

# Aplicaciones con interfaz gráfica

## Un inicio rápido utilizando wxWidgets y C++ con code::blocks – parte 1

**wxWidgets** es un conjunto de bibliotecas que permiten desarrollar “interfaces gráficas con el usuario” (o GUI: graphical user interface) y son gratis, de código abierto y multiplataforma (se puede desarrollar una aplicación y con pocos o ningún cambio compilarla para varios sistemas operativos: Windows, Unix, Linux, MacOS X, iOS, openVMS, AmigaOS, OS/2. Están programadas en C++ y pueden ser utilizadas en este lenguaje o desde otros lenguajes (Python, PHP, Perl, Ruby, Haskell, C, D, etc.)

**Code::Blocks** es un entorno integrado de desarrollo desarrollado utilizando wxWidgets. Incluye en la instalación por default una “extensión” (plugin) que permite diseñar en forma visual la(s) ventana(s) para incluir en una aplicación.

¿Qué se necesita para comenzar?

Instalar el code::blocks y las librerías wxWidgets. Dependiendo del sistema operativo esto es más o menos complejo. En linux se debe instalar el paquete de wxWidgets de la versión deseada. En windows se puede bajar el código fuente de las librerías y compilarlo, o se puede descargar un instalador que realiza estas tareas por nosotros: wxPack (<http://wxpack.sourceforge.net/Main/Downloads>)

Este tutorial se realizará con la versión 10.05 de code::blocks y la 2.8.12 de wxWidgets.

Una vez instaladas las wxWidgets, iniciamos el code::blocks...

Elegimos crear un nuevo proyecto (“Create a new project”) y aparece la ventana que nos permite elegir el tipo de proyecto. Elegimos “wxWidgets project” (fig.1)

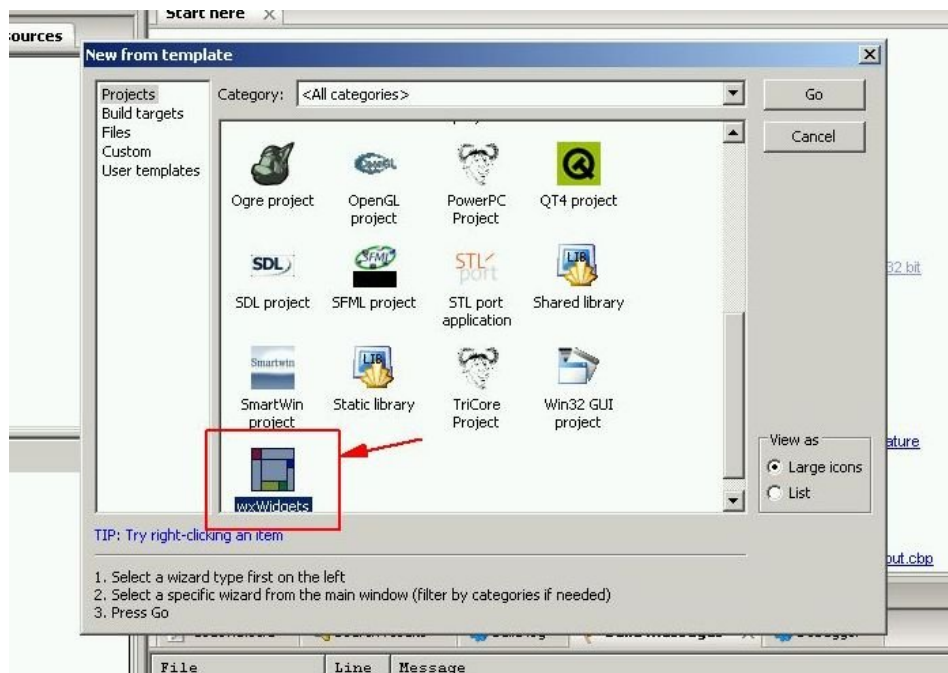


fig. 1 – Elección del tipo de proyecto

Hacer click en el botón “Go”. Aparece otra ventana, donde podemos elegir la versión de wxWidgets a utilizar. De las opciones presentadas, seleccionar “wxWidgets 2.8.x” y luego hacer click en “Next”

En la ventana que aparece escribimos un título para el proyecto (obligatorio) También podemos elegir carpeta donde se guarda el proyecto y el nombre del archivo de proyecto (default: igual que el título del proyecto)

Al hacer click en “Next” aparece una ventana donde podemos ingresar información del autor del proyecto (opcional)

Click una vez más en “next” y aparece una ventana similar a la fig.2

Esta ventana nos permite elegir qué herramienta vamos a utilizar para diseñar la interface con el usuario y qué tipo de aplicación deseamos construir.

