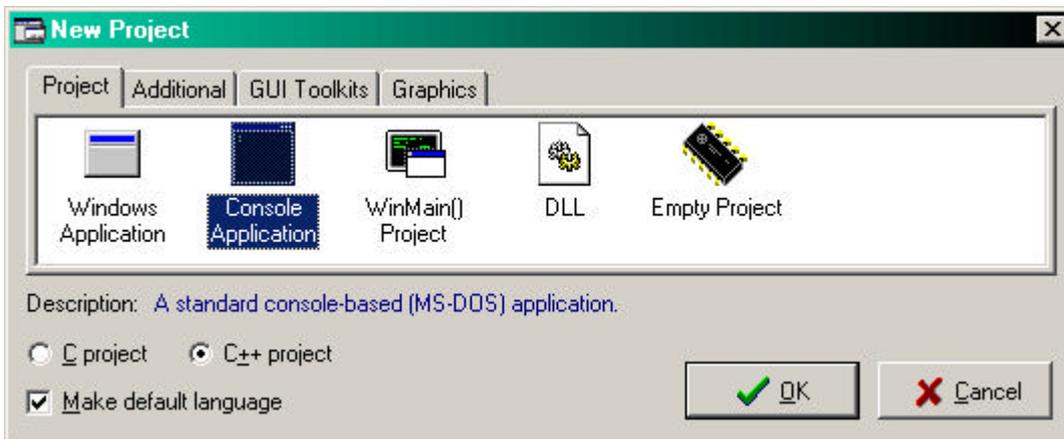


## Entorno de programación de Dev-C++ 4.01

Primeros pasos para comenzar a programar con este entorno:

File→ New Project (CTRL+N)



Después se introduce el nombre del proyecto que se guardará con extensión .dev (p.ej. proyect1.dev)

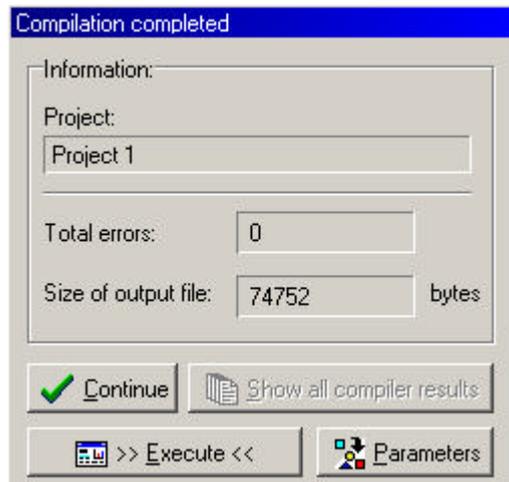
Se genera un fichero llamado *untitled1*, que habrá que renombrar. Para ello habrá que guardarlo con File→Save Unit(CTRL+S). Se recomienda guardar este fichero C++ con la extensión .cpp. (pej. prueba1.cpp)

Se puede modificar el programa para empezar a trabajar con él. Para hacer un programa básico que escriba hola mundo, se puede dejar de la siguiente forma:



```
#include <stdio.h>
#include <stdlib.h>
#include <iostream.h>
int main(int argc, char *argv[])
{
    cout << "hola mundo" ;
    system("PAUSE");
    return 0;
}
```

Para compilarlo se pulsa el botón  o Execute→Compile (CTRL+ F9). Después de esto aparecerá la ventana siguiente, en la que se podrán ver los errores (*Show all compiler results*) o ejecutar el programa si ha sido compilado correctamente (>>Execute<<).



Después de ver como ejecutar un programa, se van a describir las opciones principales de cada menú.

## Menu File

Permite realizar operaciones con ficheros y salir del programa

New Project... Ctrl+N	Nuevo proyecto
New Template... Ctrl+T	---
Open project or file... Ctrl+O	Abrir proyecto existente o fichero
Reopen	Abrir archivos recientes
New Source file Ctrl+U	Fichero nuevo
New Resource file	---
Save unit Ctrl+S	Guardar unidad
Save unit as...	Guardar con otro nombre
Save all	Guardar todos los ficheros abiertos
Close project	Cerrar proyecto
Close Alt+F4	Cerrar ventana actual
Export	Exportar a HTML o RTF (Word)
Print Ctrl+P	Imprimir
Printer Setup...	Setup impresora
Exit Dev-C++	Salir del entorno

## Menú Edit

Acciones que se pueden realizar para las tareas de edición de texto.

