

# Introducción al Dev-C++

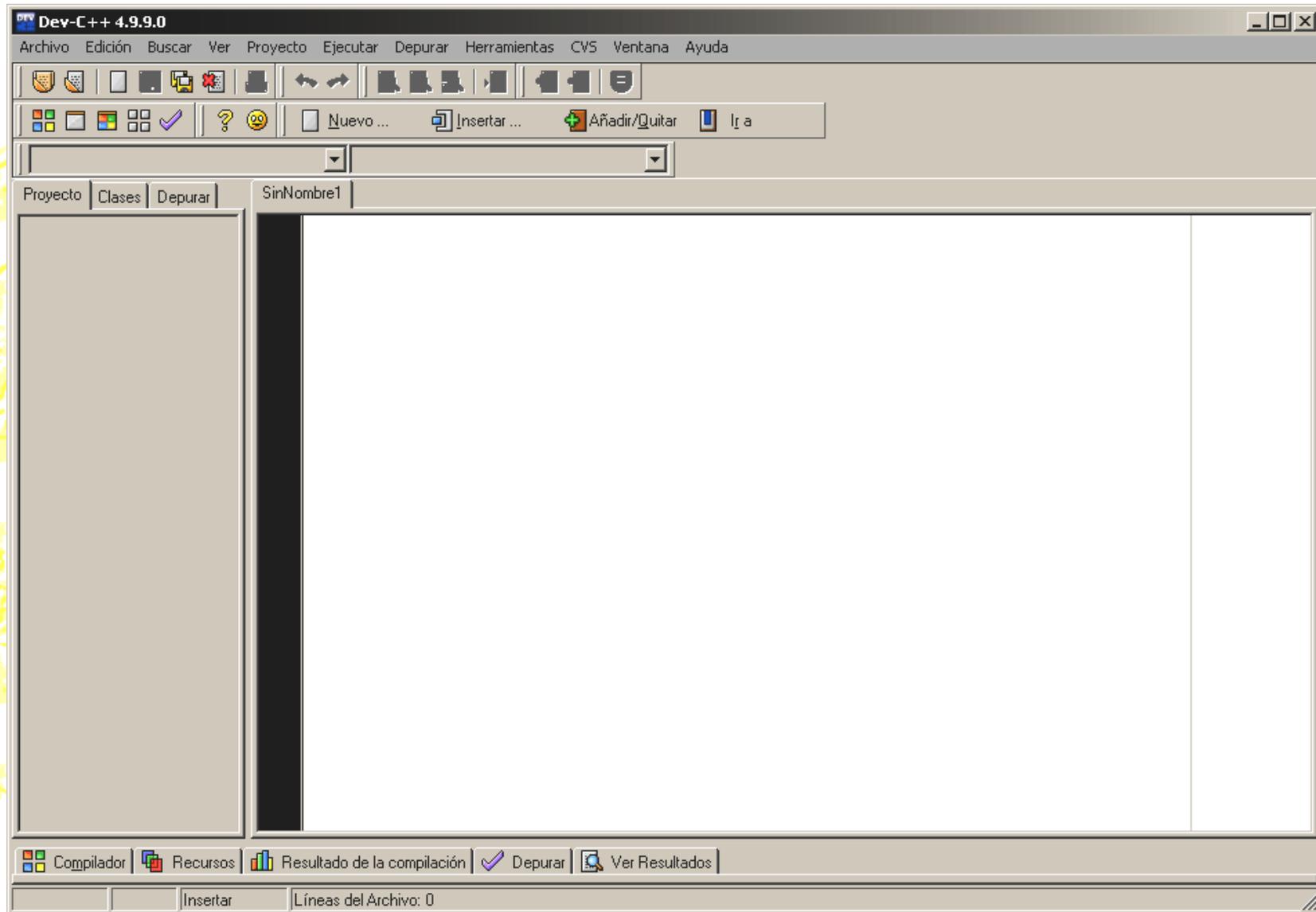


## *El Entorno de Desarrollo Dev-C++*

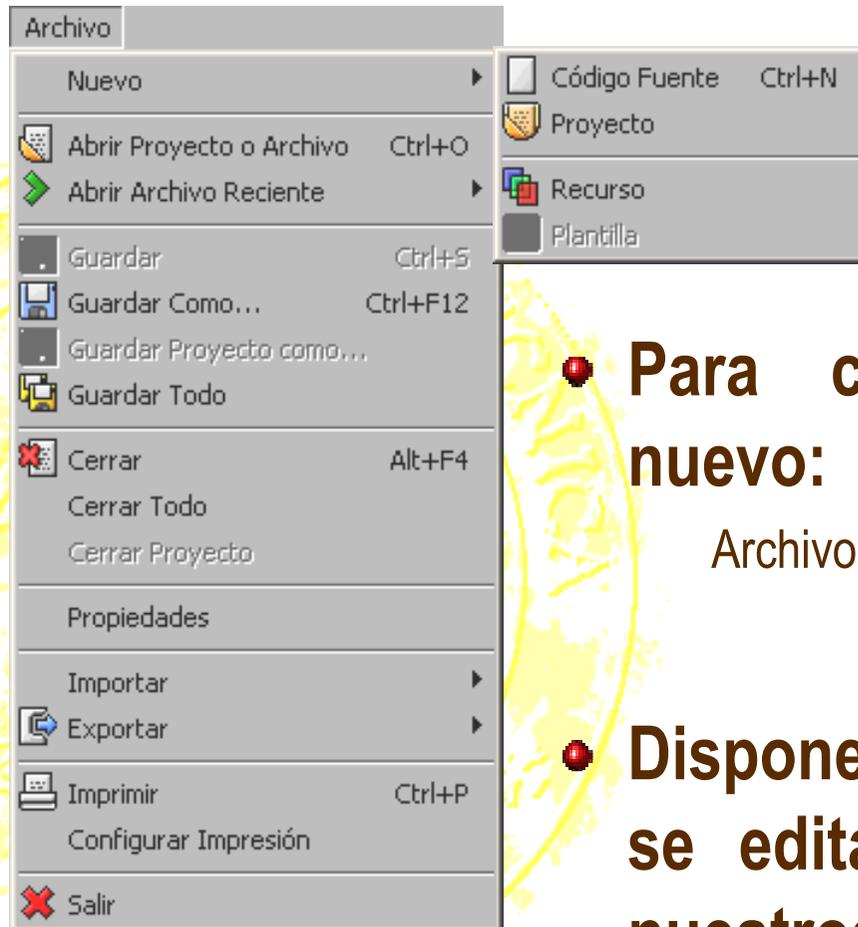
<http://www.bloodshed.net/devcpp.html>

- Incluye todas las operaciones necesarias para crear o modificar un programa.
- Posee un editor, un compilador y un enlazador.
- Permite ejecutar el programa de forma controlada así como depurarlo.

