Programación

Entorno de Programación Dev-C++

.

1. Introducción al entorno Dev-C++.

Dev-C++ es un entorno de desarrollo integrado (IDE por sus siglas en inglés) para programar en lenguaje C/C++

Un entorno de desarrollo integrado o IDE es una aplicación que aglutina en un único paquete el editor de textos, el compilador, el enlazador y el depurador. Además se encarga de organizar los distintos ficheros de código fuente de que constan normalmente los programas, agrupándolos en una estructura denominada "proyecto" que se compone de todos los elementos necesarios para la generación del programa ejecutable final.

Dev-C++ es un entorno agradable, compacto, permite crear programas fácilmente en C y su instalación y manejo es muy simple. En la actualidad se ofrece la versión 5 beta, y es dicha versión la que se encuentra disponible en las aulas de laboratorio.

Trabajar empleando un entorno de desarrollo ofrece diversas ventajas:

1 Facilidad de uso. La generación de código es sencilla e intuitiva debido al interfaz gráfico.

2 Sencilla visualización de los errores de compilación (p. ej. si el compilador detecta un error en la línea 107, el entorno del desarrollo destaca la línea correspondiente al error).

3 Herramientas de depuración de código que nos permiten limpiar los errores que se den en tiempo de ejecución. (p.ej. podemos observar paso a paso como cambian los valores de nuestras variables).

Este entorno de desarrollo permite crear aplicaciones de diferentes tipos (ejecutables, librerías, etc) empleando el lenguaje C o el lenguaje C++. Para esta asignatura se desarrollarán únicamente **aplicaciones de consola** en el **lenguaje C**.

1.1. Apartado de ayuda.

En el menú desplegable de la ayuda existe un apartado llamado "Ayuda en Dev-C++" donde es posible consultar información útil tanto de la sintaxis del lenguaje C (Ayuda > Ayuda en Dev-C++ > An Introduction to C Programming) como del propio entorno de desarrollo (Ayuda > Ayuda en Dev-C++ > Dev-C++ 5).

2. Pasos para desarrollar una aplicación con el entorno de desarrollo Dev-C++.

El proceso de desarrollo de una aplicación consta de varios pasos:

1. Creación del proyecto en el que se va a trabajar.

   En esta fase se establece el tipo de aplicación que se va a implementar y el lenguaje que se va a emplear.

2. Escritura del código fuente.

   Se escribe el programa en lenguaje C y se genera el fichero de código fuente

3. Compilación y *linkado*.

   Se compila el código fuente y se *linka* para generar el ejecutable. Se generan los demás ficheros del proyecto.

4. Corregir errores en tiempo de diseño (detectados por el compilador), si los hay.

   Si el compilador observa algún error en el código, generará un mensaje. Entonces será necesario corregirlo y volver a compilar y *linkar*.

5. Ejecutar el programa.

   Ejecutar el programa y comprobar que su comportamiento es el deseado.

6. Corregir errores en tiempo de ejecución.

   Si durante las pruebas del programa se comprueba que no se comporta como se había planeado, es necesario localizar y corregir el problema. A veces es suficiente con revisar el código. En programas más largos puede ser útil emplear el depurador.

2.1. **Creación el proyecto en el que se va a trabajar.**

2.1.1 Proyectos en Dev-C++

Dev-C++ organiza las aplicaciones que se van a desarrollar en estructuras denominadas proyectos. Cada uno de los proyectos contiene una serie de ficheros que pertenecen única y exclusivamente a ese proyecto.

No es necesario que exista un directorio para cada uno de los diferentes proyectos, sino que al generar un nuevo proyecto Dev-C++ solicitará que se indique la ubicación donde colocar sus ficheros. Aunque es posible ubicar todos los proyectos en un mismo directorio no es un hábito recomendable, pues el compilador nombra por defecto al fichero fuente como "main.c", y es muy fácil sobrescribir dicho fichero cada vez que se cree un nuevo proyecto si no se es suficientemente cuidadoso.

En los proyectos que se crearán para esta asignatura, habrá un fichero con información del proyecto, otro con el código fuente en C, otro con el código en máquina y por último uno con las instrucciones de *linkado* para nuestro programa.

