

Programación en C

Para escribir un nuevo programa ...

...en **Borland C++**: Hacer **File > New > TextEdit**

Después de escribir el código pulsar  (Botón **Guardar**), y  (Botón **Ejecutar**)

... en **Visual C++**: Hacer **Archivo > Nuevo proyecto > General > Proyecto Vacío**

Luego en el **Explorador de Soluciones** pinchar con el botón derecho en **Archivos de Código Fuente** y hacer: **Agregar > Nuevo Elemento > Código > Archivo c++**

Después de escribir el código pulsar  (Botón **Guardar Todo**), y  (Botón **Ejecutar**)

```
// Poner aquí la descripción del programa (las líneas que comienzan por // son comentarios)
// Poner aquí el nombre del programador
#include <stdio.h> // Poner tantos include's como bibliotecas usemos
int main(void) // main indica el programa principal. Va a continuación entre llaves
{
    // primero declaramos las variables
    // luego escribimos cada una de las líneas
}
```

```
// Mi primer programa
// Autor Sergio A.
#include <stdio.h>
int main(void)
{
    printf("Hola mundo");
    fflush(stdin); getchar();
}
```

// el programa principal estará escrito entre las llaves
// Cada instrucción acaba en ;
// **printf** sirve para imprimir algo en la pantalla
// Esta línea detiene el programa hasta pulsar una tecla

Para ejecutar se recomienda primero guardar (*save*) y luego pulsar el botón ejecutar (*run*)

printf realmente imprime algo y sitúa el curso a continuación.

Si luego hacemos otro **printf**, lo que mandemos imprimir saldrá a continuación.

Si en algún momento queremos pasar a la siguiente línea podemos poner el carácter NUEVA_LINEA (LF): `\n`

```
#include <stdio.h>
int main()
{
    printf("Hola mundo");
    printf("Soy Sergio");
    getchar();
}
```

Hola mundoSoy Sergio

```
#include <stdio.h>
int main()
{
    printf("Hola mundo");
    printf("\nSoy Sergio");
    getchar();
}
```

Hola mundo
Soy Sergio

Las **variables** son letras o palabras a las que se asigna un valor que puede variar.

El valor que se asigna puede ser básicamente de 3 tipos (hay más que ya veremos):

int Un número entero (es decir no tiene cifras decimales)
float Un número real, que tendrá parte entera y parte decimal
char Un carácter

Las variables hay que declararlas, es decir, informar al programa que tipo de dato guardará esa variable. Hay que declarar la variable antes de usarla, por eso lo haremos inmediatamente después de la llave del **main**

Luego le asignaremos un valor. Como es una variable, en cualquier momento podemos darle otro valor a esa variable.

```
#include <stdio.h>
int main(void)
{
    int a;
    float b;
    char c;
    a=5;
    b=7.8;
    c='h';
    printf("La variable a vale %d",a); //printf sustituirá %d por el valor de la variable entera a
    printf("\nLa variable b vale %f",b); //printf sustituirá %f por el valor de la variable real b
    printf("\nLa variable c vale %c",c); //printf sustituirá %c por el carácter guardado en la variable c
    a=8; // Damos un nuevo valor a la variable a
    printf("La variable a ahora vale %d",a); // imprimo el nuevo valor
    getchar();
}
```


Introducción de datos desde el teclado: **scanf**

scanf nos servirá para que el usuario dé un valor por teclado a una variable.

Usaremos el siguiente formato:

```
printf("Introduce el valor del primer numero (entero)=");
scanf ("%d",&a); // guardará en la variable entera a el valor introducido por el teclado
```

```
#include <stdio.h>
int main(void)
{
    int a,cua; // puedo definir varias variables del mismo tipo a la vez
    printf("Dime un numero entero=");
    scanf("%d",&a); // no te olvides de poner el &
    cua=a*a; // Calculamos el cuadrado de a y lo guardamos en la variable cua
    printf("\nSu cuadrado vale %d",cua);
    printf("\n\nEl cuadrado de %d vale %d",a,cua); // observa que puedo mostrar varias
                                                    // variables en el mismo printf
                                                    // se sustituyen en orden
    getchar();
}
```



```
Dime un numero entero=16
Su cuadrado vale 256

El cuadrado de 16 vale 256
```

```
#include <stdio.h>
int main(void)
{
    float a;
    a=8/3; // CUIDADO cuando dividimos dos enteros: Entero/Entero = Entero
           // El resultado es un entero, aunque luego convierte el resultado a float:
    printf("aquí mal: 8/3=%f\n",a);
    a=(float)8/(float)3; // así obligo a considerar el 8 y el 3 como float's
    printf("ahora bien 8/3 = %f\n",a);
    printf("ahora bien 8/3 = %.2f con 2 decimales\n",a);
    printf("ahora bien 8/3 = %.1f con 1 decimales\n",a);
    printf("ahora bien 8/3 = %.0f con 0 decimales\n",a);
    getchar();
}
```

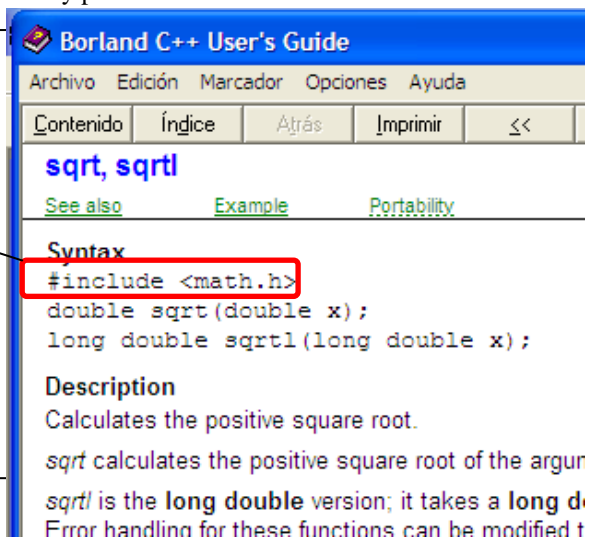
```
Aquí mal: 8/3 = 2.000000
Ahora bien 8/3 = 2.666667
Ahora bien 8/3 = 2.67 con 2 decimales
Ahora bien 8/3 = 2.7 con 1 decimales
Ahora bien 8/3 = 3 con 0 decimales
```

Comentario:

Si utilizo una función que no está en **stdio.h** debo poner el **#include** de la biblioteca que contiene a esa función. Para saber en qué biblioteca está lo mejor es usar la ayuda: selecciono la palabra y pulso F1

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    float n,r;
    printf("Dime un numero=");
    scanf("%f",&n); // no te olvides de poner el &
    r=sqrt(n); // sqrt calcula la raíz cuadrada de n y la guarda en r
    printf("\nLa raíz cuadrada vale %f",r);
    getchar();
}
```

```
Dime un numero=49
La raíz cuadrada vale 7.000000
```



Borland C++ User's Guide

Archivo Edición Marcador Opciones Ayuda

Contenido Índice Atrás Imprimir <<

sqrt, sqrt1

[See also](#) [Example](#) [Portability](#)

Syntax

```
#include <math.h>
double sqrt(double x);
long double sqrt1(long double x);
```

Description

Calculates the positive square root.

sqrt calculates the positive square root of the argur
sqrt1 is the **long double** version; it takes a **long d**
Error handling for these functions can be modified t