

## Mi primer proyecto en Dev-C++

Para realizar esta actividad deberás disponer de un ordenador en el que esté instalado el Dev-C++. Debes ir realizando cada uno de los pasos indicados, en el mismo orden en que aparece en el guión. Lo ideal es tener abierto en la pantalla del ordenador el documento que contiene el guión. Es posible que debas consultar de vez en cuando el manual de Dev-C++. Por tanto, tenlo cerca también.

Al finalizar la práctica serás capaz de:

- Construir un proyecto Dev-C++ partiendo de cero, editando, compilando, montando y ejecutando un programa.
- Construir un proyecto Dev-C++ partiendo de un archivo de código en lenguaje C ya existente.
- Identificar y corregir algunos errores sencillos de compilación

### ■ Paso 1.

Crema una carpeta en la unidad `C:` del ordenador, que se llame, por ejemplo, `C:\Fprog\Practical`.

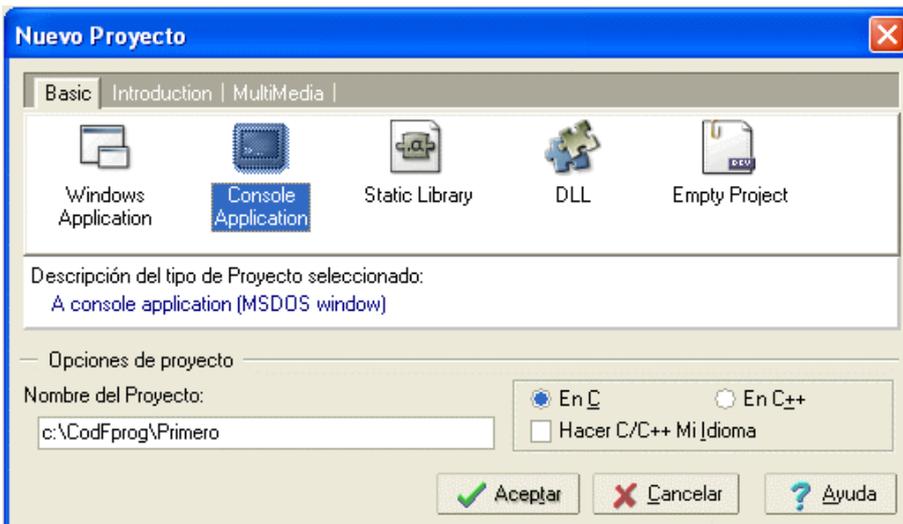
### ■ Paso 2.

Inicia el Dev-C++ y crea un proyecto nuevo. Para ello ve al menú Archivo y haz clic en Nuevo, luego en Proyecto. Un diálogo que contiene los distintos tipos de Proyectos se abrirá.

Aquí se nombran los proyectos básicos:

- Aplicación de Windows (Windows Application): Crea un programa para Windows, que utiliza el API de Win32.
- Aplicación en modo de consola (Console Application): Crea un programa en modo consola.
- Biblioteca estática (Static Lib): Crea un proyecto vacío con las opciones necesarias para construir una biblioteca estática.
- DLL (Dynamic Link Library DLL): Crea una Biblioteca de Enlace Dinámico DLL para Win32.

Ahora cuando has seleccionado un tipo de proyecto, debes introducir el nombre de tu proyecto en donde corresponde y selecciona el lenguaje de programación que usarás en el proyecto C o C++ (en nuestro caso selecciona C), y después haz clic en Aceptar.