2.2.1 Tipos de proyectos

El tipo de proyecto especifica qué tipo de aplicación se desea crear. Para cada tipo de proyecto se establecen las opciones que el entorno de desarrollo usa para crear el programa ejecutable (opciones del compilador, librerías que usa el *linkador*, etc)

En esta asignatura aprenderá a hacer proyectos de tipo consola, es decir programas que se ejecutan en una única ventana sin gráficos. El motivo de elegir estos programas es que son los más fáciles de programar.

La interacción con un programa de tipo consola se hace leyendo los mensajes en la consola y escribiendo caracteres en la consola, a través de comandos de lectura y escritura.

Se llaman de consola porque su apariencia es la de los programas que se ejecutan en una consola conectada a un ordenador mainframe.

2.3.1 Pasos para crear un proyecto de tipo consola escrito en C.

a. Iniciar el entorno de desarrollo Dev-C++

En las aulas de los laboratorios el compilador Dev-C++ se encuentra ubicado típicamente dentro de las carpetas "DevCpp" o "BloodShell DevCpp" tal y como puede apreciar en la Figura 1.

Si desea instalar Dev-C++ en su ordenador personal, puede descargarlo libremente desde la URL: http://prdownloads.sourceforge.net/dev-cpp/devcpp-4.9.9.2_setup.exe

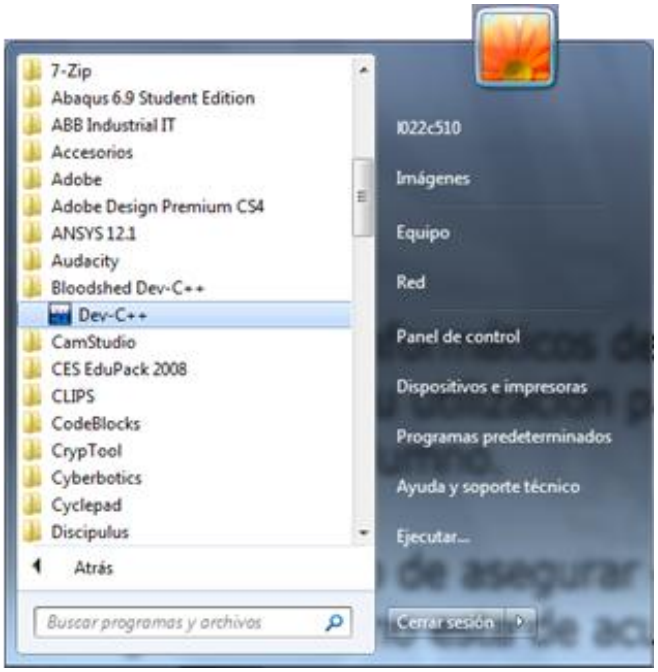

Figura 1. Ubicación de Dev-C++ en las aulas del laboratorio.

b. Crear un proyecto

Para crear un programa en Dev-C++ lo primero que hay que hacer es crear un proyecto que lo contenga. A continuación se describen los pasos necesarios para crearlo.

Seleccione Archivo > Nuevo > Proyecto. Verá como aparece la ventana de Nuevo proyecto (Figura 2). Puede observar que aparecen las distintas opciones de proyectos disponibles, debe elegir como tipo de proyecto "Console Application". El siguiente paso será elegir un nombre para el proyecto, dicho nombre se introducirá en la casilla "Nombre del Proyecto".

Por último se marcará el lenguaje en el cual se implementará el programa. Para esta asignatura sólo se van a desarrollar programa en lenguaje C, por ello se elegirá la opción "En C". Si desea que el compilador recuerde su elección del lenguaje en futuros proyectos puede marcar la casilla de "Hacer C/C++ Mi Idoma".

