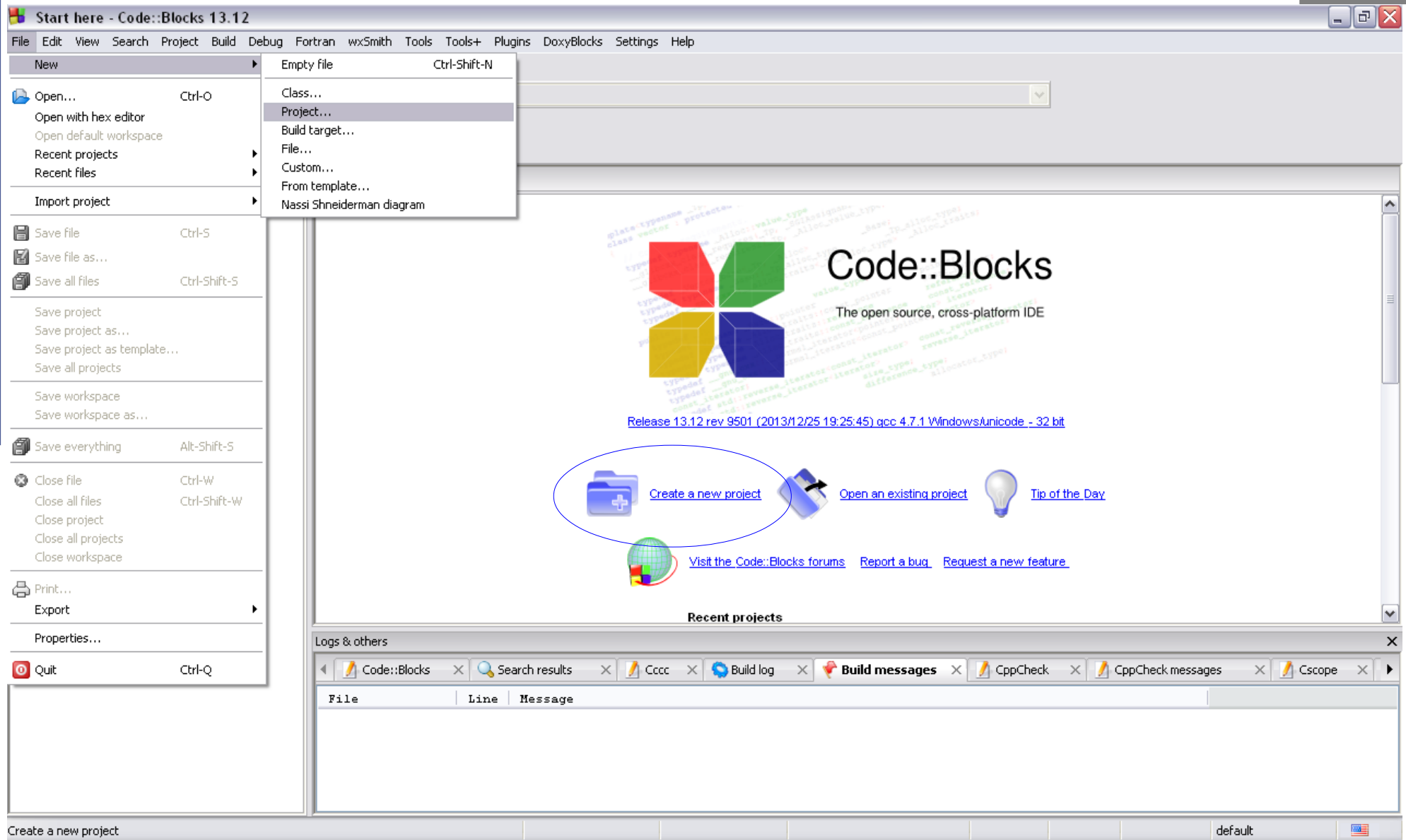
Un proyecto agrupa los archivos de código fuente que forman la aplicación que desarrollamos, y guarda la configuración específica para poder compilarla (por ej. bibliotecas y otros recursos utilizados, optimizaciones de compilación, etc.).

La estructura física de un proyecto (en el sistema de archivos):

Carpeta del *proyecto*

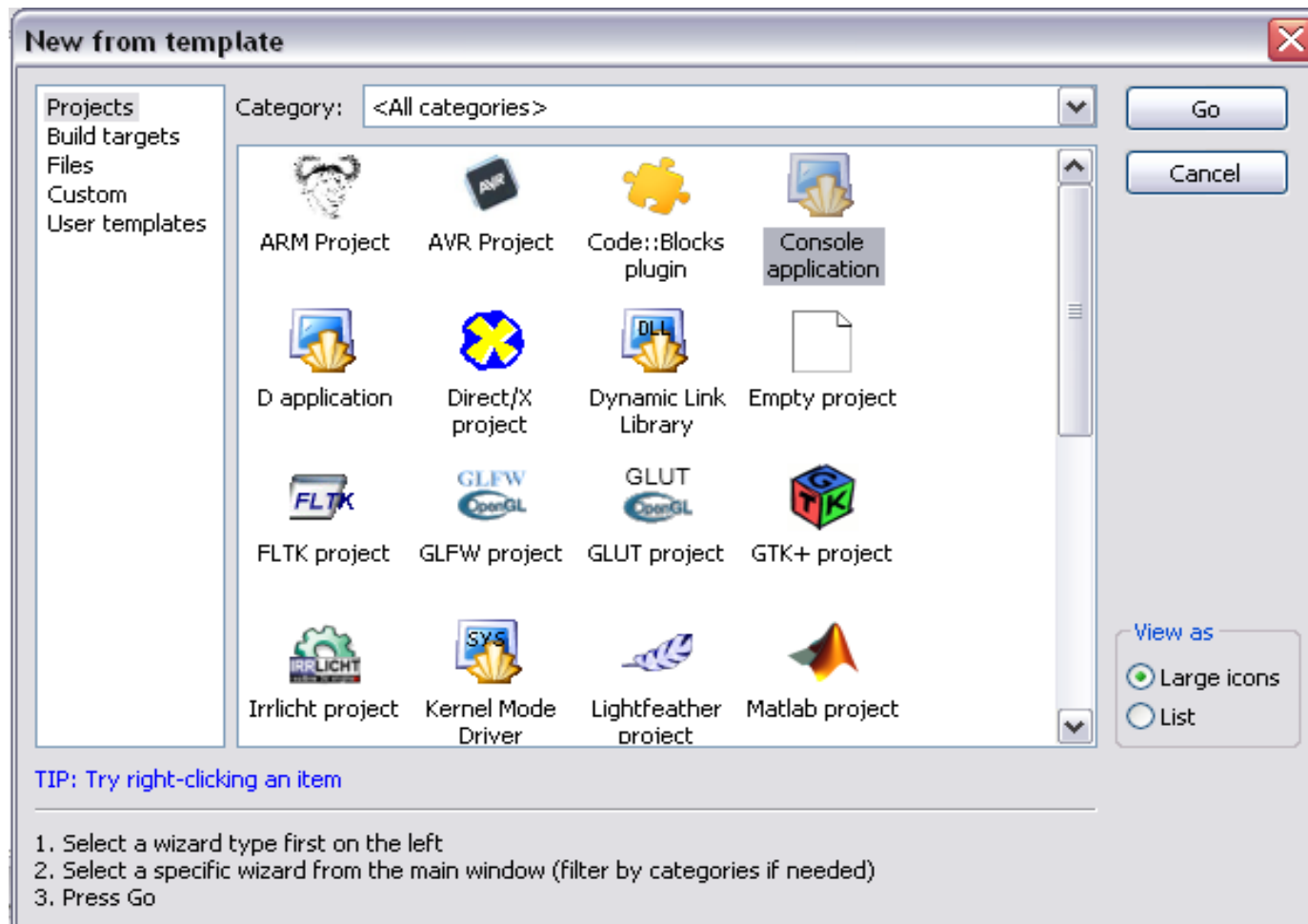
- *proyecto.cdb* (el archivo de configuración del proyecto)
- *main.cpp* (o *proyecto.cpp*)
- ...
- *biblioteca_x.h*
- *biblioteca_x.cpp*
- *biblioteca_y.h*
- *biblioteca_y.cpp*
- ...