 Undo	Ctrl+Z	Deshacer última modificación
 Redo	F4	Rehacer
 Cut	Ctrl+X	Cortar
 Copy	Ctrl+C	Copiar
 Paste	Ctrl+V	Pegar
 Insert		Insertar en código(main,if,for, fecha, comentario...)
 Toggle Bookmarks		Poner marca en el texto (CTRL+0...CTRL+9)
 Goto Bookmarks		Ir a marca (ALT+0..ALT+9)
Select all	Ctrl+A	Seleccionar todo el texto de la ventana actual

Las marcas (bookmarks) se utilizan para realizar accesos rápidos a diferentes partes del código. Por ejemplo, a la zona de declaración de variables, o al procedimiento o método con el que se este trabajando.

## Menú Search

Para hacer búsquedas en el texto

 Find...	Ctrl+F	Buscar
Find next	F3	Buscar siguiente
 Find and Replace...	Ctrl+R	Reemplazar
 Go to line...	Ctrl+G	Ir a una línea del programa

## Menú Project

En este menú se pueden gestionar los elementos que pertenecen a un proyecto

 New unit in project	Ctrl+F1	Nuevo fichero
 Add to project...	Ctrl+F2	Añadir fichero existente al proyecto
 Remove from project...	Ctrl+F3	Eliminar fichero del proyecto
 Edit resource file	Ctrl+F6	--
 Generate MakeFile...		--
 Project options	Alt+P	Opciones del proyecto (nombre, opciones adicionales de compilación, tipo, etc.)

## Menú Execute

 <b>C</b> ompile      Ctrl+F9	Compilar
 <b>R</b> un              F9	Ejecutar
 <b>C</b> ompile and <b>R</b> un      Ctrl+F10	Compilar y Ejecutar
 <b>R</b> ebuild all          Ctrl+F11	Regenerar todos los binarios del proyecto
 <b>D</b> ebug              F8	Ejecutar el Debugger (Ver apdo. de debugging)

## Menú Options

Opciones de compilación y entorno

 <b>C</b> ompiler options	<ul style="list-style-type: none"> <li>-Opciones de compilación (Directorios/ características compilador para C yC++/ Generación de código y optimización y opciones de linkado</li> <li>- Opciones del entorno(Preferencias, Edición, Colores, Completar código y varios)</li> <li>- Tipo de iconos de las barras de herramientas (normal, gnome)</li> </ul>
 <b>E</b> nvironment options	
 <b>I</b> cons style      ▶	

## Menú Tools

Algunas herramientas

 <b>C</b> ompile results      F12	Ventana de resultados de compilación
 <b>T</b> ools configuration...	--
 <b>D</b> os shell              F10	Salir al Dos
 <b>E</b> xplorer              F2	Explorer de Windows
 <b>P</b> ackage Manager      Ctrl+F12	--
 <b>S</b> etup Creator          F11	Para hacer instaladores sencillos de tus programas

## Menú Windows

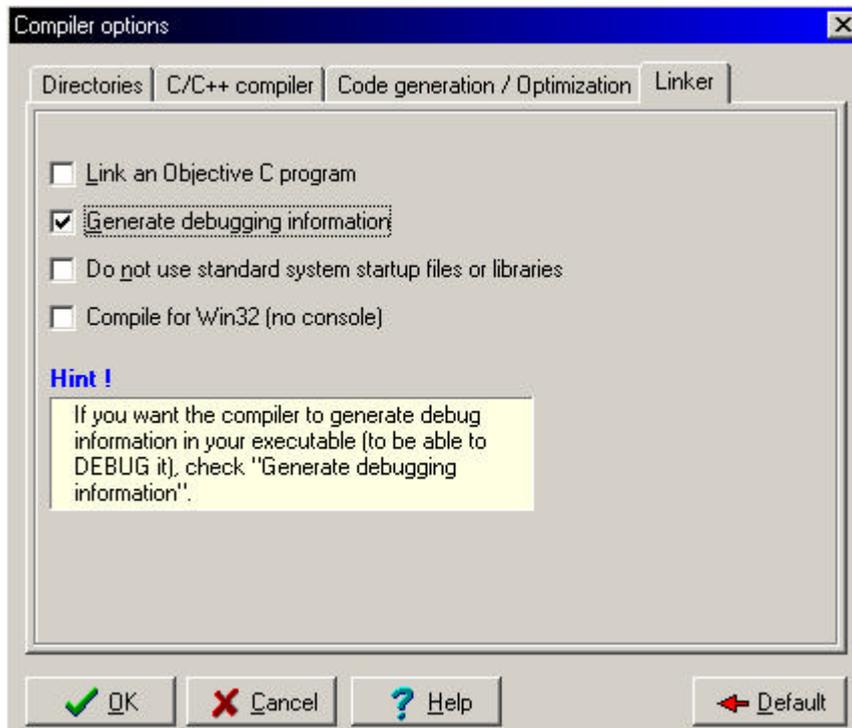
Gestión de ventanas internas al entorno

 Tile	Organizar para ver todas las ventanas
Cascade	Organizar en cascadas
 Arrange icons	--
<hr/>	
 Close All	Cerrar todas la ventanas
 Minimize All	Minimizar todas las ventanas
<hr/>	
 Full screen mode   Alt+Enter	Pantalla completa
<hr/>	
 Next                   F6	Siguiente ventana
 Previous               F5	Ventana anterior
<hr/>	
<input checked="" type="checkbox"/> prueba1.cpp	Lista de ficheros abiertos