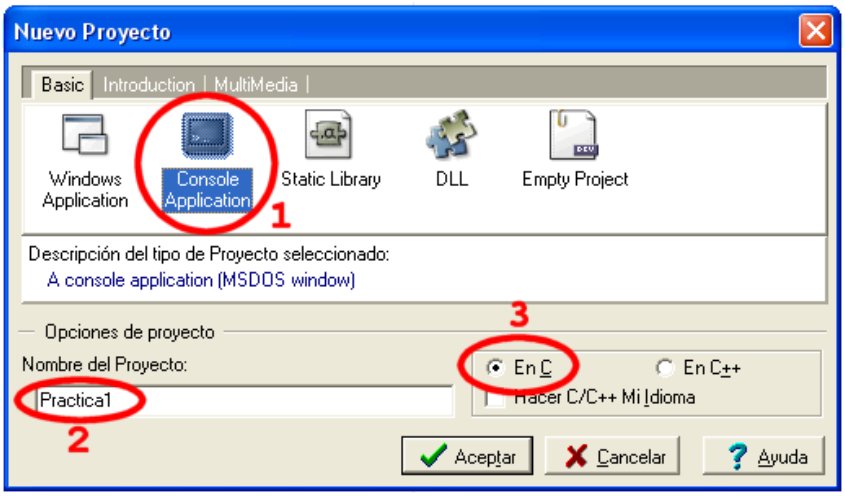

Figura 2. Ventana para la creación de un nuevo proyecto.

Al pulsar "Aceptar" aparecerá una ventana donde se le solicitará introducir la ubicación de los ficheros del proyecto. Recuerde que no es recomendable almacenar los ficheros de diferentes proyectos en un mismo directorio.

Al pulsar "Guardar" verá que el entorno de desarrollo genera un primer fichero tipo de código fuente con las sentencias básicas de un programa en C. Dicho fichero esta nombrado temporalmente como "main.c", pero hasta que no se compile o salve el programa no se almacenará una copia en el directorio del proyecto.

El entorno de desarrollo genera, para los proyectos de tipo consola, un único fichero para el código fuente (source files), con extensión .c. (Si elige como tipo de proyecto la opción aplicación en blanco -Empty Project-, comprobará como dicho primer fichero de código fuente no se genera, sino que deberá incluirse posteriormente seleccionando Archivo > Nuevo > Código Fuente)

Nota: La sentencia 'system("PAUSE");' es incluida de forma automática por el entorno de desarrollo para permitir al usuario observar la consola antes de terminar la ejecución del programa.

Al llegar a este punto ya se ha creado el proyecto y la apariencia de la pantalla será la indicada en la Figura 3.