# Visión del Entorno Dev-C++



# Editor



- **Para crear un archivo fuente nuevo:**

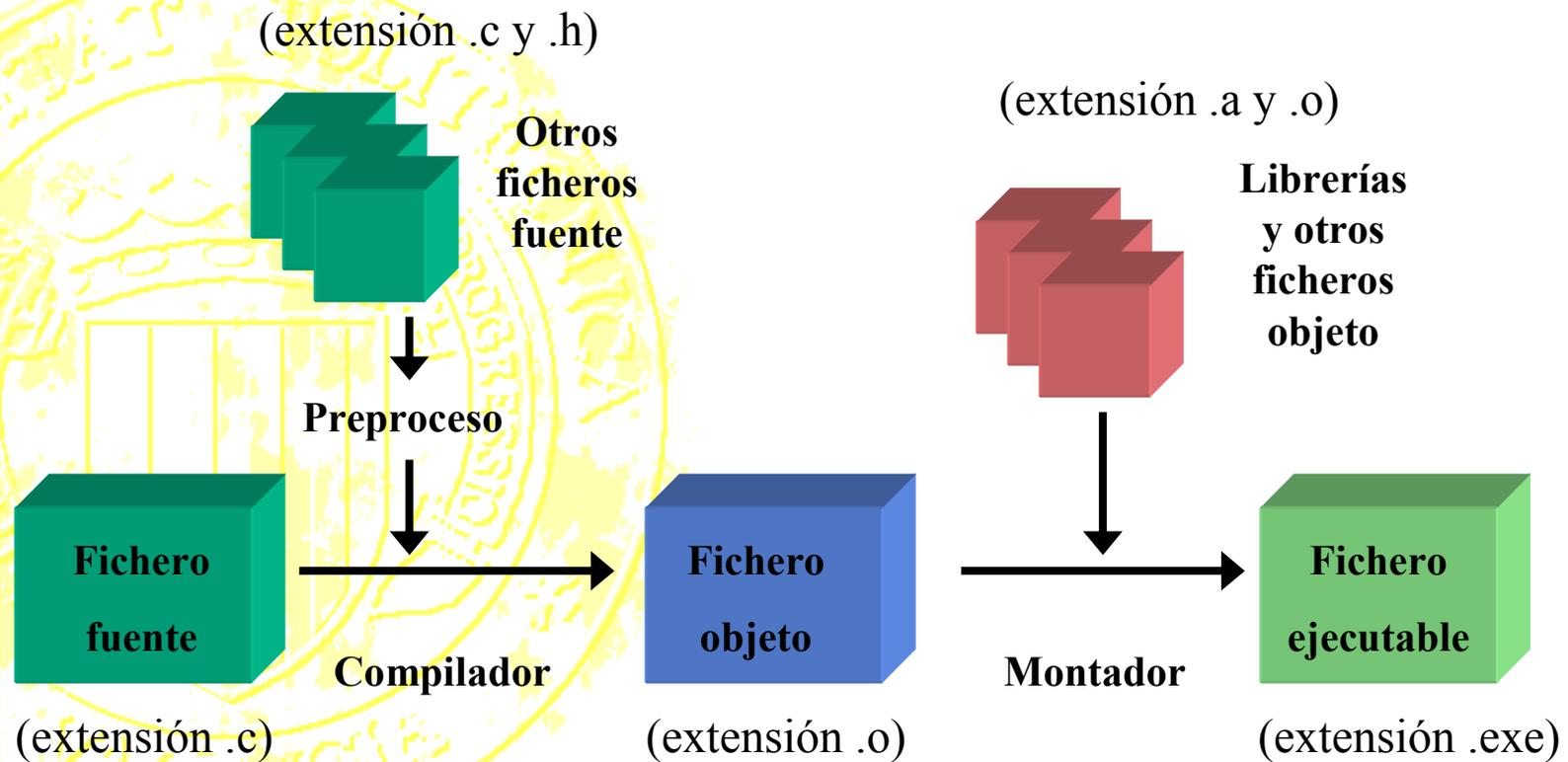
Archivo → Nuevo → Código Fuente (Ctrl+N)

- **Dispone de una ventana en la que se edita el texto y escribiremos nuestros programas.**

## *Tipos de Ficheros en C*

- **Fuente (.c):** Uno o más ficheros escritos en ASCII con las instrucciones del programa.
- **Objeto (.o o .obj):** Son ficheros intermedios creados a partir de la compilación, entendibles por el montador y transparentes al programador.
- **Librería (.a o .lib):** Reune varios ficheros objeto en uno solo. Suelen agrupar funciones estándar.
- **Cabecera (.h):** Contienen definiciones y declaraciones compartidas por varios ficheros fuente, así como las correspondientes a las funciones estándar de las librerías.
- **Ejecutables (.exe):** Se almacena en un único fichero todo el código máquina de los ficheros objeto asociados al programa, ya en un formato ejecutable.