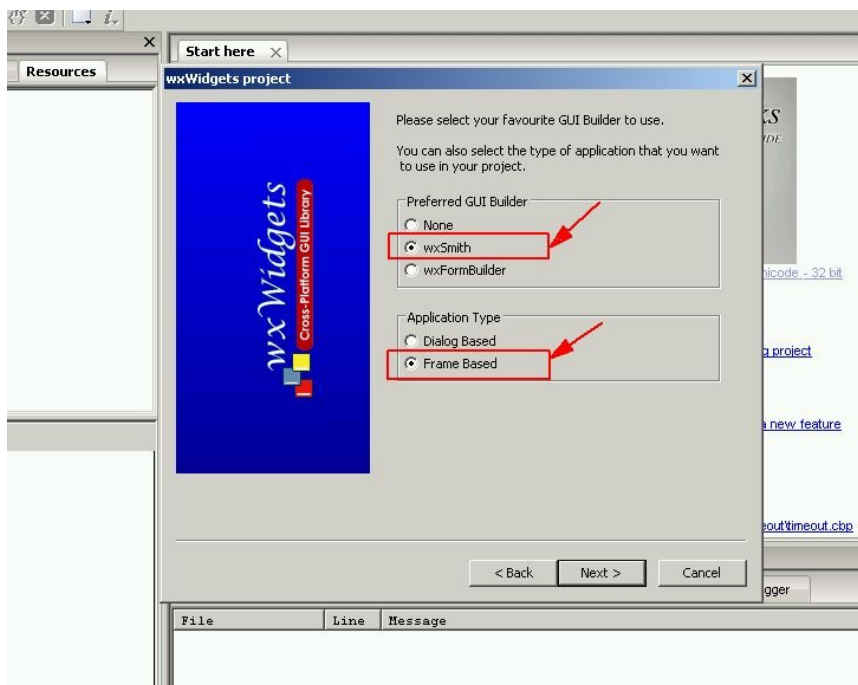


fig. 2 . elección de la herramienta para diseñar la interface gráfica.

Elegimos “wxSmith” (incluido en code::blocks) y “frame based” Luego hacemos click en “Next”

Si es la primera vez que creamos un proyecto con wxWidgets, en la próxima pantalla (ver fig. 3) tenemos que indicar la ubicación de las librerías wxWidgets que instalamos. Por default, con wxPack instala estas librerías en C:\SourceCode\Libraries\wxWidgets2.8 Indicamos