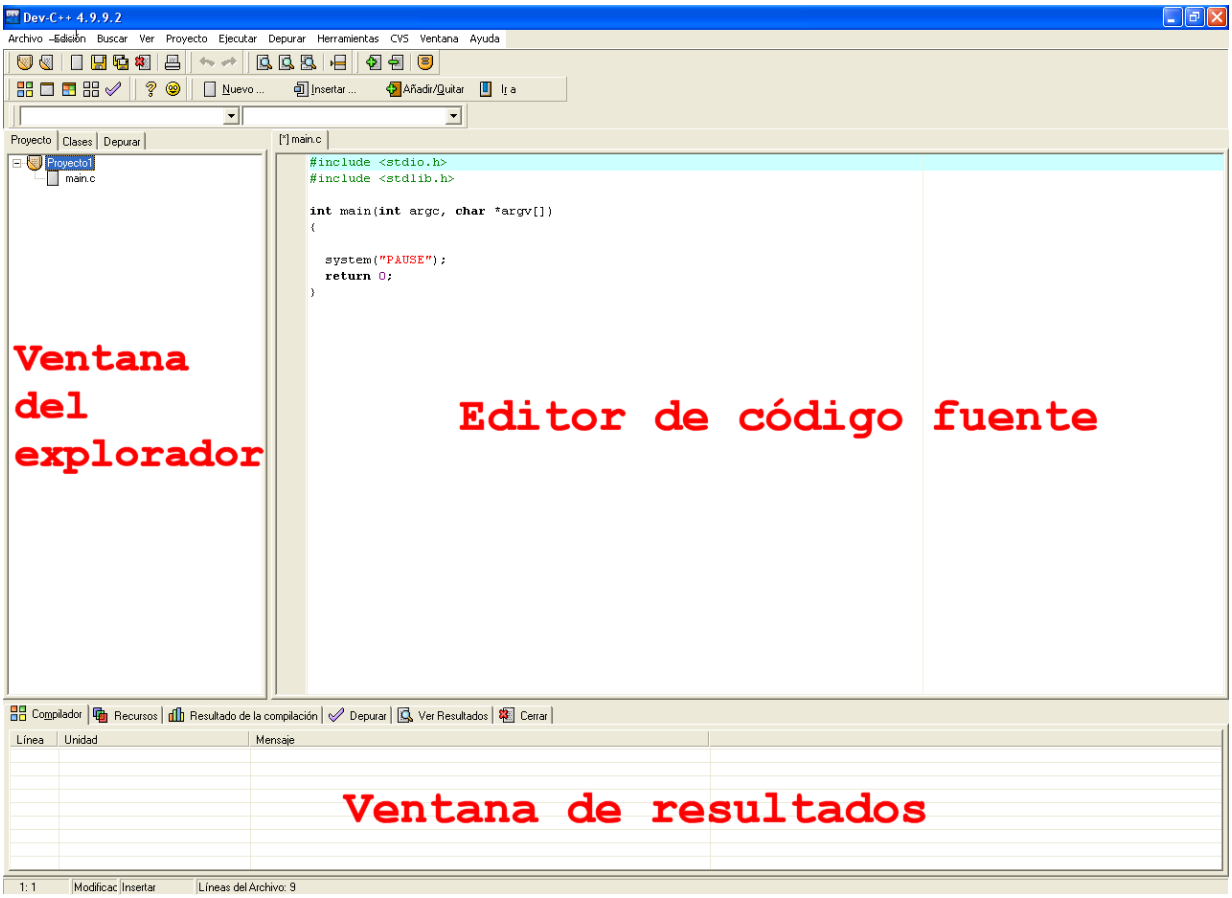

Figura 3. Dev-C++ en el inicio de un nuevo proyecto.

Como ve, se estructura la pantalla en tres ventanas principales: la ventana del explorador, la ventana con las pestañas de resultados y la ventana principal, con el código fuente en C. El tamaño de estas ventanas puede ser modificado por el usuario (y también se pueden minimizar).

En la ventana del explorador se puede observar el nombre del proyecto en el que se está trabajando y los ficheros que contiene. La pestaña "Proyecto" típicamente solo contendrá el fichero con código fuente la de aplicación. En este mismo panel hay otras dos pestañas. En la pestaña "Clases" aparecerán las funciones del programa mientras que en la pestaña "Depurar" pueden añadirse las variables que se deseen someter a un seguimiento durante un proceso de depuración.

La ventana de resultados es donde se mostrarán, como verá más adelante, resultados de las diferentes acciones del programador, como por ejemplo los errores detectados por el compilador, las directivas empleadas al compilar o los comandos para desarrollar la depuración.

c. Ficheros generados

Puede comprobar viendo el disco duro que al crear un nuevo proyecto ningún fichero es generado. Solo cuando se guarde el fichero fuente se generará su correspondiente fichero

con extensión .c. Y sólo cuando se compile la aplicación se generará la serie de ficheros pertenecientes al proyecto. Dichos ficheros se almacenarán todos en el mismo directorio, el correspondiente al proyecto en curso.

La siguiente tabla define los ficheros que existen en Dev-C++ para cada proyecto:

Fichero	Extension	Descripcion
Fichero del proyecto	.dev	Almacena información sobre la configuración del proyecto.
Makefile	.win	Fichero necesario para el proceso de compilación. Gestiona las dependencias y almacena las instrucciones de *linkado* para nuestro programa.
Fichero fuente C	.c	Contiene el código fuente
Fichero código objeto	.o	Fichero con el código en lenguaje máquina o bytecode resultante de la compilación del código fuente. Existe un archivo en código objeto por cada uno de nuestros archivos en código fuente.
Fichero ejecutable	.exe	Fichero ejecutable de la aplicación.

Tabla 1. Ficheros existentes en un proyecto de Dev-C++.

2.2. **Escritura del código fuente en C.**

Una vez generado el proyecto puede comenzar a escribir el código fuente, en el fichero .c .

En este punto se recomienda emplear una configuración de colores clásica en el editor (Herramientas → Opciones del Editor → Sintaxis → Pre-configuraciones: Classic) y activar las ayudas gráficas para el balanceo de paréntesis y llaves (Herramientas → Opciones del Editor → Principal → √ Resaltar llaves y paréntesis concordantes).

El editor de texto del entorno de desarrollo cambiará el color de lo que escriba para hacer más sencilla la tarea. Si se opta por el esquema de colores clásico (por defecto):

- Los comentarios se mostrarán en azul claro
- Las bibliotecas empleadas en verde
- Las cadenas de texto en rojo
- Los elementos del lenguaje C en negro

Por ejemplo el programa "Hola Mundo" se mostrará tal y como ve en la Figura 4.