# El Proceso de la Compilación



# Compilación y Ejecución



- **Para compilar un archivo fuente:**

Ejecutar → Compilar (Ctrl+F9)

- **Para ejecutar:**

Ejecutar → Ejecutar (Ctrl+F10)

- **Para compilar y ejecutar:**

Ejecutar → Compilar y Ejecutar (F9)

# Depuración

- **Para ejecutar en modo de depuración:**

Depurar → Depurar (F8)

- **Para parar la ejecución:**

Depurar → Parar ejecución (Ctrl+Alt+F2)

- **Para poner/quitar un punto de parada:**

Depurar → Añadir/Quitar Punto de Ruptura (Ctrl+F5)

- **Para ver variables:**

Depurar → Añadir watch (F4)

Depurar		
<input checked="" type="checkbox"/>	Depurar	F8
<input checked="" type="checkbox"/>	Parar ejecución	Ctrl+Alt+F2
	Parámetros...	
	Añadir/Quitar Punto de Ruptura	Ctrl+F5
	Siguiente Paso	F7
	Avanzar Paso a paso	Shift+F7
	Saltar Paso	Ctrl+F7
	Ir a cursor	Shift+F4
	Añadir watch	F4
	Observar Variables	Ctrl+W
	Ver ventana del CPU	

## Depuración (II)

### ● Opciones de ejecución paso a paso:

#### ● Siguiente Paso (F7)

Ejecuta una instrucción.

Si es una llamada a función, realiza todas sus instrucciones antes de devolver el control.

#### ● Avanzar Paso a paso (Shift+F7)

Ejecuta una instrucción.

Si es una llamada a función, entra en ella y espera antes de ejecutar sus instrucciones. Esto es útil para depurar esa función.

#### ● Saltar Paso (Ctrl+F7)

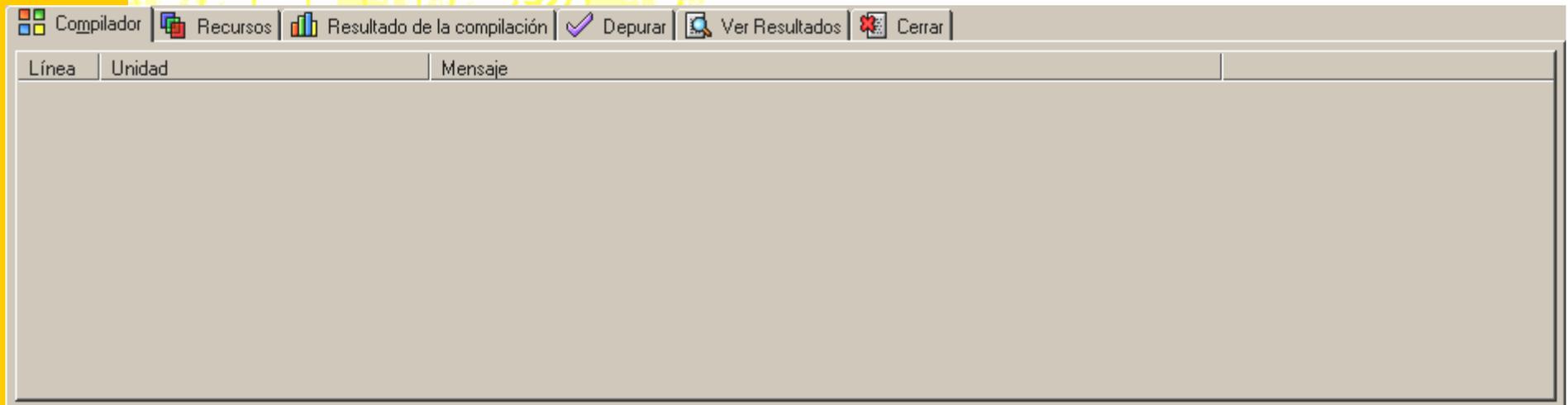
Continúa la ejecución del programa.