■ **Paso 3:**

El siguiente paso es crear un archivo que contenga el código de tu primer programa. Para ello, selecciona la opción *Archivo->Nuevo->Código Fuente*. Aparece una ventana solicitando confirmación, y la pulsar Yes se muestra en la ventana de edición (derecha) el código por defecto que hemos escrito al instalar y configurar el entorno. Acabas de crear tu primer



archivo, que, por defecto, se denomina SinNombre1. Lo primero que tienes que hacer es guardarlo (grabándolo en disco) con la opción *Archivo->Guardar* y el nombre que quieras (por ejemplo, primero.c). Recuerda que las extensiones en el nombre de los archivos permiten identificar el tipo de archivo. En este caso la extensión ".c"

indicará que es un archivo que contiene código en lenguaje C.

```
[*] SinNombre1
/*-----
| Autor:
| Fecha:                      Versión: :
|-----
| Descripción del Programa:
|-----

// Incluir E/S y Librerías Standard
#include <stdio.h>

// Zona de Declaración de Constantes

// Zona de Declaración de Tipos

// Zona de Cabeceras de Procedimientos y Funciones

// Programa Principal
int main()
{
    // Zona de Declaración de Variables del Progr:

    system("Pause"); // Hacer una pausa
    return 0;        // Valor de retorno al S.C
}

// Implementación de Procedimientos y Funciones
```

#### ■ Paso 4.

Escribe el siguiente código en la ventana que acaba de aparecer (si tienes el guión en pantalla, puedes hacer *copy y paste*), eliminando las líneas que sobren:

```
/*-----  
| Autor:                                     |  
| Fecha:                                     | Versión: 1.0  
|-----  
| Descripción del Programa:                 |  
|-----*/  
  
// Incluir E/S y Librerías Standard  
#include <stdio.h>  
  
// Zona de Declaración de Constantes  
  
// Zona de Declaración de Tipos  
  
// Zona de Cabeceras de Procedimientos y Funciones  
  
// Programa Principal  
int main()  
{  
    // Zona de Declaración de Variables del Programa principal  
  
    int numero;  
  
    printf ("Escribe un número: \n");  
    scanf ("%d",&numero);  
  
    if ((numero %2) ==0)  
        printf ("El número es par\n");  
    else  
        printf ("El número es impar\n");  
  
    system("Pause");    // Hacer una pausa  
    return 0;           // Valor de retorno al S.O.  
}  
  
// Implementación de Procedimientos y Funciones
```

El programa tiene, en primer lugar un comentario sobre lo que hace. Después tiene una directiva `#include` para poder usar la librería estándar `stdio`, en la que están las funciones `printf` y `scanf` que permiten leer y escribir cosas en el terminal. A continuación viene la declaración del programa principal (`int main`). En ese programa principal hay una declaración de la variable entera `numero`. Después se pide al usuario que escriba un número en el teclado. El programa lee el número y usando una sentencia condicional averigua si es par o impar, y escribe el mensaje apropiado en la pantalla.

Todas las líneas que comienzan por `//` no forman parte del código, sino que son simples comentarios (puedes escribir lo que quieras) que facilitan la escritura de los programas.

#### ■ Paso 5.

Compila ahora el programa. Para ello, usa la opción *Ejecutar->Compilar*. Recuerda que la compilación consiste en traducir el código escrito en lenguaje C a código en lenguaje máquina (el lenguaje de 0s y 1s que entiende del computador). Comprueba que no ha

habido errores de compilación (sentencias mal escritas), y verifica con el explorador de Windows que el sistema ha generado, dentro de la misma carpeta donde guardaste el archivo `primero.c`, una serie de archivos entre los que se encuentra uno con extensión `".o"` (ese es el que contiene el código en lenguaje máquina).

#### ■ Paso 6.

El siguiente paso es el montaje y la ejecución. El montaje consiste en añadir al código en lenguaje máquina de tu programa, el código correspondiente a la librería que vas a usar (en este caso, la librería estándar `stdio`). Para realizar el montaje, usa la opción *Ejecutar->Compilar y Ejecutar*. Comprueba que no se ha producido ningún error, y que se han generado, dentro de la carpeta, nuevos archivos, entre los cuales está ya el archivo ejecutable. Si no se han producido errores el programa se ejecuta a continuación. Te aparecerá una ventana en la que deberás escribir el número (seguido de un retorno de carro) y te contestará indicándote si el número es par o impar. Comprueba que el programa funciona correctamente. ¡Enhorabuena! Ya tienes tu primer programa en marcha.