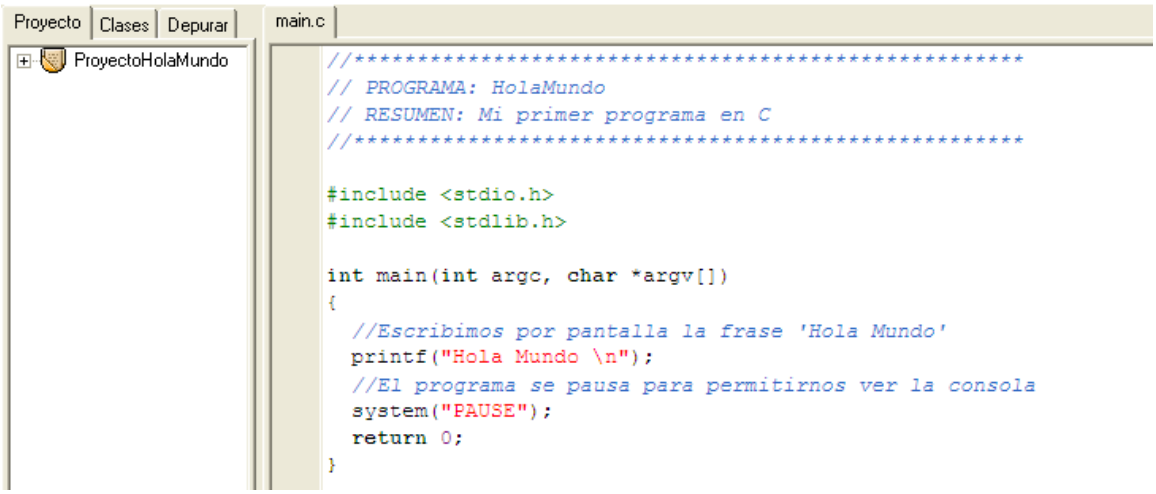

Figura 4. Programa “Hola Mundo” en Dev-C++.

Ejercicio 1. Escribir el código fuente del programa HolaMundo en C. Abrir el fichero creado (.c) en un editor de texto (p.ej. notepad) y ver que aspecto tiene.

2.3. Compilar y *linkar*

Para ver como funciona su programa debe compilarlo, *linkarlo* y ejecutarlo. En el entorno de desarrollo Dev-C++ la combinación de compilar y *linkar* se ejecuta pulsando el botón Compilar (Ctrl + F9).

Aparecerá una ventana que mostrará mensajes con información acerca del proceso de compilación. Si no hay errores en dicha ventana aparecerá la frase “Done” y en la pestaña “Resultado de la compilación” de la ventana de resultados podrá ver información sobre el proceso de compilación y comprobar que el mismo ha concluido con éxito. Tras completar la compilación puede comprobar como se han creado los ficheros anteriormente mencionados en la Tabla 1.

Ejercicio 2. Compile el programa “Hola Mundo” y compruebe qué se han generado todos sus ficheros, incluyendo el ejecutable. Compruebe también que haciendo doble clic sobre el fichero .exe, es posible ejecutarlo fuera del entorno de desarrollo.

2.4. Corrección de errores en tiempo de diseño

Se llaman errores en tiempo de diseño a los errores que son detectados por el compilador. Para corregirlos es siempre recomendable empezar por el primer error detectado y volver a compilar, pues es posible que los siguientes errores detectados no sean realmente errores, sino elementos que el compilador no ha sabido interpretar como consecuencia del primer error.

Dev-C++ marca con un subrayado rojo la línea del código fuente en la que se ha detectado el error. En la pestaña “Compilador” de la ventana de resultados puede ver la descripción del error para una mayor claridad y en la pestaña “Resultado de la compilación” de la misma ventana el mensaje de error emitido por el compilador.