#### ● Ir a cursor (Shift+F4)

Coloca un punto de parada donde está el cursor y continúa la ejecución del programa, que parará al llegar a ese punto.

# Errores

- Evidentemente pueden producirse errores en el proceso.
- La ventana del compilador está en la parte inferior de la pantalla, apareciendo si ocurre algún error al compilar.
- También aparece si el compilador ve algo que quizás no sea un error, pero es sospechoso: un *warning*.



# Tipos de Errores

## ● Errores sintácticos.

- Los lenguajes de programación tienen una sintaxis determinada para que puedan ser interpretados por el compilador.
- El compilador detecta estos defectos de forma y muestra este tipo de errores.
- Ej.: Dejarse un punto y coma al acabar una instrucción.

## ● Errores en el enlace.

- Se suele tratar de errores a la hora de nombrar las funciones, en los tipos o número de parámetros o del lugar donde se encuentran al llamar a una función...

## ● Errores en ejecución.

- Estos errores se dan porque en la ejecución de los programas ciertos valores pueden ser ilegales para ciertas operaciones.
- Ejs.: División por cero, la raíz cuadrada de un valor negativo...

## ● Errores semánticos.

- Son los más difíciles de detectar y el entorno no puede ayudarnos, puesto que se tratan de discrepancias entre lo que hace el programa y lo que se pretende que haga.

# El Primer Programa en C

- Introduce el siguiente código fuente en el editor de Dev-C++ y guárdalo como **hola.c**
  - ¡Ojo! Hay que usar la extensión “.c” (minúscula)

```
hola.c
#include <stdio.h>
#include <conio.h>

int main()
{
    printf("Hola, mundo.\n");
    /* getch(); */
    return 0;
}
```

## *El Primer Programa en C (II)*

- **Ahora prueba a compilarlo y crear así el ejecutable.**
- **Como puedes comprobar aún no ha pasado nada.**
- **Ahora ejecútalo desde el menú “Ejecutar”.**
- **Tampoco parece pasar nada.**
- **Descomenta la línea que aparece comentada en el código y repite el proceso.**

## *El Primer Programa en C (III)*

- **Ahora quita del programa un punto y coma.**
- **Compila de nuevo.**
- **¿Qué ocurre? ¿Qué tipo de error se produce?**
- **Vuelve a ponerlo y ahora pon “print” en vez de “printf”.**
- **Trata de compilar.**
- **Vuelve a poner “printf” y ahora cambia “stdio” por otro nombre.**
- **Trata de compilar.**

## Programas Ejemplo

- Ahora prueba este otro programa (para ello usa la opción de menú para crear un nuevo código fuente).

```
#include <stdio.h>
#include <conio.h>
int main()
{
    printf("El resultado es: %d\n",4/0);
    getch();
    return 0;
}
```

- ¿Cuál es ahora el error al intentar mostrar el resultado de dividir 4 entre 0?
- Ahora pon 4/2 , 4/3 ...

## Programas Ejemplo (II)

- Prueba ahora este otro programa:

```
#include <stdio.h>
#include <conio.h>
#define PI 3.1416f
int main()
{
    float radio, longitud;
    printf("Introduce el radio: ");
    scanf("%f",&radio);
    longitud = 2.f*PI*radio;
    printf("La longitud de una circunferencia "
           "de radio %.2f es %.2f.\n",radio,longitud);
    getch();
    return 0;
}
```

- Intenta completarlo para que calcule también el área del círculo.

## Programas Ejemplo (III)

### ● Prueba ahora este otro programa:

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

int main()
{
    printf( "\nTamaño en bytes de los tipos básicos:\n" );
    printf( " char: %d\n", sizeof(char));
    printf( " short int: %d\n", sizeof(short int));
    printf( " int: %d\n", sizeof(int));
    printf( " long int: %d\n", sizeof(long int));
    printf( " float: %d\n", sizeof(float));
    printf( " double: %d\n", sizeof(double));
    printf( " long double: %d\n", sizeof(long double));
    getch();
    return 0;
}
```