Creación de un proyecto



Creación de un proyecto

Se nos presentará una serie de plantillas a partir de las cuales podemos definir los proyectos:



Creación de un proyecto

Elegimos “Console Application”, y luego veremos una pantalla de selección del lenguaje a usar. A continuación podremos elegir el nombre y carpeta de destino de nuestro proyecto.

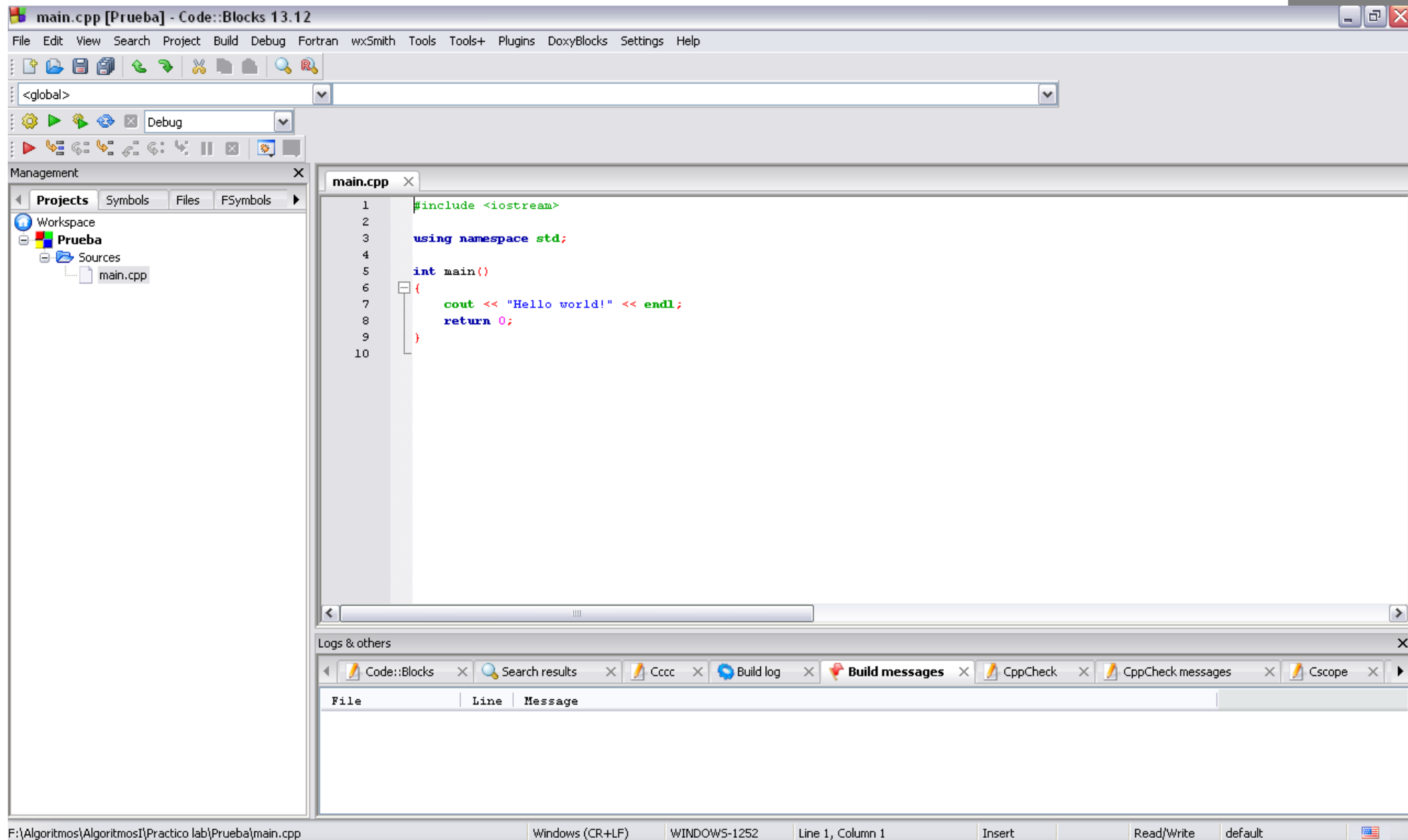


Creación de un proyecto

Por último, debemos seleccionar el compilador y las configuraciones a crear:

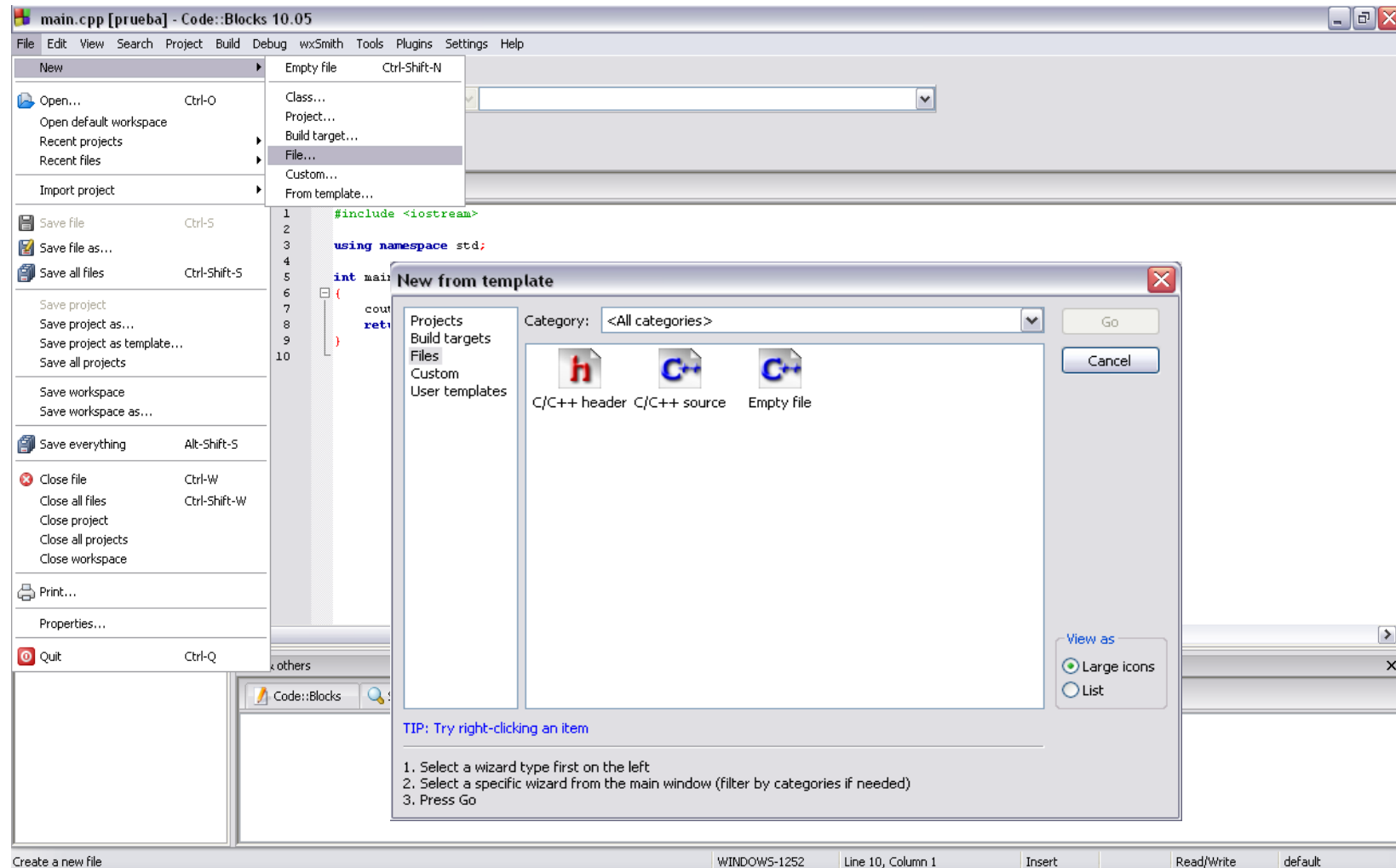


Creación de un proyecto



Creación de proyectos

Para crear un nuevo archivo:

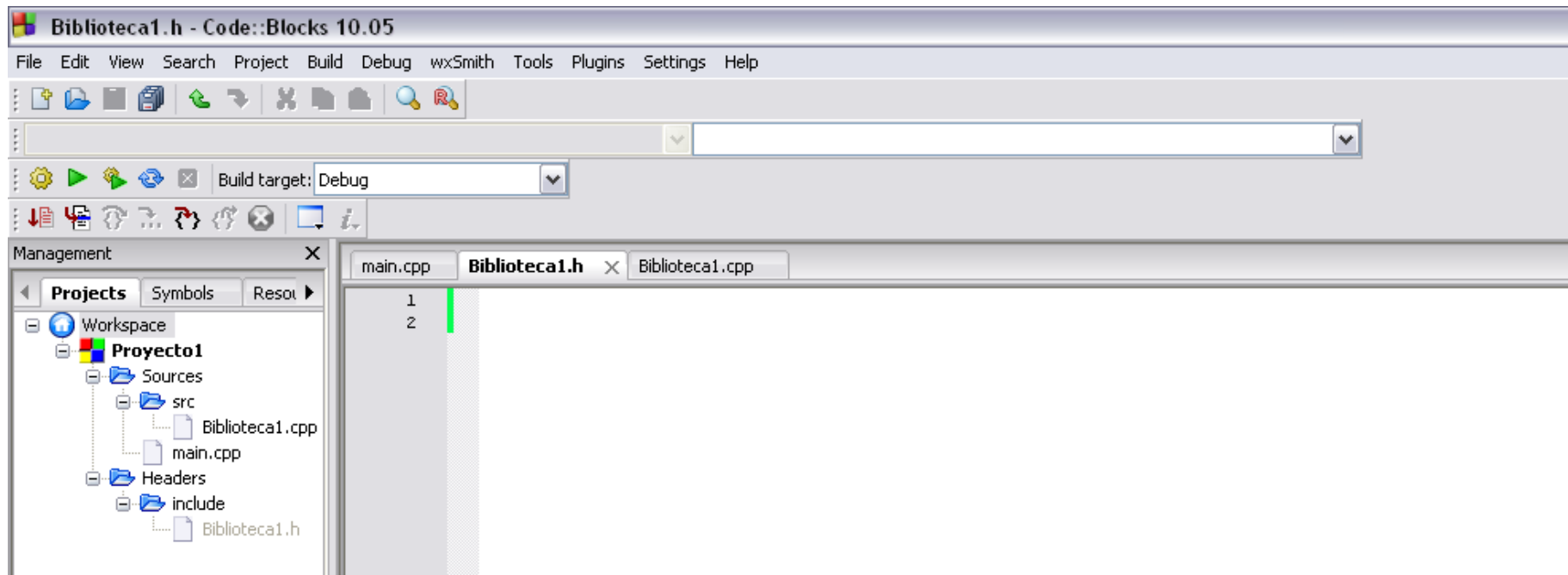


Administración de proyectos

Al crear un proyecto, Code::Blocks nos ayudará a administrar los archivos de código fuente del mismo.