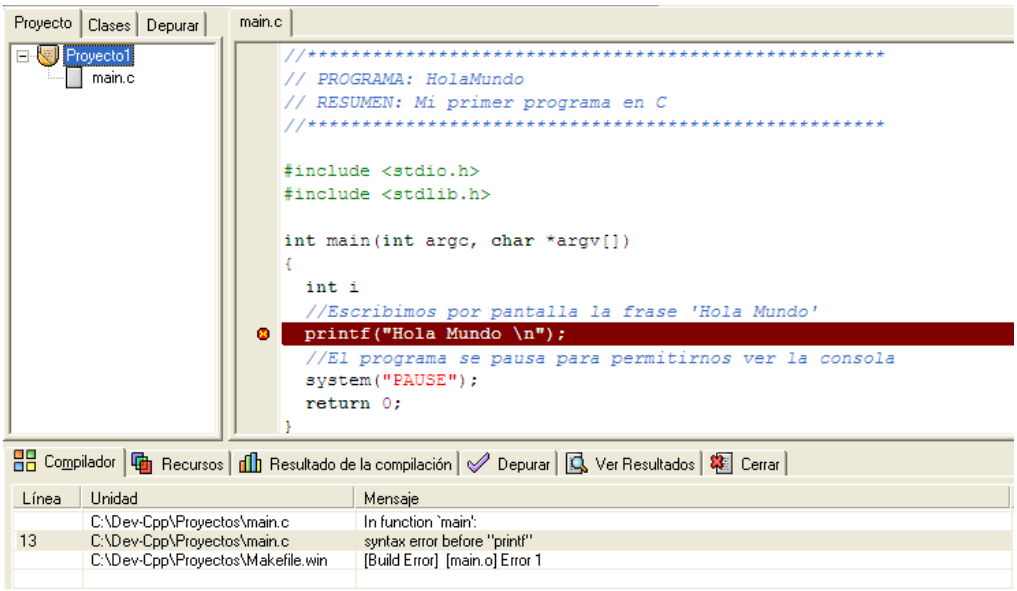

Figura 5. Error detectado por el compilador en Dev-C++.

Errores típicos que se pueden dar son: no seleccionar correctamente el tipo de proyecto –consola-, no seleccionar correctamente el tipo de lenguaje –C-, o error en la inclusión de las bibliotecas (sintaxis incorrecta o falta de alguna). Esto se debe tener en cuenta a la hora de analizar los mensajes de error de compilador y *linkador*.

2.5. Ejecución del programa

El resultado del *linkado* es el programa ejecutable. Para ejecutarlo puede pulsar el botón Ejecutar (Ctrl + F10) y verá como aparece la consola donde el usuario puede interactuar con el programa. Otra opción es buscar el fichero ejecutable (con extensión .exe) en la carpeta del proyecto y hacer doble clic sobre él.

Nota: Es posible compilar y ejecutar el programa en un solo paso si se pulsa el botón Compilar y Ejecutar (F9).

Ejercicio 3. Introducir un error en el programa HolaMundo (por ejemplo escribir la palabra reservada printf de modo incorrecto, o quitar una de las comillas dobles de la cadena de caracteres). Compilar y ejecutar el programa (comprobar que puede hacerse todo el proceso simplemente pulsando el botón de Compilar y Ejecutar). Observar el mensaje o mensajes de error que se generan.

2.6. Corrección de errores en tiempo de ejecución.

Cuando se crean programas con una cierta complejidad es frecuente que tengan distintos tipos de errores. El compilador es capaz de detectar errores de sintaxis, pero es posible que el programa sea correcto desde ese punto de vista pero no se comporte como había planeado.

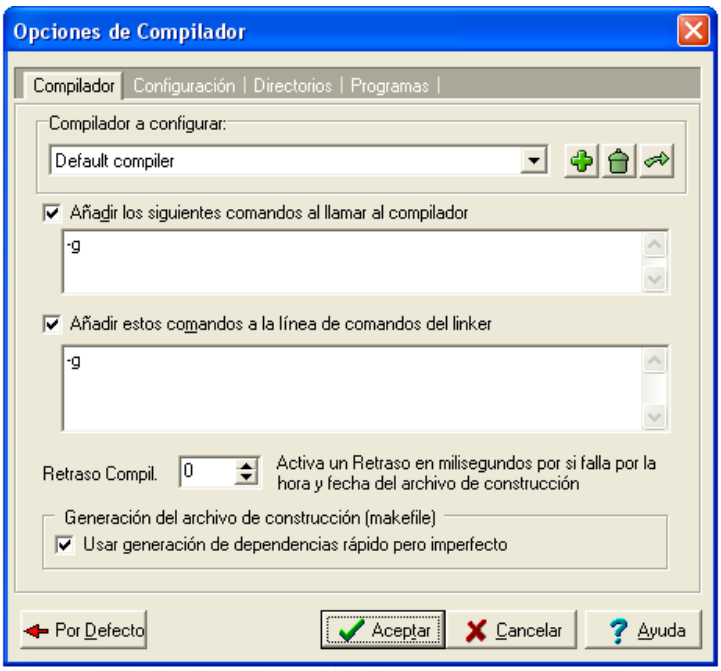

Figura 6. Directivas para incluir información de depuración.