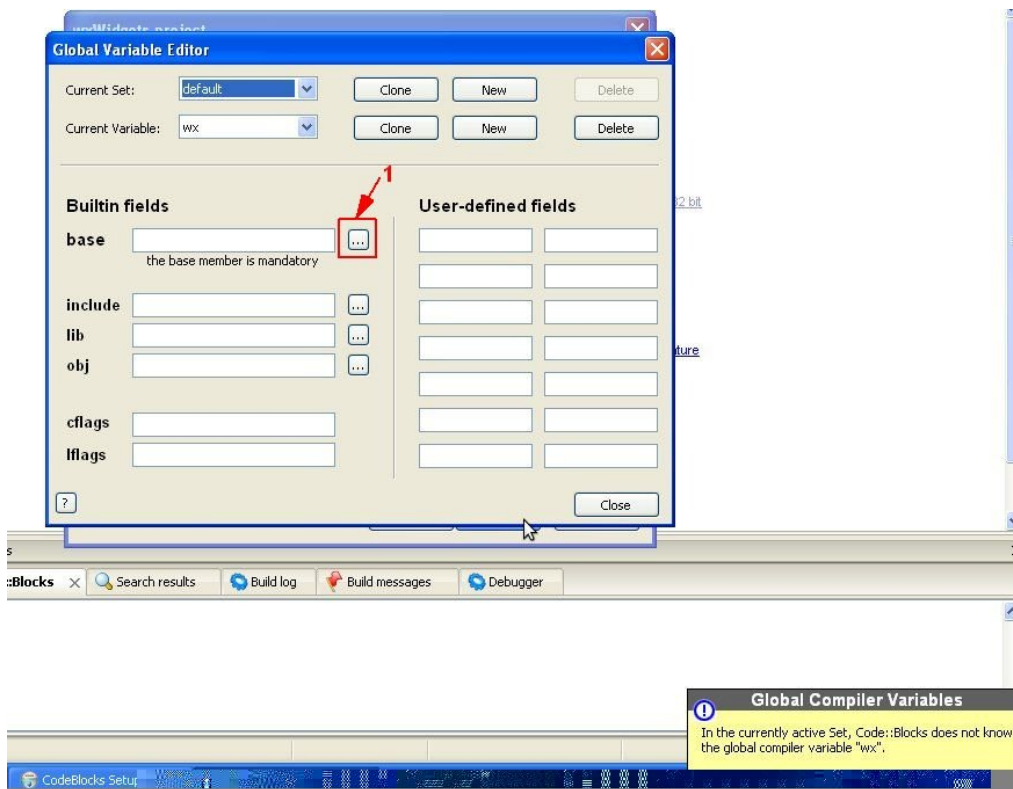


fig. 3 – la primera vez que se crea un proyecto wxWidgets no está definida la carpeta donde están las librerías

Podemos especificar la ubicación en la pantalla de la fig 3 haciendo click en el botón (1) y buscar la carpeta de las wxWidgets.

O desde la pantalla que nos presenta la fig. 4 (procedimiento muy similar al anterior) Hacer click en (1) y buscar la ubicación deseada (2)

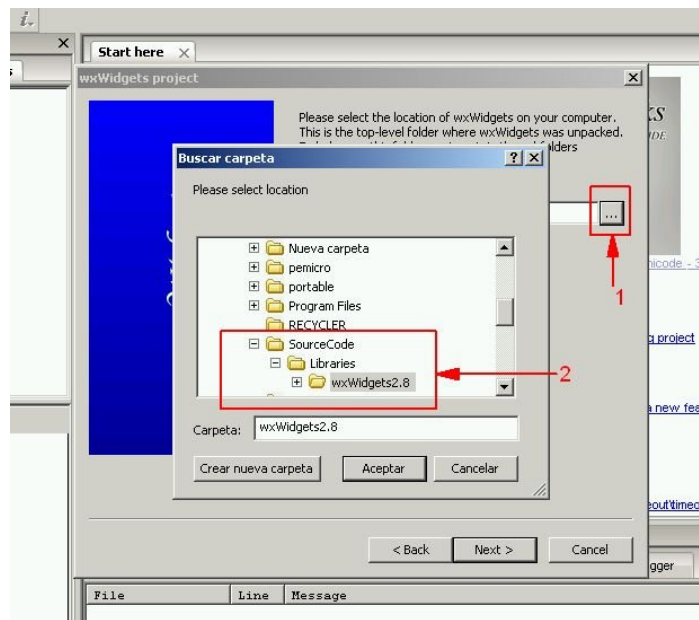


fig. 4 – definir la ubicación de las librerías wxWidgets

Luego hacemos click en “Next” y nos aparece una ventana donde podemos elegir el compilador que vamos a usar; “GNU GCC Compiler” en nuestro caso, que es el que aparece por default. Hacemos click en “Next” y aparece otra ventana, donde podremos seleccionar algunas opciones más para la construcción del proyecto. Por ahora seleccionamos y deseleccionamos hasta que nos quede como la fig. 5

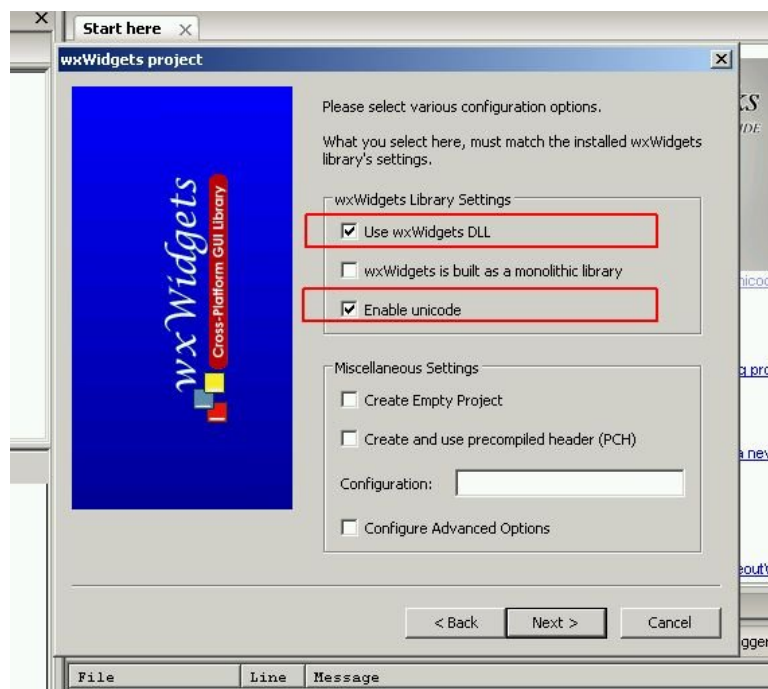


fig. 5 – opciones para la construcción de nuestro proyecto

Al hacer click en “Next” se completa el asistente. Debería aparecer algo similar a la fig. 6