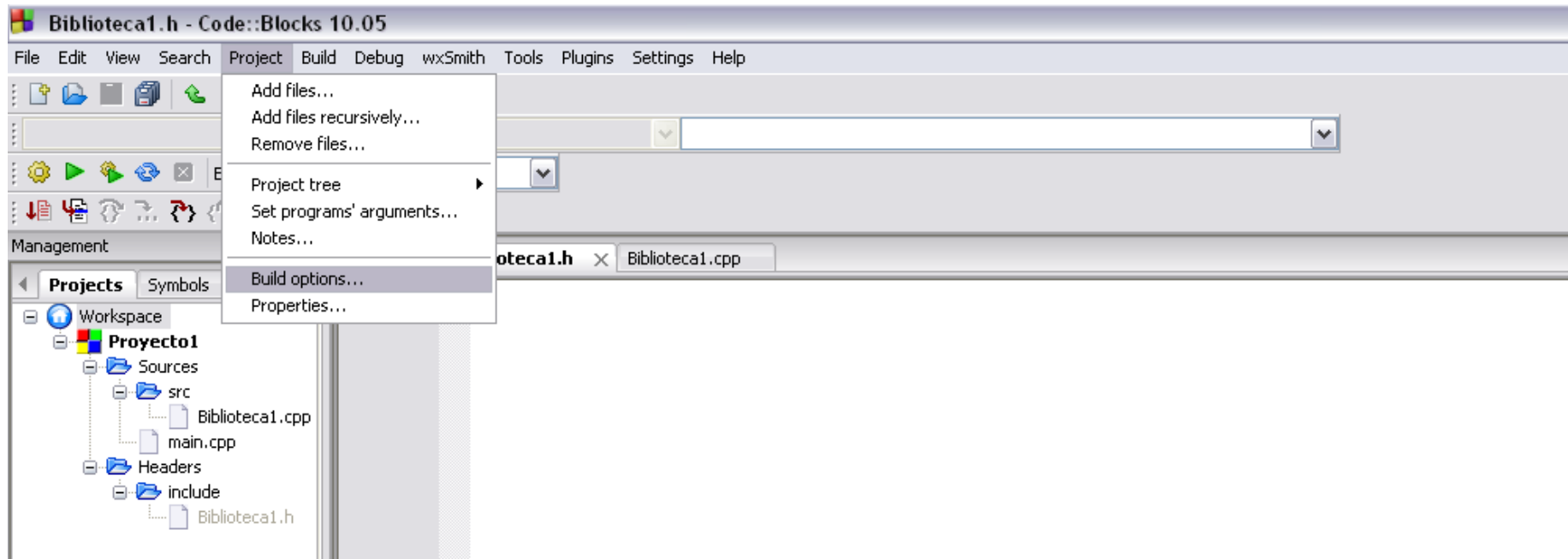
Entre otra funciones, el entorno determinará qué archivos hay que (re)compilar, en base a las modificaciones hechas.

También nos permitirá editar y organizar archivos (mover, renombrar, etc).



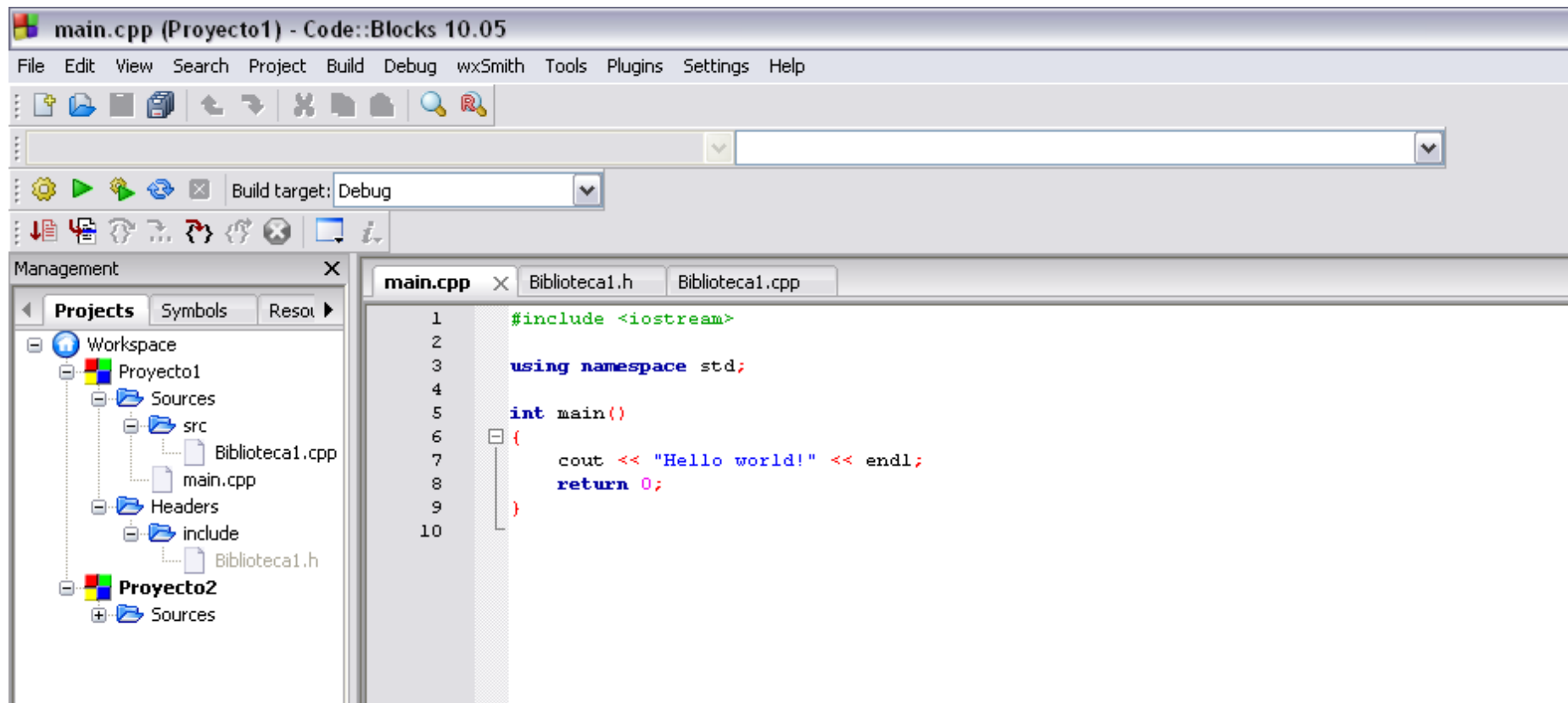
Configuración de proyectos

Las opciones de un proyecto específico (**Build options** y **Properties**) nos permiten cambiar la configuración del compilador, las bibliotecas utilizadas, nombre e información incluidas en el ejecutable, entre otras opciones.