Para localizar estos errores es muy útil usar un depurador como el que incorpora el entorno de desarrollo Dev-C++. Para ello es recomendable modificar primero las opciones del compilador (Herramientas → Opciones del Compilador) para que se incluya información de depuración en el proceso. Incluyendo el parámetro "-g" (tal y como se indica en la Figura 6) en la línea de comandos del compilador y del *linker* se indica al entorno de desarrollo que se incluya dicha información de depuración.

Después ya es posible ejecutar el programa en modo depuración pulsando el botón Depurar (F8), lo que permitirá ir ejecutando el programa paso a paso, establecer puntos de parada (llamados puntos de ruptura) en los que se detenga el programa o ver el valor que toma una variable en un determinado momento.

Puntos de ruptura o interrupción

Pulsar Ctrl+F5, pinchar con el botón derecho del ratón seleccionando "Añadir/Quitar Punto de Ruptura" o bien pinchar sobre la barra vertical gris que se encuentra al lado izquierdo del código fuente nos permite poner o quitar un punto de ruptura.

Añadir un punto de ruptura hará que se marque la instrucción, quedando la línea de código fuente subrayada en rojo claro. Esta acción obligará al depurador a detenerse en esa línea de código fuente haciendo que la ejecución del programa se detenga.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  //Escribimos por pantalla la frase 'Hola Mundo'
  printf("Hola Mundo \n");
  printf("Frase1 \n"); //punto de ruptura
  printf("Frase2 \n");
  printf("Frase3 \n"); //punto de ruptura activo