Podemos diferenciar tres zonas: En (1) podemos ver, según la solapa seleccionada, entre otras cosas (depende de las opciones seleccionadas en el menú “view”), los **archivos fuente** (solapa Projects) que componen el/los proyecto/s (.cpp, .h) los **recursos** (solapa Resources) que se van agregando al proyecto (botones, imágenes, etc.)

En (2) nos aparecen por ahora una vista previa de la pantalla que estamos diseñando. Al ir agregando código irán apareciendo otras solapas donde podremos escribir código.

(2.a) es una sección con botones que realizan comandos específicos de wxSmith (El “plug-in” que elegimos anteriormente, que nos permite diseñar la GUI de forma visual)

(2.b) Es la ventana en la que vamos a agregar controles.

(2.c) Es el conjunto de controles disponibles para utilizar en la construcción de nuestra “pantalla”

(2.d) es un botón que nos muestra una “vista previa” de la ventana (solo de la parte gráfica, lo que se muestra no ejecuta nuestro programa)

(3) Es la zona donde podremos cambiar las propiedades de los controles que usemos.

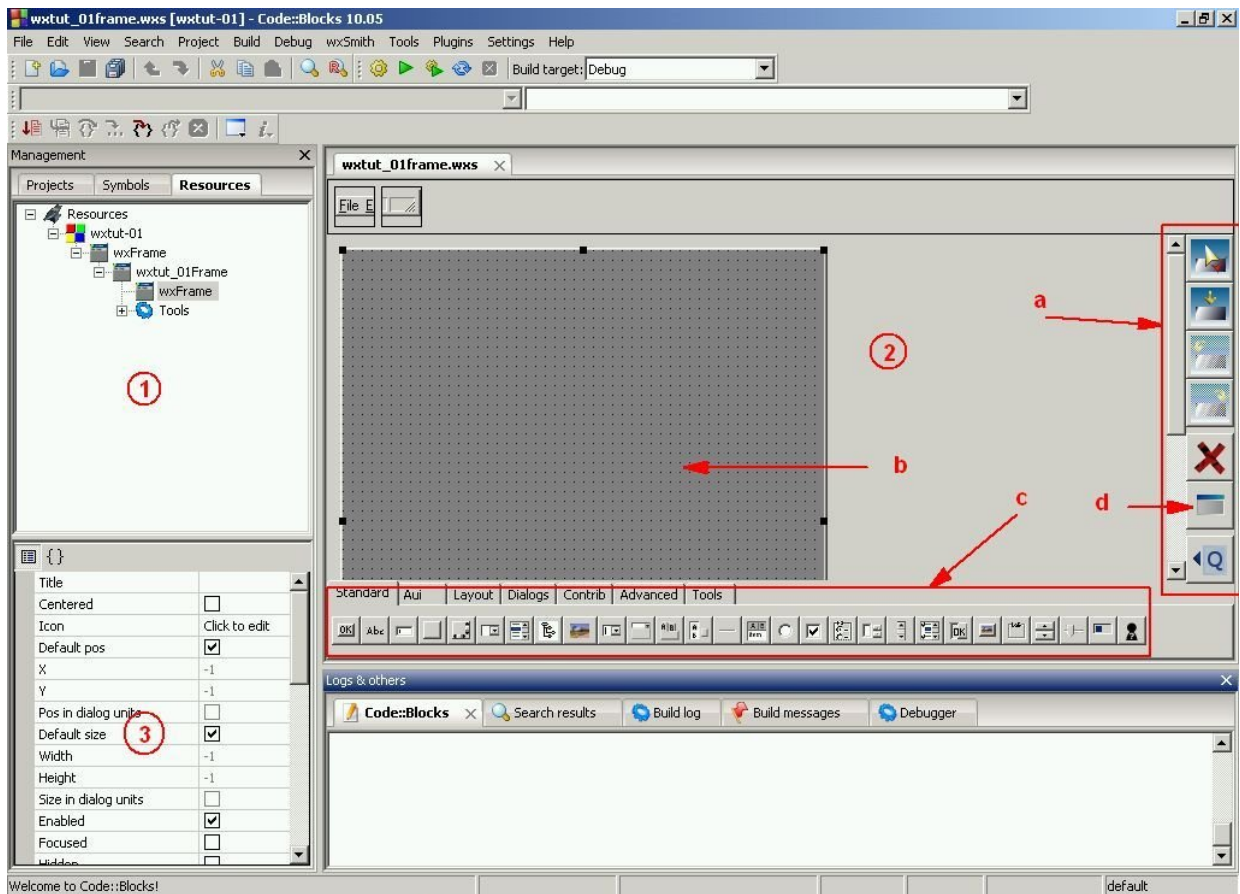


fig. 6 – diferentes zonas de trabajo

Si es la primera vez que se usa wxWidgets, antes de continuar verifiquemos que todo se instaló y funciona correctamente. Hacemos “Build & Run” (F9) y si todo va bien aparece la siguiente pantalla (fig. 7)

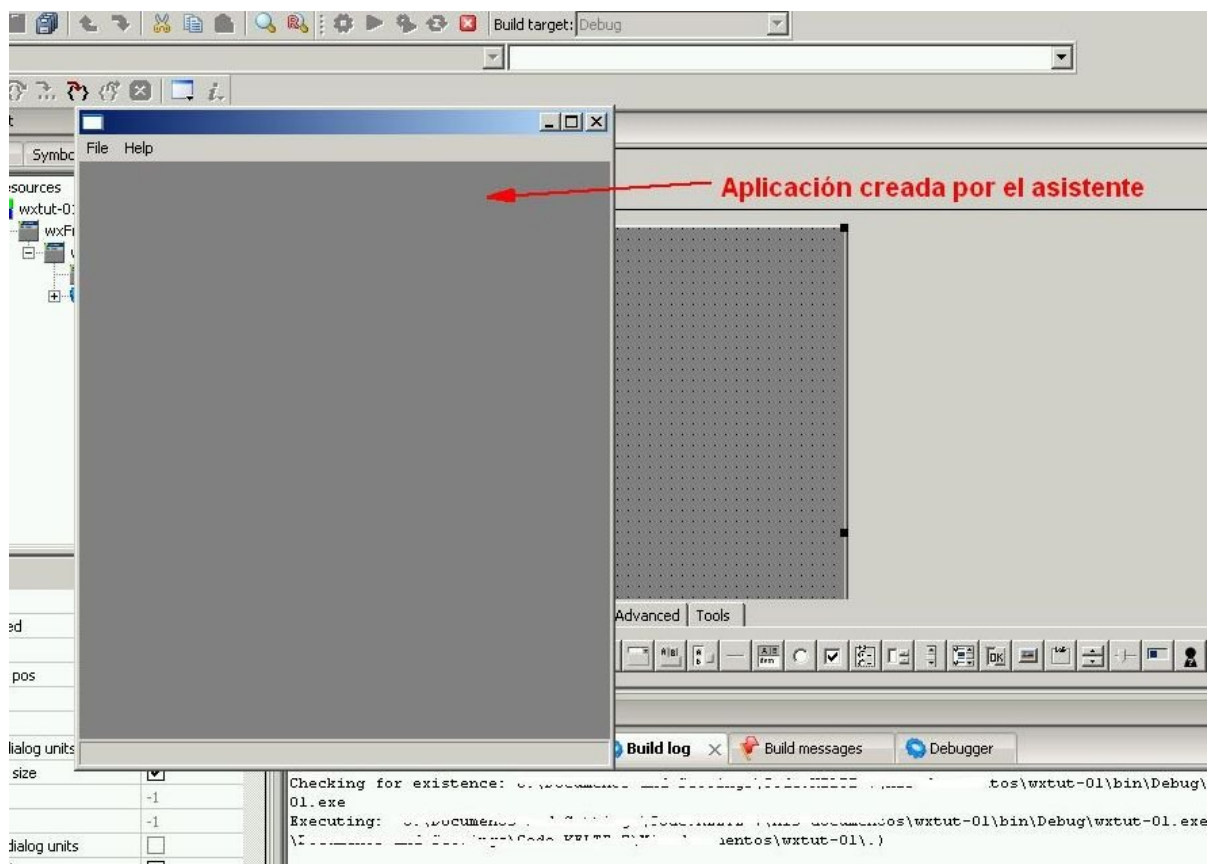


fig. 7 – Ejecutando la aplicación creada por el asistente...

Si apareció la aplicación, tenemos un entorno listo para desarrollar aplicaciones con C++ y wxWidgets.