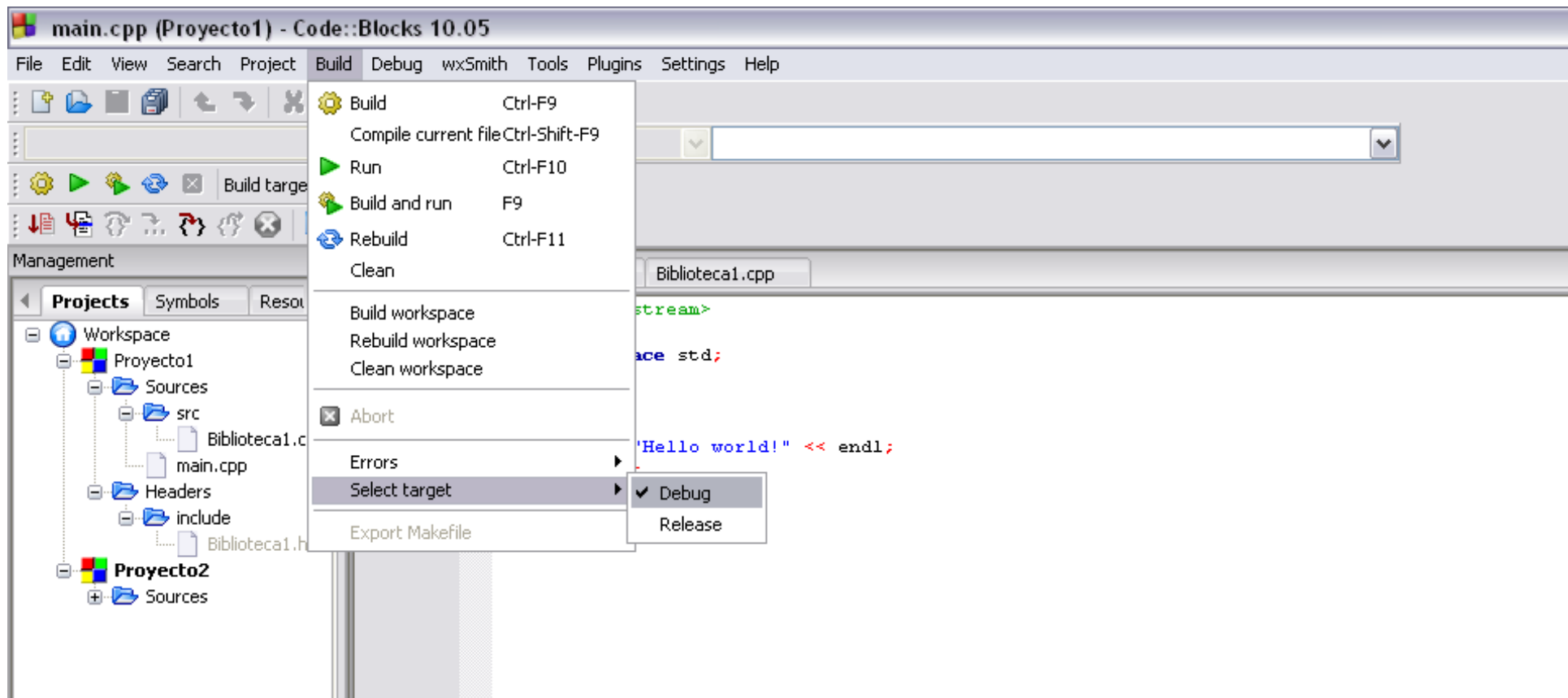
Espacios de trabajo

Los espacios de trabajo contienen a los proyectos. Pueden existir varios espacios de trabajo pero sólo uno puede estar activo al mismo tiempo. Además de contener a los proyectos el espacio de trabajo lleva registro de las dependencias entre ellos.