Si sólo se quiere ejecutar (F9) , o si que quiere Compilar y ejecutar 

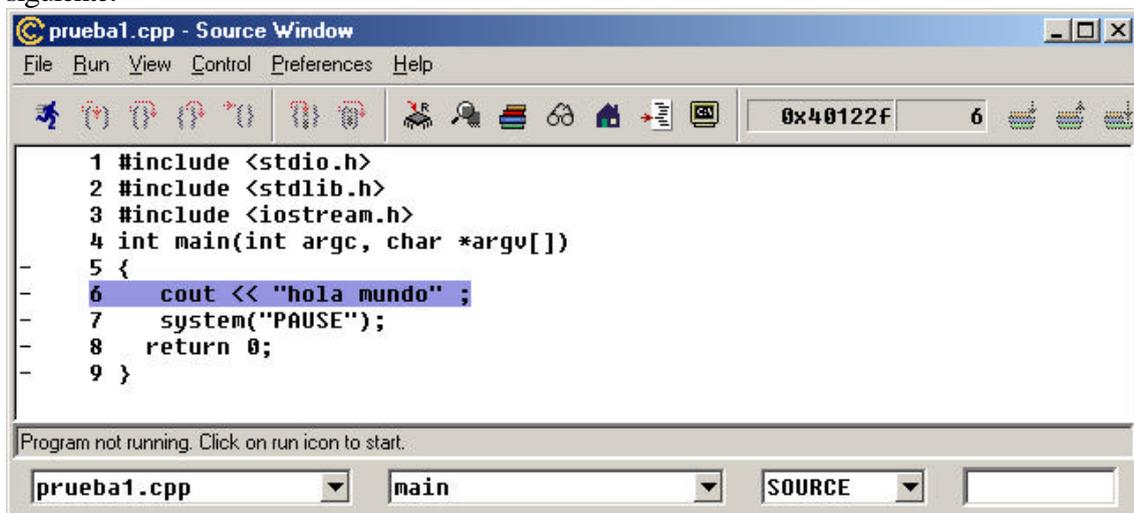
## Debugging

El depurador(debugger) nos permite visualizar la ejecución paso a paso de un programa para corregir errores. Para que funcione el debugger hay que añadir al ejecutable información para el debug. Esto se debe hacer en Options → Compiler Options → Linker, y marcar la casilla *Genarate debugging information*



Con este apartado se pretende dar una ligera idea de cómo funciona el debugger sin entrar en muchos detalles.

El debugger se encuentra en bin/gdb.exe (debajo del directorio de instalación de Dev-C++). Aunque puede activarse pulsando F8 en el entorno Dev-C++. Su aspecto es el siguiente:



Para que empiece a funcionar habrá que abrir un fichero ejecutable (File→Open o CTRL.+O) que haya sido guardado con opciones de debug, como se explicó anteriormente. El funcionamiento básico consiste en comenzar a ejecutar el programa después de cargarlo (Run→Run) e ir ejecutando paso a paso las instrucciones ya sea mediante el menú Control o mediante la barra de herramientas:



Las opciones son (siguiente instrucción- S), (siguiente instrucción, sin entrar en procedimientos-N), (Salir del procedimiento actual para volver al procedimiento llamante- F) (Continuar-C)

Se pueden tener vistas de variables con (View→Watch Expresión CTRL+W)

Ver variable locales con (View→Local Variables CTRL+L)

Ver puntos de ruptura con (View→Breakpoints CTRL+B)

Los puntos de ruptura (breakpoints) son marcas que se asocian a líneas del programa donde se quiere que la ejecución pare para analizar el estado del programa en ese punto. Muy útiles cuando se quieren saltar bucles y zonas del programa que no se necesitan examinar.

Para poner un punto de ruptura es suficiente con 'pinchar' con el ratón en la línea deseada. Esta línea se marcará con un punto rojo a la izquierda.

```
- 5 {  
- 6 cout << "hola mundo" ;  
■ 7 system("PAUSE");  
- 8 return 0;  
- 9 }
```