



El Entorno Integrado de Desarrollo Dev-C++

(Ayuda básica para las primeras sesiones de prácticas)

➤ Sobre Dev-C++

Dev-C++ es un Entorno Integrado de Desarrollo para el lenguaje de programación C/C++ que usa Mingw (Minimalist GNU for Windows, www.mingw.org) de GCC (GNU Compiler Collection www.gnu.org/home.es.html). Es un software de libre distribución (www.bloodshed.net) sujeto a los términos de la Licencia Pública General (GPL) de GNU.

Algunas de las características de Dev-C++ son:

- Soporta compiladores basados en GCC, por ejemplo Mingw.
- Tiene integrado un depurador basado en GDB (Gnu DeBugger).
- Mantiene una lista con las clases utilizadas durante la edición de un programa.
- Mantiene una lista de las funciones definidas en la implementación del programa.
- Tiene un manejador de proyectos.
- Soporta la actualización del software y bibliotecas a través de Internet.

➤ Antes de trabajar con Dev-C++

Debemos tener un acceso directo al fichero **devcpp.exe**, que nos permita la ejecución del entorno de trabajo tras hacer un doble click en el mismo.

Antes de trabajar con Dev-C++ debemos personalizarlo al menos con las siguientes opciones:

- **Herramientas → Opciones del Entorno → Interfaz → Idioma: Español (Castellano)**
- **Herramientas → Opciones de Compilador → Configuración → Compilador de C → Soporte para Programas ANSI Estándar en C: Yes.**
- **Herramientas → Opciones de Compilador → Configuración → Linker → Generar Información de Debug: Yes.**
- **Herramientas → Opciones del Entorno → Principal → Crear archivos de respaldo (Copia de Seguridad).**
- **Herramientas → Opciones del Editor → Principal → Resaltar llaves y paréntesis concordantes.**
- **Herramientas → Opciones del Editor → Sintaxis → Pre-configuraciones: Classic.**



➤ Cómo Editar un programa

Un fichero fuente nuevo:

- **Archivo → Nuevo → Código Fuente** , o bien
- **Ctrl+N**

Un fichero fuente ya existente:

- **Archivo → Abrir Proyecto o Archivo** , o bien
- **Ctrl+O**

Un fichero fuente ya existente, y que se editó recientemente (Dev-C++ guarda un historial de los últimos 10 ficheros fuente distintos que se han abierto):

- **Archivo → Abrir Archivo Reciente**

➤ Cómo Guardar un Fichero Fuente

- **Archivo → Guardar**, o bien
- **Ctrl+S**

Se recomienda que durante el proceso de construcción del programa se guarde el código con relativa frecuencia.

➤ Cómo Compilar un programa

- **Ejecutar → Compilar** , o bien
- **Ctrl+F9**

Si todo ha ido bien, obtenemos un fichero con el código objeto de nuestro programa, pero *¡¡ Aún no hemos obtenido el fichero ejecutable !!*.

Si quisiéramos compilar y después obtener el fichero ejecutable deberíamos realizar los siguientes pasos:

- **Ejecutar → Compilar y Ejecutar**, o bien
- **F9**

Tras la compilación, los resultados de la misma podremos verlos en las pestañas **Resultado de la compilación** y **Compilador**.



- **Detección de Errores.**

Cuando la fase de compilación no ha terminado con éxito, en la parte inferior de la ventana principal aparece activada la pestaña **Compilador**, con tres columnas:

- **Línea:** Indica la línea donde el compilador detectó el error sobre el fichero fuente. Realmente el error puede estar en dicha línea o en alguna de las anteriores. Haciendo doble click en esta fila de la pestaña Compilación, nos posicionaremos sobre la línea del fichero fuente donde el compilador detectó el error.
- **Unidad:** Indica la ubicación del fichero en la unidad de almacenamiento.
- **Mensaje:** Indica el tipo de error detectado por el compilador.

En esta ventana también aparece una fila que no tiene asociado número de línea, y que da información sobre la función en la que se produjo el error.

Después de corregir un error, debemos guardar el archivo y volver a compilar. Esto lo haremos siempre, aunque aparezcan más errores en la ventana, ya que es muy posible que el resto de los errores sean consecuencia del primero.