#### ■ Paso 7.

Comprueba que en las barras de herramientas hay botones que te permiten realizar con un simple clic las operaciones de compilación, montaje o ejecución, sin tener que pasar por la opción *Ejecutar*, tal y como se ha indicado en los pasos anteriores. Comprueba también que, una vez generado el archivo ejecutable, puedes ejecutar tu programa sin necesidad del Dev-C++. Simplemente, con el explorador de Windows, localiza el archivo ejecutable (en este caso, *primero.exe*), haz doble clic sobre él y volverá a salir la pantalla en la que te pide un número.

#### ■ Paso 8.

Construye ahora un proyecto nuevo con un programa en Dev-C++ que lea de teclado tres números enteros y escriba en pantalla el mayor y el menor. Comienza desde el principio, creando una carpeta nueva para este proyecto, etc.

#### ■ Paso 9.

Con mucha frecuencia te ocurrirá que ya tienes el archivo `".c"` que contiene el código del programa (por ejemplo, ya lo ha preparado un compañero de grupo) y tu tienes que construir el proyecto a partir de ese archivo de código. Veamos cómo hacerlo. Cierra primero el Dev-C++ y crea un nuevo proyecto en una carpeta diferente para que no se mezcle la información de este nuevo proyecto con la del proyecto anterior.

#### ■ Paso 10.

Conéctate a la página web de la asignatura y consigue el archivo que se llama `mayor.c`. Coloca ese archivo en la carpeta que acabas de construir. A continuación utiliza la opción *Proyecto->Añadir a proyecto* para incluir el archivo `mayor.c` en el nuevo proyecto.



#### ■ Paso 11.

Haz doble clic sobre el archivo `mayor.c` en la ventana de la izquierda y observa cómo se pone en marcha automáticamente el Dev-C++. Te mostrará la pantalla típica en la que encontrarás el código del programa. Échale un vistazo y verás enseguida lo que hace. Simplemente lee tres números del teclado y escribe en pantalla el mayor de los tres.

#### ■ Paso 12.

Comprueba que si intentas compilar el programa te aparece un mensaje que indica que todavía no se han creado los archivos correspondientes al nuevo proyecto. Te pregunta si quieres hacerlo ahora y debes contestar que sí. Comprueba que efectivamente te ha generado los archivos típicos (tal y como viste en el paso 2). Comprueba también que NO te ha generado el archivo con extensión “.o”. Esto es así porque el programa tiene algunos errores de compilación (no está bien escrito), además de algunos avisos (warnings). Debemos arreglar esos errores para que pueda compilar correctamente.

#### ■ Paso 13.

Los errores de compilación se muestran en la ventana inferior. Comprueba que se han producido 4 errores de compilación. Verás que en esa ventana te muestra una línea explicativa para cada error. Si haces doble clic sobre el primer error te mostrará la línea de código en la que está el error. Lo recomendable es ir resolviendo los errores uno por uno, empezando por el primero. Una vez resuelto un error, debes volver a compilar para ver que errores quedan por resolver. Ten presente que con frecuencia un solo fallo en la escritura del código provoca varios errores de compilación. En este caso, a pesar de que hay cuatro errores de compilación, en realidad sólo hay tres fallos en el código.

#### ■ Paso 14.

Los mensajes que acompañan a cada error te darán a veces una buena pista sobre el error. En este caso, los mensajes para los dos primeros errores son muy claros, pero seguramente el mensaje para el tercer error no te parecerá tan claro. Es muy importante que te fijes en los mensajes y los asocies con los errores cometidos. Construye, por ejemplo, una tabla en la que copiarás el mensaje de error tal y como sale en la pantalla y añadirás al lado una explicación de en qué consiste el error. De esta forma, cuando te vuelva a aparecer el mensaje podrás recordar el tipo de error correspondiente, lo cual te ayudará a ir más rápido en la corrección de los errores. Intenta corregir ahora los tres errores y poner el programa en funcionamiento.