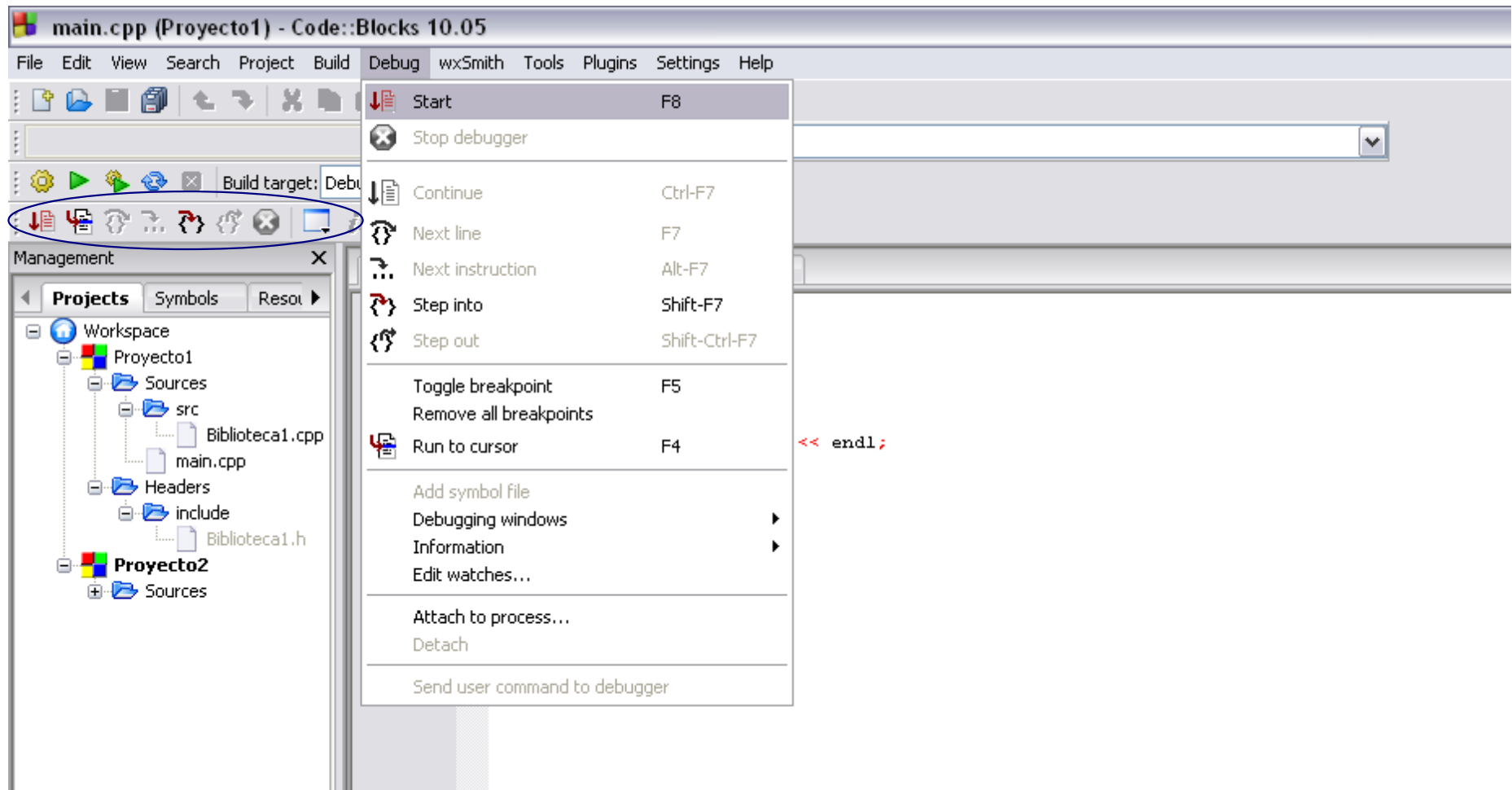
Depuración - Configuración

Para poder depurar los proyectos es que existe la configuración “**Debug**” que podremos seleccionar cuando creamos los proyectos. Para poder utilizarla debemos habilitarla como objetivo del proceso de construcción.



Depuración

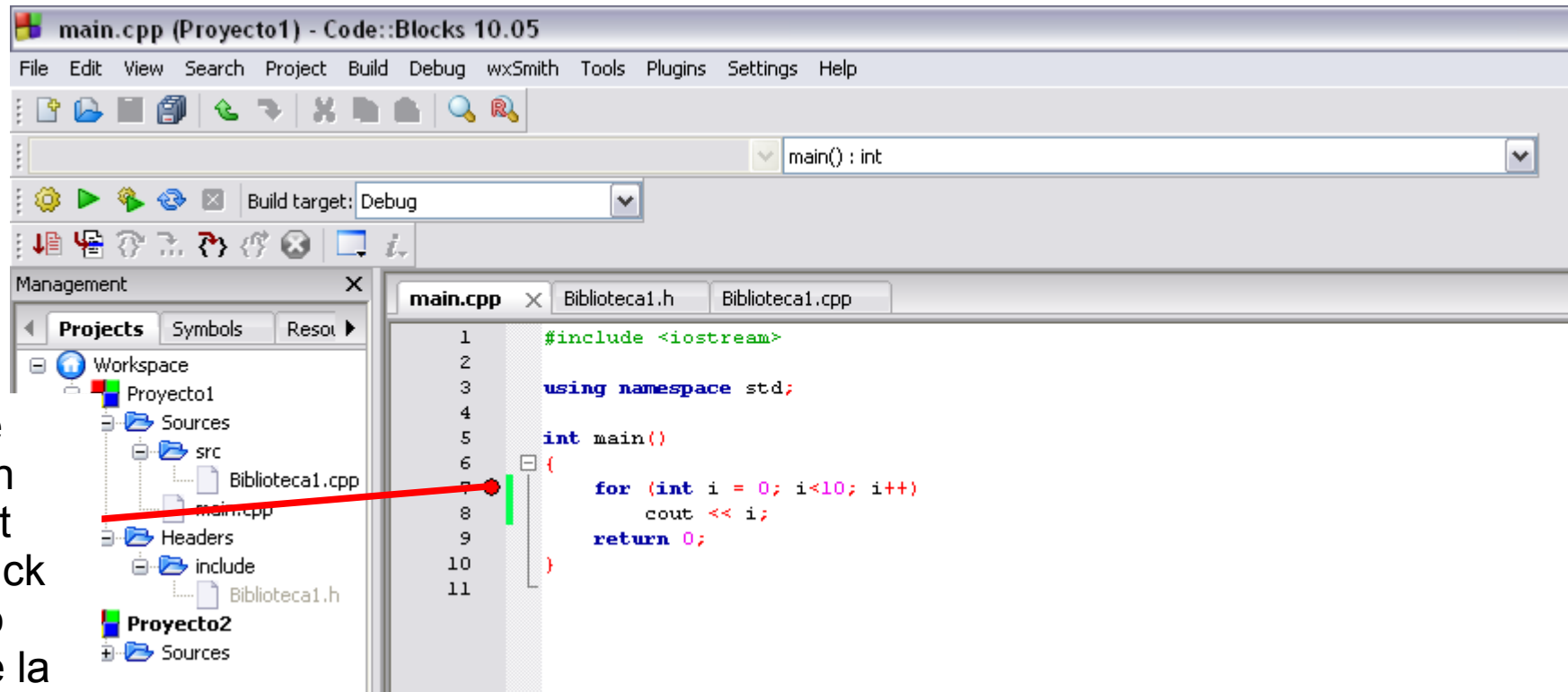
Hay dos lugares donde se encuentran los controles de depuración de la aplicación, así como una serie de accesos rápidos por teclado:



Depuración

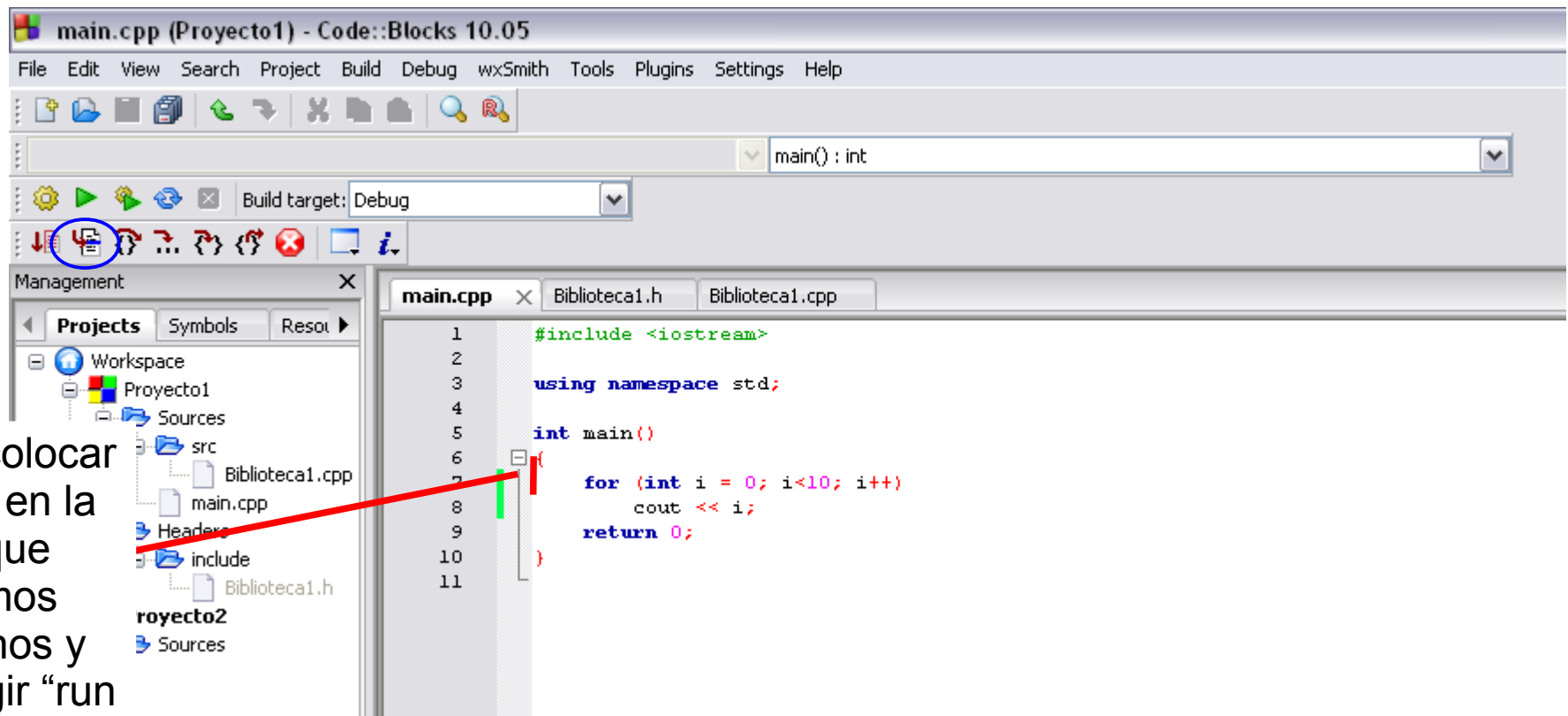
Para depurar se debe detener la ejecución del programa en algún punto de interés. Para esto utilizamos: **breakpoints**.

Se puede agregar un breakpoint haciendo click en el lado izquierdo de la línea en el editor



Depuración

Otra opción más dinámica es la de avanzar la ejecución hasta donde se encuentre el cursor.



Hay que colocar
el cursor en la
línea que
queremos
detenernos y
luego elegir "run
to cursor"

Depuración

Luego podremos controlar la ejecución paso a paso, con la posibilidad de entrar a funciones, permitir que la misma continúe normalmente (hasta el próximo breakpoint) o detenerla.

The screenshot shows the wxSmith IDE interface. The 'Debug' menu is open, displaying various debugging actions and their keyboard shortcuts. Red circles highlight the 'Stop debugger', 'Continue', 'Next line', and 'Step into' options. Red arrows point from these options to labels on the right: 'Detener' for 'Stop debugger' and 'Continuar' for 'Continue'. Another red arrow points from the 'Step into' option to the text 'Paso a paso (sin entrar en funciones)'. A final red arrow points from the 'Step into' option to the text 'Entrar en la función'.

main.cpp (Proyecto1) - Code::Blocks 10.05

File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Help

Start F8

Stop debugger

Continue Ctrl-F7

Next line F7

Next instruction Alt-F7

Step into Shift-F7

Step out Shift-Ctrl-F7

Toggle breakpoint F5

Remove all breakpoints

Run to cursor F4

Add symbol file

Debugging windows

Information

Edit watches...

Attach to process...

Detach

Send user command to debugger

Management

Projects Symbols Reso

Workspace

Proyecto1

Sources

src

main.cpp

Headers

include

Biblioteca1.h

Biblioteca1.cpp

Proyecto2

sources

Paso a paso
(sin entrar en
funciones)