  //El programa se pausa para permitirnos ver la consola
  system("PAUSE");
  return 0;
}
```

Figura 7. Puntos de ruptura en Dev-C++.

Pulsar el botón Depurar (F8) cuando se tiene marcado uno o más puntos de ruptura permite que se ejecute la aplicación hasta que se alcance unos de dichos puntos. A partir del punto de ruptura puede avanzar paso a paso por instrucciones o continuar la ejecución normal. En la pestaña "Depurar" de la ventana de resultados puede ver las instrucciones de depuración que puede utilizar en Dev-C++. Como comprobará es posible avanzar paso a paso (instrucción a instrucción), saltar pasos o detener la ejecución.

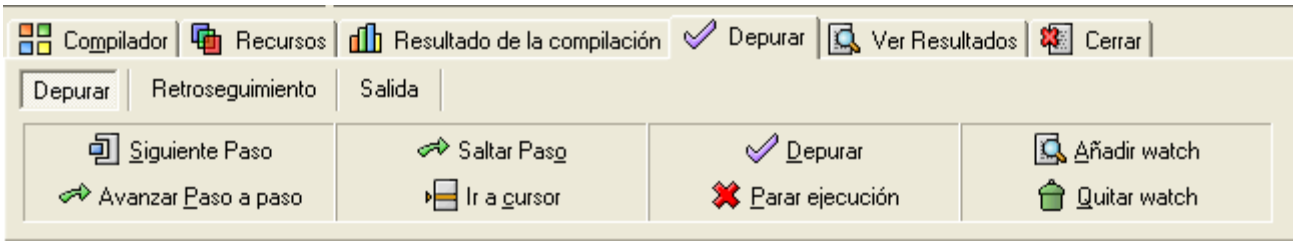

Figura 8. Instrucciones de depuración en Dev-C++.

En cada momento, la instrucción en la que se encuentra el programa aparece resaltada en color azul.

Nota: Es posible iniciar una depuración sin incluir puntos de ruptura pulsando el icono "Ir al cursor". En ese caso el depurador considerará la línea donde este situado el cursor en ese momento como un punto de ruptura.

Ejercicio 4. Modifique el programa HolaMundo añadiendo otras tres instrucciones de escritura en pantalla. Colocar un punto de ruptura en la primera de ellas. Inicie el programa en modo depuración y comprobar que se detiene en el punto de ruptura. Avanzar paso a paso por instrucciones y comprobar cómo tras la ejecución de cada instrucción se muestra el correspondiente mensaje en la consola.

Inspección del valor de una variable

Es posible ver el valor que toma una variable en un momento de la ejecución de un programa. Para ello se debe detener la ejecución (con un punto de interrupción o empleando "Ir hasta el cursor"). En ese momento podrá ver el valor de la variable, seleccionando "Añadir watch" (Depurar > Añadir watch, F4 o pulsando el icono "Añadir watch" de la pestaña "Depurar" de la ventana de resultados) y escribiendo el nombre de la variable, o bien haciendo doble clic sobre el nombre de la variable.

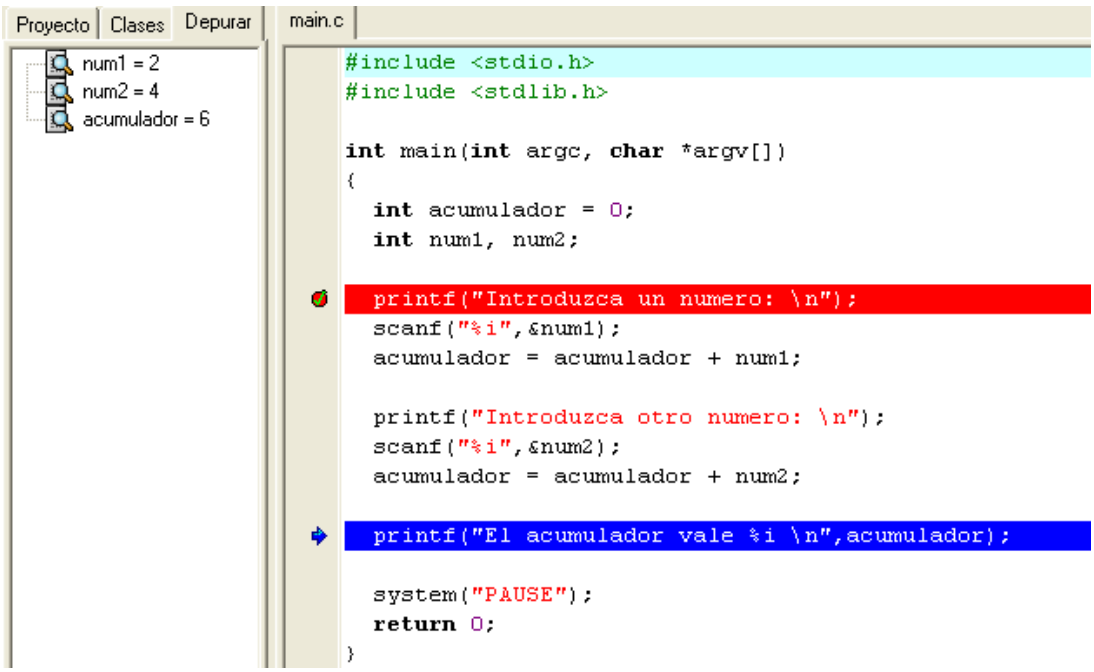

Figura 9. Inspección de variables en Dev-C++.

En ese momento observará como las variables añadidas figuran en la pestaña "Depurar" de la ventana del explorador. Si continua la ejecución del código verá como el valor de dichas variables se va modificando instrucción a instrucción.

Ejercicio 5. Revise el código de la Figura 9 y los valores de las variables. ¿Se comportan cómo esperaba?. Ejecute el código en modo depuración y observe como cambia el valor de las variables.

2.7. Pasar a un nuevo programa

Para escribir un nuevo programa, puede repetir todos los pasos que se han descrito (crear el proyecto con su fichero de código fuente), o bien, reutilizar el proyecto y fichero fuente que ya ha creado, y sustituir el código. Pulsando con el botón derecho del ratón sobre el nombre de su proyecto en la pestaña "Proyecto" de la Ventana del explorador de proyectos puede agregar ficheros en blanco a su proyecto.

Ejercicio 6. Escriba un programa que pida dos números al usuario, los multiplique y muestre luego el resultado por pantalla. Agregue todas las variables del programa a la ventana de inspección y ejecute paso a paso para ver como se van modificando sus valores. ¿Qué valor tienen los números antes de ser leídos?