➤ **Cómo Ejecutar un programa.**

Para ejecutar un programa, **previamente compilado**, realizaremos lo siguiente:

- **Ejecutar → Ejecutar**, o bien
- **Ctrl+F10**

Para realizar compilación y ejecución en un solo paso:

- **Ejecutar → Compilar y Ejecutar**, o bien
- **F9**

La ejecución del programa se realizará en una ventada de MS-DOS emergente.

Por defecto, el fichero ejecutable se crea en el mismo directorio donde se almacenó el fichero fuente.



➤ Cerrar ventanas de edición

- **Archivo** ➔ **Cerrar** (cierra el documento que se encuentra activo), o bien
- **Alt+F4**, o bien
- Pulsar sobre el editor que se desea cerrar con el botón derecho del ratón y elegir la opción **Cerrar**.
- Si tuviésemos activados varios documentos y queremos cerrarlos todos, basta con seguir cualquiera de los procesos anteriores y seleccionar **Cerrar Todo**.

➤ Cómo Depurar programas

Un Depurador nos permite ejecutar paso a paso un programa, viendo en cualquier momento el valor de las variables que más nos interesen. Por lo tanto es una herramienta muy útil para detectar errores lógicos que de otro modo resultarían muy difíciles de localizar.

Utilizaremos el depurador **gdb** integrado en el entorno **Dev-C++**.

Antes de compilar, y para que se añada al fichero ejecutable cierta información que luego utilizará el depurador de programas, debemos comprobar que está seleccionada la siguiente opción:

- **Herramientas** ➔ **Opciones de Compilador** ➔ **Configuración** ➔ **Linker** ➔ **Generar Información de Debug: Yes**.

• Comenzar la depuración

- 1.- Colocar el cursor sobre la primera instrucción de **main**.
- 2.- Pulsar **Ctrl+F5**, o bien mover el ratón a la parte más izquierda de una línea de código y pulsar el botón izquierdo del ratón en dicha posición, o bien **Depurar** ➔ **Añadir/Quitar punto de ruptura**.
Esto hará que quede marcada la instrucción, indicando que se insertado un **punto de ruptura** (la línea quedará marcada en rojo) que obligará al depurador a detenerse en este punto. Para eliminar los puntos de ruptura se realiza la misma operación que para marcarlos.
- 3.- Pulsar **F8**, o bien **Depurar** ➔ **Depurar** (bien en el Menú Principal, bien en las pestañas inferiores), con lo que comenzará la ejecución paso a paso del programa, indicándose en azul cuál será la siguiente línea que se ejecutará. Además aparecerá una ventana de MS-DOS para que se realicen las entradas y salidas que el programa necesite durante su depuración.



- **Control de la depuración.**

- **Siguiente Paso (F7):** Ejecuta la siguiente instrucción del programa (la marcada en color azul). Cuando dicha instrucción contiene una llamada a una función/método, dicha función/método se ejecuta completamente tras seleccionar *Siguiente Paso*. Esta posibilidad es muy útil cuando estamos seguros de que esa función/método no contiene ningún tipo de error.
- **Avanzar Paso a Paso (Shift+F7):** Ejecuta la siguiente instrucción del programa. Si dicha instrucción contiene una llamada a una función o método el *índice de ejecución* (línea marcada en azul) se situará en la primera sentencia de dicha función/método para ejecutarlo *paso a paso*.
- **Saltar Paso (Ctrl+F7):** Ejecuta instrucciones hasta alcanzar el siguiente *punto de ruptura*. Cuando no hay puntos de ruptura situados después de la sentencia actual, se ejecutarán todas las instrucciones hasta finalizar el programa.
- **Ir a cursor (Shift+F4):** Ejecuta sentencias hasta el lugar donde se sitúa el cursor.
- **Parar ejecución (Ctrl+Alt+F2):** Detiene la ejecución paso a paso, sin necesidad de llegar al final del programa.

- **Visualización del contenido de las variables.**

Para ello existe la ventana *Watch*, que se activa pulsando la pestaña **Depurar** situada en la parte izquierda del entorno.

Cuando queremos añadir una variable a dicha ventana podemos hacerlo bien pulsando la opción **Añadir Watch (F4)** para después escribir el nombre de la variable, bien seleccionando la variable con el ratón y pulsando **Añadir Watch**.

Para eliminar una variable de la ventana *Watch*, la marcamos y pulsamos **Quitar Watch**.