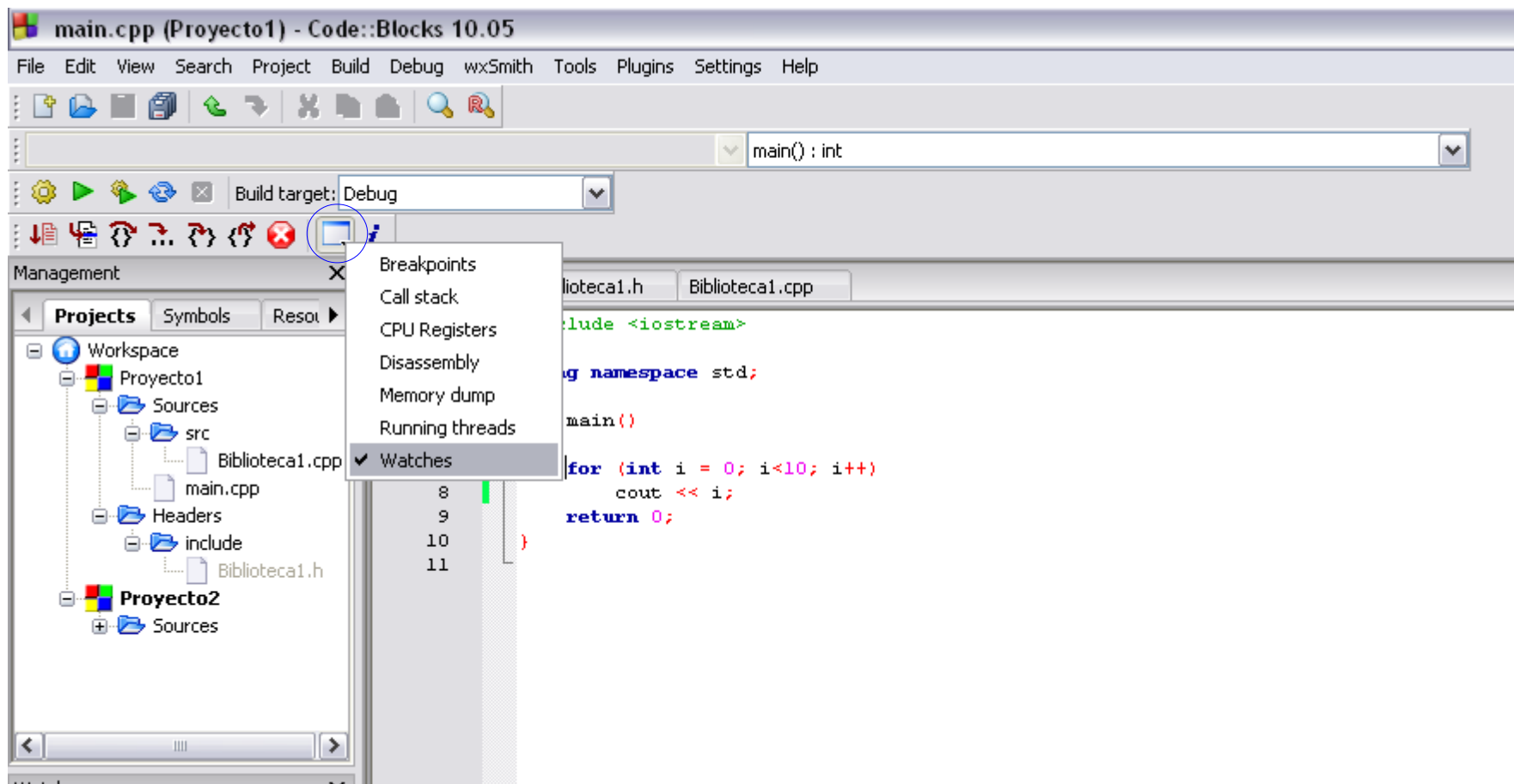
Entrar en la
función

Detener

Continuar

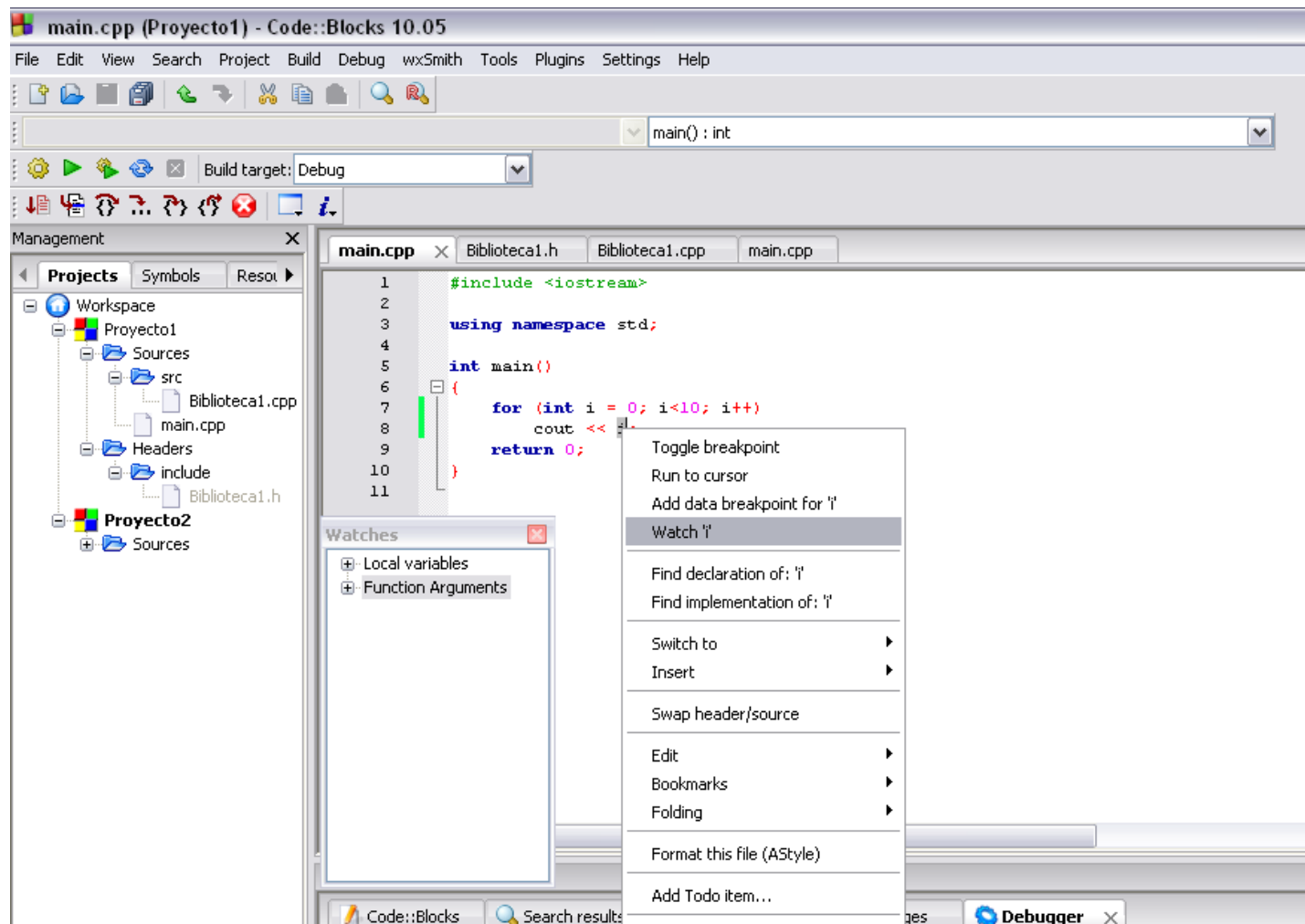
Depuración

El proceso de depuración nos permite observar (**watch**) el valor de cualquier variable durante el proceso de depuración. Para esto primero tenemos que activar la vista de variables observadas (**Watches**)



Depuración

También se puede agregar un watch pintando en el editor el nombre de la variable y eligiendo “Watch <variable>”.



Federico Améndola

Martín Fernández

Consultas: laboratorio.ayda@alumnos.exa.unicen.edu.ar

Licencia creative commons

Atribución-Compartir Obras Derivadas Igual 2.5 Argentina



<http://creativecommons.org/licenses/by-sa/2.5/ar/>